Universidad Nacional Autónoma de México
Posgrado en Ciencia e Ingeniería de la Computación

Perfilado de autor en redes sociales aplicando técnicas de transferencia de conocimiento

T E S I S

QUE PARA OPTAR POR EL GRADO DE:

Maestro en Ciencia e Ingeniería de la Computación

PRESENTA:

Aquilino Francisco Sotelo

Director de Tesis:

Dra. Helena M. Gómez Adorno
Posgrado en Ciencia e Ingeniería de la Computación

Ciudad Universitaria, Cd. Mx., enero de 2020

Perfilado de autor en redes sociales utilizando técnicas de transferencia de conocimiento

Tesis de Maestría

Aquilino Francisco Sotelo
Posgrado en Ciencia e Ingeniería de la Computación
Universidad Nacional Autónoma de México

Resumen

Las redes sociales han modificado la forma en que nos comunicamos, ahora es posible establecer contacto con una gran cantidad de personas a las cuales no conocemos. Identificar los rasgos de una persona a partir de lo que escribe se ha convertido en una nueva área de la lingüística computacional llamada *Perfilado de Autor*. Sin embargo, tradicionalmente se hace con textos académicos y en el idioma inglés. Por otra parte, se tiene que las personas escriben muy diferente en las redes sociales que en textos formales, por ejemplo usan abreviaturas no estándares y otros signos como *emoticones* y *emojis*. Esto ha hecho que los métodos tradicionales para conocer los rasgos de las personas a través de lo que publica en sus redes sociales no sean tan eficientes. En este trabajo proponemos aplicar la transferencia de conocimiento para tratar éste problema.

Los sistemas que utilizan transferencia de conocimiento son entrenados en una sola tarea o en varias y después usan el conocimiento aprendido en nuevas tareas. Un ejemplo es entrenar un sistema que clasifique un tipo de imagen y ocupar lo aprendido para que pueda clasificar otro tipo de imagen. Esto difiere del enfoque tradicional de las técnicas estándares de *aprendizaje de máquina*, las cuales se se entrenan en una sola tarea y son probadas en ejemplos seleccionados de esa tarea. Posiblemente el caso de éxito más sonado de la transferencia de conocimiento ha sido el ajuste fino de los pesos de la red de visión utilizando el entrenamiento previo de ImageNet, la cual es una base de datos de imágenes entrenada en una sola tarea.

Este trabajo presenta un enfoque novedoso, proponemos utilizar la transferencia de conocimiento para realizar el perfilado de autor de los textos de *Twitter* en español. A diferencia de los métodos tradicionales nosotros no vamos a quitar los caracteres especiales, sino que los vamos a incluir para que no se pierda información, pues en las redes sociales las personas ocupan casi la mitad de su lenguaje con los signos no estándares.

Author profiling in social networks using Transfer Learning techniques

MSc Thesis

Aquilino Francisco Sotelo
Posgrado en Ciencia e Ingeniería de la Computación
Universidad Nacional Autónoma de México

Summary

Social networks have modified the way we communicate, it is now possible to talk to a large number of people we don't know. Knowing the traits of a person from what he/she writes has become a new area of computational linguistics called *Author Profiling*. However, it is traditionally done with academic texts and in the English language. On the other hand, you have to write very different people in social networks than in formal texts, for example they use unused abbreviations and other signs such as emoticons and emojis. This has made traditional methods to know people's traits through publication in their social networks not so efficient. Therefore, we propose to apply the knowledge transfer to deal with this problem.

Systems that use knowledge transfer are trained in a large number of tasks and then tested in their ability to learn new tasks. An example is to train a system that classifies one type of image and occupy what has been learned so that it can classify another type of image. This differs from the traditional approach of machine learning techniques, which are trained in a single task and are proven in selected examples of that task. Possibly the most popular case of knowledge transfer has been fine-tuning of the weights of the vision network using the previous ImageNet training.

This work presents a novel approach. We propose to use knowledge transfer to perform author profiling of *Twitter* texts in Spanish. Unlike traditional methods we are not going to remove special characters, but we are going to include so that information is not lost, because in social networks people occupy almost half of their language with non-isolated signs.

Jurado asignado:

Presidente:	Dr. Demetrio Fabián García Nocetti
Vocal:	Dra. Helena M. Gómez Adorno
Secretario:	Dr. Gibrán Fuentes Pineda
Suplente:	Dra.Gemma Bel Enguix
Suplente:	Dr. Oscar A. Esquivel Flores

Directora de tesis:

Dra. Helena M. Gómez Adorno

A mi Amigo, el Maestro José Ramón Benito Alzaga:

Dios me dió el ser,
mis padres me concibieron,
y Usted me ayudó a conocer mi Libertad.

Gracias por todo Maestro, esta no es mi maestría, es nuestra maestría.

Agradecimientos

"Un individuo solo, sólo no puede transcender."

Carlos Salinas de Gortari

Tuve que estudiar una maestría para entender el significado de esa frase.

Gracias al Dr. José David Flores Peñaloza, nunca olvidaré que gracias a Usted logré entrar a la UNAM. Al Dr. Armando Castañeda Rojano por la confianza depositada en mí, y por la carta de recomendación que me dio para entrar a la maestría.

A mis sinodales, por el tiempo que dedicaron a este proyecto y sus acertados comentarios. Al Dr. Gibran Fuentes, con quien además tomé un excelente curso de *Deep Learning*, a la Dra. Gemma Bel por sus atinadas observaciones, al Dr. Fabian García, por el tiempo invertido al presente. Y muy especialmente al Dr. Oscar A. Esquivel Flores, por compartir conmigo su vasto conocimiento, por sus atenciones y espero podamos ser amigos.

A mis amigos del posgrado: Diego Velázquez Cervantes, gracias por compartir tu conocimiento y amistad. Carlos Romero Casanova, por tu amistad, las idas a la caritativa y tus consejeros para match. Eduardo López Bolaños, por siempre escuchar y ser un excelente amigo.

AL CUC, gracias por todas esas horas de escucha mutua. Muy especialmente a Lobato por todas sus sugerencias. También a mis compañeros Milton y Manuel por todas las interminables pláticas.

A mis amigos del ITAM: en especial al Rector Dr. Arturo Fernández por toda la ayuda recibida mientras estudié ahí. Al Dr. Carlos Zozaya y Dr. Roberto Zocco, por tantas horas de pláticas y consejos. Al Dr. Emilio Suárez Licona por la oportunidad dada en la Cámara de Diputados. A mi gran amigo el Ministro José Fernando Franco González-Salas, por darme la oportunidad de mi primer empleo y que a pesar de la distancia siempre me ha apoyado.

Al pueblo mexicano que me permitió realizar mis estudios a través de la beca Conacyt. A Lulú, Cecy y Álvaro. A la UNAM, porque en la Universidad la verdad todavía importa.

Finalmente, a mi directora de tesis, Dra. Helena M. Gómez Adorno, por compartir conmigo todo su conocimiento, por apoyarme y confiar en mí, aún cuando yo no lo hacía. ¡Gracias por todo Dra. Helena! Esta no sólo es mi tesis, es ¡nuestra tesis!

Ciudad Universitaria, IIMAS, 2020

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

> "As the light increased I discovered around me an ocean of mist,..."
>
> *Henry David Thoreau - Mount Greylock, MA*

La necesidad de comunicar ideas, experiencias, emociones, etcétera, siempre ha existido, es inherente de la naturaleza humana, por lo cual las personas utilizan diversos medios de comunicación que cambian de vez en cuando.

La comunicación entre las personas se realiza cada vez más a través de las redes sociales, de acuerdo con la Asociación Internet.mx que presentó la edición 14º del "Estudio sobre los Hábitos de los Usuarios de Internet en México 2018", donde se indica que hay 79.1 millones de usuarios de Internet, es decir, 67 % de los mexicanos, que pasan más de 8 horas diarias en este medio, el 89 % accede a través de su *smartphone*, el 89 % lo usa para acceder a sus redes sociales, de los cuales *Facebook* tiene el 98 %, le sigue *Whatsapp* con 91 %, *Youtube* con 82 % y *Twitter* con 57 % [1], esto genera una gran cantidad de información textual, voz y video todos los días, por lo que es de vital importancia que se propongan métodos computacionales para analizarla.

El área de la computación encargada del análisis de información textual es el Procesamiento del Lenguaje Natural (PLN), entendiéndose por lenguaje natural como aquél que utilizan los humanos para comunicarse cotidianamente entre ellos. Existen muchas tareas de PNL como atribución de autoría, análisis de sentimientos, clasificación de textos, entre otros. Para este trabajo se estudiará el perfilado de autor mediante textos.

El Perfilado de Autor (PA) determina las características del o los autores, como su género, nacionalidad, estado emocional, personalidad, entre otros rasgos aunque los escritores no sean conscientes de ello. Existen dos fuentes textuales dónde realizar el PA: textos formales e informales, los primeros tienen cierta estructura y siguen reglas, en cambio los segundos no siguen reglas y no están estandarizados como son las redes sociales. Para el presente trabajo se usarán textos informales, en específico textos de *Twitter*.

La tarea de PA para textos informales es sumamente compleja ya que tiene los inconvenientes propios del lenguaje como la ambigüedad y vaguedad, por otra parte las redes sociales no siguen las normas de escritura como abreviación, signos de puntuación, además incluyen nuevos caracteres como *hashtag* #, mención a usuario @ para citar, etc.

[1] https://www.asociaciondeinternet.mx/es/component/remository/func-startdown/81/lang,es-es/?Itemid=, para un análisis detallado. Fecha de consulta:15 de julio de 2019

Dada la importancia y la enorme cantidad de información textual que se produce en las redes sociales es necesario contar con métodos computacionales que nos permitan analizarla de forma automatizada.

1.1. Motivación

Con la información que las personas publican y consumen en sus redes, sean conscientes o no, las empresas pueden perfilar a sus clientes, los gobiernos pueden hacer mejor sus tareas de seguridad, por ejemplo identificar potenciales casos de pedofilia, secuestros virtuales, entre otros. De hecho los proveedores de éstos servicios ya perfilan a los usuarios, por ejemplo *Twitter* en sus términos de privacidad establece que:

"Cuando su navegador o dispositivo lo permiten utilizamos tanto cookies de sesión como cookies permanentes para ***conocer*** *mejor cómo interactúa usted con nuestros servicios, obtener patrones de uso agregado y personalizar y administrar de otras formas nuestros servicios, como proporcionando seguridad a la cuenta, personalizando el contenido que le mostramos, incluyendo los anuncios, y recordando sus preferencias de idioma."* [2]

Aunque no es parte de éste trabajo vale la pena cuestionar sobre ética de éstos actos de las empresas, ¿tiene derecho de espiar mi actividad?, ¿pueden vender mi información y perfil?, ¿para qué ocupan el conocimiento que tienen sobre nosotros? No lo sabemos y es una laguna jurídica, política y social que se está permitiendo, el tiempo nos mostrará las consecuencias de esto.

En otro ámbito más técnico, las computadoras entienden muy poco del significado del lenguaje humano, esto limita nuestra habilidad de dar instrucciones a las computadoras para que procesen los datos que les damos y nos ayuden a aplicar los modelos de *aprendizaje de maquina* para el perfilado de autor, también afecta la habilidad de ellas de explicarnos sus acciones y su habilidad para analizar y procesar textos [23], por lo que necesitamos técnicas eficientes que procesen y "entiendan" nuestro lenguaje.

Si bien los métodos tradicionales han demostrado tener buenos resultados y son el sistema base (*baseline*) en la tarea de perfilado de autor, el paradigma estándar con el que trabajan es enfocarse en una sóla tarea en particular, por ejemplo atribución de autoría, y después entrenar un modelo, para aprender una tarea desde cero porque no requiere conocimiento humano substancial o experiencia para resolver la tarea. Sin embargo, desde la perspectiva del aprendizaje humano, no tiene sentido que un modelo aprenda una sola, individual tarea desde cero. Esto es como si le pidiéramos a un niño que tradujera del inglés al español sin antes haber desarrollado un entendimiento de un vocabulario básico en cualquiera de los dos idiomas [10].

1.2. Objetivos

El objetivo general de este trabajo consiste en:

> Desarrollar un modelo de identificación automática de perfiles de autor utilizando técnicas de transferencia de conocimiento.

Los objetivos particulares consisten en:

- *Estudiar técnicas de transferencia de conocimientos que puedan ser utilizadas para obtener características textuales.*

[2] https://twitter.com/es/privacy#update, para un análisis detallado. Fecha de consulta:1 de agosto de 2019

- *Medir el impacto del preprocesamiento de textos en las técnicas de transferencia de conocimiento estudiadas.*
- *Utilizar el modelo de Universal Sentence Encoder (USE), para obtener un modelo de identificación de perfiles de autor.*
- *Evaluar el modelo de perfilado de autor con el corpus del foro de evaluación en textos forenses digitales (PAN 2017).* [3]

1.3. Hipótesis

La transferencia de conocimiento ayuda mejorar el rendimiento de los algoritmos de aprendizaje automático que han sido entrenados sobre características tradicionales como la bolsa de palabras o n-gramas de caracteres, para la identificación de perfiles de autor.

1.4. Preguntas de investigación

Las preguntas de investigación surgen en este trabajo son:

1. ¿Qué ventajas y desventajas tiene aplicar las técnicas de transferencia de conocimiento en lugar de los métodos tradicionales para la identificación de perfiles de autor?
2. ¿El preprocesamiento afecta el desempeño del PA utilizando USE?
3. ¿Qué características específicas del corpus de *Twitter* influyen más para realizar PA?

1.5. Contribución

Las contribuciones de este trabajo son:

1. Evaluación del impacto del preprocesamiento de textos de redes sociales en los modelos de transferencia de conocimiento.
2. Modelo de perfilado de autor utilizando técnicas de preprocesamiento de textos y transferencia de conocimiento con el *Universal Sentencer Encoder* (USE).

1.6. Organización de la tesis

Este trabajo se desarrolla en 6 capítulos, incluyendo esta introducción, que a continuación se describen:.

En el Capítulo 2, se presentan los **Antecedentes**. Donde se citan algunas definiciones necesarias para entender los demás capítulos, además se describen los algortimos de aprendizaje utilizados y se describe las propiedades del *Universal Sentence Encoder*.

[3] PAN: Foro de evaluación en textos forenses digitales: `https://pan.webis.de/clef17/pan17-web/author-profiling.html`

En el Capítulo 3, **Estado del arte**. Se presentan los métodos que han tenido los mejores resultados en la tarea de Perfilado de Autor. Estos siguen la metodología de *aprendizaje de máquina*, entre los que destacan máquinas de vectores de soporte y regresión logística.

En el Capítulo 4, **Metodología propuesta**. En este capítulo se explica con detalle el método propuesto para la tarea de PA en *Twitter*. Se empieza explicando el conjunto de datos que fue utilizado, las técnicas de preprocesamiento evaluadas, los algoritmos de aprendizaje utilizados y se describen las métricas para la evaluación del método propuesto.

En el Capítulo 5, **Resultados experimentales**. Se presentan los resultados obtenidos del modelo y su evaluación con las distintas métricas propuestas. También se realiza una discusión sobre el desempeño del modelo en general.

Se concluye el trabajo con una sección 6 de **Conclusiones y Trabajo Futuro**. Se exponen las conclusiones y comentarios finales del método propuesto, también se sugiere el trabajo a realizar en próximas investigaciones.

Capítulo 2

Antecedentes

> "You shall know a word by the company it keeps."
>
> *John Rupert Firth, (1957, 'A synopsis of linguistic theory')*

En este capítulo se describen conceptos y términos que se utilizarán en el desarrollo de esta tesis. En especial, se describirán los aspectos relacionados con el perfilado de autor, haciendo énfasis en la determinación del sociolecto (manera de hablar propia de las personas que pertenecen a un mismo grupo sociocultural) por medio de la extracción de características léxicas, sintácticas y semánticas.

2.1. Modelos de clasificación

Una definición popular del aprendizaje automático, es la dada por el Dr. Yoshua Bengio como: *"La investigación del aprendizaje automático es parte de la investigación de la inteligencia artificial, busca dotar de conocimiento a las computadoras a través de los datos, observaciones e interacciones con el mundo. El conocimiento adquirido permite a las computadoras generalizar correctamente nuevas observaciones."* [1]

De la definición anterior, se desprende que es necesario algoritmos para obtener conocimiento de los datos. Para ello, los *algoritmos de aprendizaje* procesan datos etiquetados, es decir, reciben un conjunto de datos con sus respectivas clases, y generan una función que pueda hacer predicciones a partir de nuevas entradas (un conjunto de datos diferente).

En [6] se establece que la clasificación de patrones trata de categorizar algún objeto dentro de una de las categorías llamadas clases a partir de un conjunto de patrones asociados a un objeto. Por *patrón* se entiende el vector de datos x de dimensión p, $x = (x_1, x_2, ..., x_p)$, donde las x_j son los pesos de las características de un objeto, los cuales son especificados por quien realiza el modelo, usualmente lo hacen basándose en modelos previos o por su experiencia obtenida.

El problema de clasificación consiste en etiquetar los datos de entrada en clases discretas, por ejemplo: de una imagen determinar si se trata de un perro o gato. Un modelo de clasificación intenta obtener conocimiento de los valores observados. Esto puede ser clasificación binaria o multiclase. Dada un dato de entrada el modelo tratará de predecir el valor de una o

[1] Fragmento traducido al español de https://emerj.com/ai-glossary-terms/what-is-machine-learning/ Fecha de consulta: 5 de octubre de 2019+

más clases. Para el presente trabajo utilizaremos la clasificación supervisada que a cada dato de entrada le asocia una *etiqueta* la cual define la clase del objeto.

Los clasificadores de aprendizaje supervisado emplean un conjunto de pares entrada-salida (un vector x_i y una etiqueta y_i asociada al vector de entrada), mediante el cual se aprende una función de decisión que asocia a un nuevo vector x_i una etiqueta de clase dentro de las clases establecidas.

En [16] se define el perfilado de autor (PA) como un problema que consisten en identificar los rasgos demográficos del autor del texto. Por rasgos demográficos se entiende sexo, edad, nivel de estudios, nacionalidad, nivel socio-económicos, entre otros. Por lo que concluimos que el perfilado de autor es un problema de clasificación multiclase.

Para poder realizar el PA a los textos, se pueden utilizar varias técnicas de aprendizaje supervisado. A continuación describimos los algoritmos de clasificación más utilizados para resolver el problema de PA y detallamos conceptos necesarios para el correcto entendimiento de este trabajo de tesis.

2.1.1. Funciones de decisión

Las característica de la clasificación binaria están dadas por el patrón x, el cual puede pertenecer a una de dos clases posibles. Para determinar a cuál pertenece ocupamos las *funciones de decisión*. Vamos a poner un ejemplo para entenderlas. Supongamos que las funciones $f_1(x)$ y $f_2(x)$ definen a las clases 1 y 2 respectivamente, clasificamos al punto x dentro de 1 o 2 si :

$$x = \begin{cases} 1 \text{ si } & f_1(x) > 0 \text{ y } f_2(x) < 0 \\ 2 \text{ si } & f_1(x) < 0 \text{ y } f_2(x) > 0 \end{cases}$$

Para cumplir con su objetivo las funciones de decisión se *entrenan*. El entrenamiento consiste en encontrar funciones a partir de pares de entrada-salida. Usualmente los métodos de entrenamiento encuentran funciones para que cada entrada-salida sea correctamente clasificada en la clase a la que pertenezcan. La figura 2.1 muestra un ejemplo de cómo las funciones de decisión separan perfectamente a la clase 1 de la clase 2.

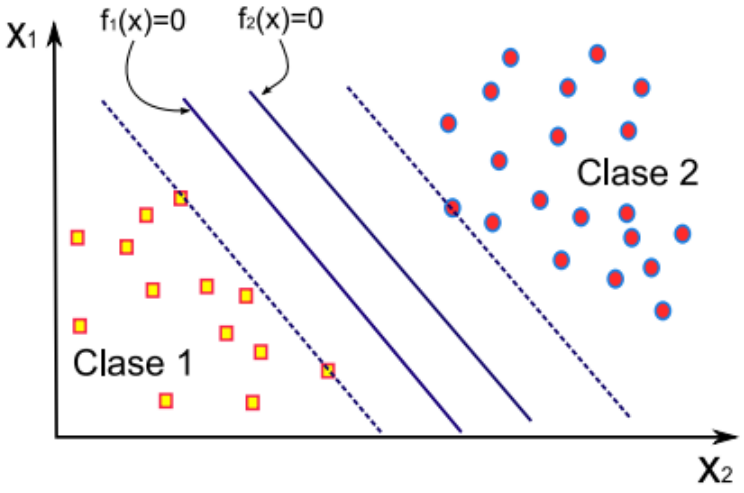

Figura 2.1: Funciones de decisión. Imagen tomada de [6]

2.1.2. Regresión logística

La regresión logística es un modelo predictivo que describe la relación entre una variable de respuesta dependiente, que puede ser dicotómica (sólo puede tomar dos posibles valores) o politómica (más de dos posibles valores), y una o mas variables explicativas independientes. Nos dice la probabilidad de que ocurra algún valor de la variable de respuesta. Para el presente trabajo usaremos el caso dicotómico, a continuación pondremos un ejemplo. Supongamos un experimento donde sólo se tenga dos posibles respuestas :

$$y = \begin{cases} 1 & \text{si tiene la característica} \\ 0 & \text{no tiene la característica} \end{cases} \tag{2.1}$$

La regresión logística intentar predecir la probabilidad de que ciertos datos pertenezcan a la clase 1, (ver 2.2) contra la probabilidad de que pertenezca a la clase 0, (ver 2.3.)

$$Pr[y_i = 1|x_i; \beta] = h(x_i) \tag{2.2}$$

$$Pr[y_i = 0|x_i; \beta] = 1 - h(x_i) \tag{2.3}$$

Se le llama regresión logística por la función en la cual se basa, la **función logística** (ecuación 2.4) también llamada función *sigmoide*, mapea los valores de entrada entre 0 y 1 como se muestra en la figura 2.2, pero nunca exactamente en la frontera.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.4}$$

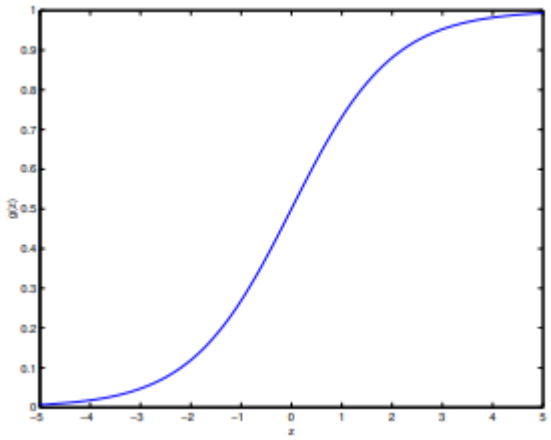

Figura 2.2: Función logística

La regresión logística utiliza la función logística para obligar que el resultado quede entre 0 y 1.

2.1.3. Máquinas de vectores de soporte

Las máquinas de vectores de soporte (*Support Vector Machine* SVM), fue introducida por Vapnik [7], son clasificadores de aprendizaje supervisado. Una SVM pretende construir un separador de clases para un conjunto de datos etiquetados, esto es, dado un conjunto de datos

de entrenamiento encuentra un separador que encuentra la distancia máxima entre las dos clases, para ello utiliza los vectores de soporte, para que dado un nuevo dato de otro conjunto prediga a qué clase pertenece.

Como se muestra en figura 2.3 a pesar de que existe un número infinito de hiperplanos que realizan tal tarea, sólo un hiperplano garantiza el máximo margen de separación, al se le llama *hiperplano de separación óptima.*

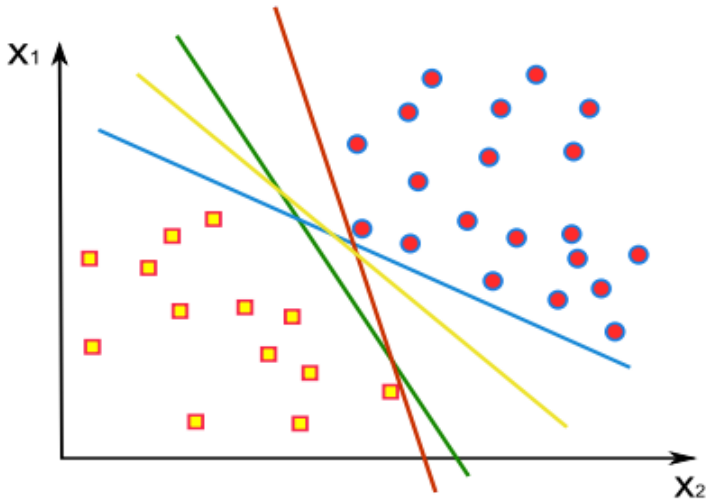

Figura 2.3: Ejemplos de hiperplanos de separación, de entre los infinitos posibles. Imagen tomada de [6]

El caso linealmente separable en el espacio de características para clasificación binaria, es el más simple de las SVM. Los datos son linealmente separables y existen diferentes hiperplanos que pueden realizar la separación. Se optimiza el margen geométrico fijando el margen funcional $\kappa_i = 1$, por lo que el clasificador lineal queda de la siguiente manera:

$$y_i = \begin{cases} 1 \quad \text{si} \quad (w \cdot x^+) + b = 1 \\ -1 \quad \text{si} \quad (w \cdot x^-) + b = -1 \end{cases} \tag{2.5}$$

donde w define el hiperplano de separación óptima y b es el sesgo.

A la distancia entre el hiperplano de separación y el dato de entrenamiento más cercano a él, se le llama *margen.* Al resolver el problema de programación cuadrática se trata de encontrar el hiperplano óptimo y dos hiperplanos H_1 *y* H_2 paralelos.

Cuando la distancia entre H_1 y H_2 es la máxima, quedan puntos tanto sobre H_1 y H_2, éstos puntos son llamado *vectores de soporte* porque participan directamente para definir el hiperplano de separación, es decir, la solución de las SVM esta dada por éste pequeño conjunto de vectores de soporte. La figura 2.4, nos muestra la representación geométrica del problema de programación cuadrática con el separador óptimo H y los hiperplano H_1 y H_2.

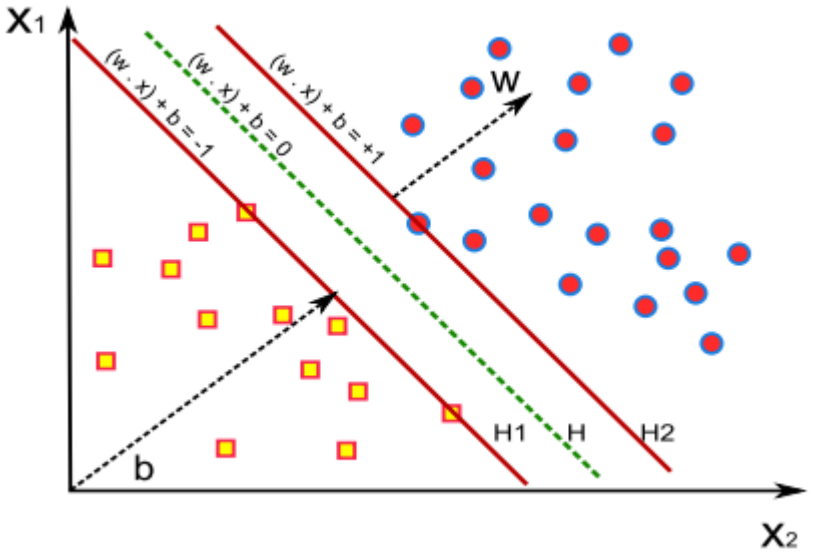

Figura 2.4: Hiperplano de separación óptimo H y su margen asociado (máximo) a sus vectores de soporte H_1 **y** H_2 Imagen tomada de [6]

2.1.4. Redes neuronales

La unidad más simple de una red neuronal es la **neurona artificial** [13], la cual está inspirada en las **neuronas** del cerebro. En la figura 2.5 se muestra una imagen de la neurona biológica (izquierda) y la del modelo matemático (derecha). Cada neurona esta conectada a las demás neuronas a través de la **sinapsis**, recibe señales de entrada de sus **dendritas** y produce una señal de salida a través de su **axón**. El axón eventualmente se ramifica y se conecta vía las sinapsis a las dendritas de otras neuronas.

La neurona artificial está compuesta por valores de entrada $x_1, x_2, ..., x_n$, pesos $w_1, w_2, ..., w_n$, un sesgo b, una función de activación f y una salida y como se muestra en la figura 2.5.

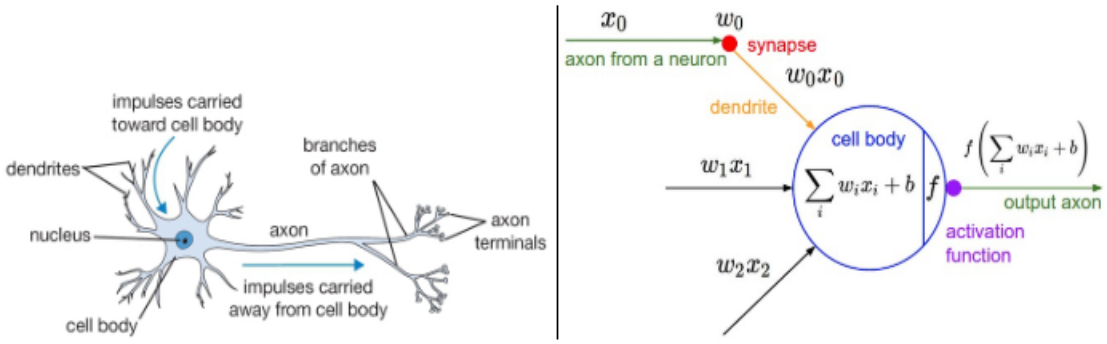

Figura 2.5: A la izquierda la representación de una neurona del cerebro. A la derecha el modelo matemático más simple de una nuerona artificial. (Imagen tomada del curso ÇS231n: Convolutional Neural Networks for Visual Recognition(spring 2019)"de la universidad de Stanford).

La neurona se define matemáticamente como la suma de todas sus entradas multiplicadas por sus pesos más un sesgo (definida en la ecuación 2.6) y esto se pasa por una función de activación f, (ver ecuación 2.7).

$$z = \sum_{i=0}^{n} (w_i \cdot x_i) + b \tag{2.6}$$

$$y = f(z) \tag{2.7}$$

El propósito de la función de activación es introducir la no linealidad a la red, lo que nos permite aproximar funciones complejas arbitrarias.

La arquitectura básica de una red neuronal es la de *propagación hacia adelante con capas totalmente conectadas* (*Feed Forward*, FF), que consiste en un conjunto de neuronas conectadas en un sólo sentido. Consta de tres capas como se muestra en la figura 2.6: la **capa de entrada** (*input layer*) representa los valores que recibe la red, la **capa oculta** (*hidden layer*) que pueden ser varías capas ocultas y cada una consta de neuronas simples, y la **capa de salida** (*output layer*) que representa el resultado de la red.

Para saber cuantas capas ocultas se necesitan y la cantidad de neuronas por capa, actualmente se hace por experimentación.

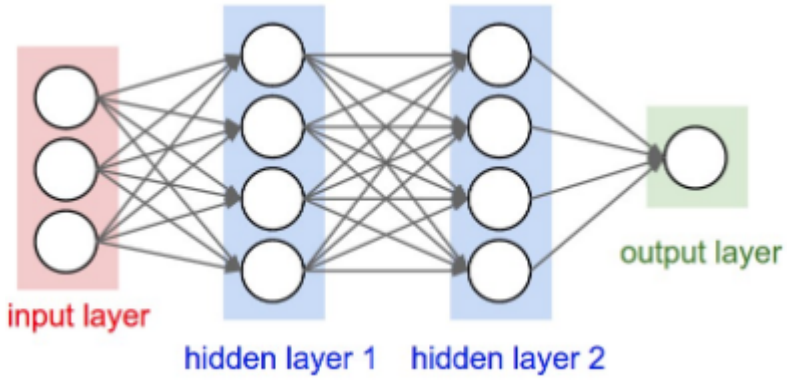

Figura 2.6: Red *Feed forward*. (Imagen tomada del curso ÇS231n: Convolutional Neural Networks for Visual Recognition(spring 2019)"de la universidad de Stanford).

2.1.5. Redes neuronales convolucionales

De todas las arquitecturas de redes neuronales que existen nos interesan las Redes Neuronales Convolucionales (*Convolutional Neural Networks, CNN, ConvNets*). Esta arquitectura de red fue diseñada inicialmente para la clasificación de imágenes [20]. Surgieron porque con las redes FF, no se podía escalar a imágenes grandes Para explicarlo tomemos el ejemplo de una imagen de CIFAR-10, que son de tamaño $32X32X3$ (32 de ancho, 32 de alto y 3 canales de color), en un red FF se tendrían $32 * 32 * 3 = 3072$ pesos, ahora si la imagen de un tamaño $200x200x3$, se tendrían $200 * 200 * 3 = 120,000$ pesos.

A diferencia de las FF, las capas de una CNN tienen tres dimensiones: **ancho, alto, profundidad**, la profundidad se refiere a la tercera dimensión de un volumen de activación, no a las capas escondidas. En la figura 2.7 , se observa la diferencia entre una FF y una

CNN. La convolución de una imagen se hace a través de filtros, obviamente, diferentes filtros producen diferentes imágenes.

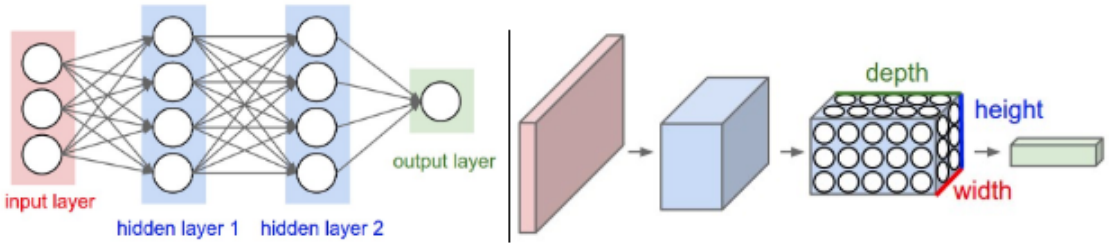

Figura 2.7: Red *A la izquierda una red FF*, a la derecha una red CNN. (Imagen tomada del curso "CS231n: Convolutional Neural Networks for Visual Recognition(spring 2019)" de la universidad de Stanford).

En la práctica muy pocas personas entrenan una CNN desde cero, porque es muy raro que se tenga un *dataset* de suficiente tamaño. En su lugar, es muy común preentrenar una CNN en un *dataset* muy grande como ImageNet[2]. Esto se tratará a detalle en la sección 2.4.

2.2. Extracción de características de textos

Debido a la gran cantidad de información que se produce a diario en forma de textos escritos en lenguaje natural en las redes sociales, es imposible que se analicen completamente todos y cada uno de los aspectos de los mismos. En su lugar nos interesan las representaciones de los documentos, es decir sus *características* representativas. Entre mejor sean éstas mejor se podrán realizar las tareas del procesamiento del lenguaje natural.

Las características de los textos son términos específicos, que nos permiten analizar y extraer patrones útiles o conocimientos a partir de los documentos analizados. En el pasado, ésta tarea se realizaba *a mano*, lo que evidentemente limitaba la cantidad de información que se podía analizar. Sin embargo, con el avance de la ciencia y tecnología cambiaron la forma para la extracción de características, siendo los algoritmos de aprendizaje automático los más utilizados. Para ésta sección nos basaremos en el trabajo de Gómez [16] que describe los siguientes esquemas de representación de textos.

2.2.1. Bolsa de palabras

Para poder tratar con documentos completos es necesario utilizar una estructura que sea computacionalmente viable, para cumplirlo vemos a los documentos como cadenas [15]. Se define una cadena $S = s_1, s_2, ..., s_k$, donde una palabra es una subcadena de S de longitud 1, que de acuerdo con [16], se puede referir a:

- El elemento tal cual está en el texto.
- El elemento en minúscula o mayúscula.
- La palabra con su etiqueta de partes de la oración (POS).

[2]http://cs231n.github.io/transfer-learning/. Fecha de consulta: 10 de octubre de 2019

- Lema de la palabra.
- Cualquier otra variante de la palabra.

Por ejemplo, el ítem "palabras" puede ser representada por las siguientes subcadenas de longitud uno: "palabras","PALABRAS","palabra_N", etc. Es decir, sólo nos interesan las palabras distintas y su frecuencia.

2.2.2. *N*-gramas

Sea $S = s_1, s_2, ...s_k$ una cadena. Los n-gramas se definen en Gusfield [15] como subcadenas de S de longitud n. Los 1-gramas se llaman unigramas, los 2-gramas se llaman bigramas y así sucesivamente.

Hay dos tipos de n-gramas, los de **palabras** y los de **caracteres**. Los de palabras se refieren a n-palabras continuas del documento. En cambio, los de caracteres se refieren a los n-caracteres dentro del límite de la palabra sin incluir espacios.

Por ejemplo, para la oración "Soy un ene grama" se tiene:

- 2-gramas de palabras: Soy un, un ene, ene grama
- 3-gramas de palabras: Soy un ene, un ene grama
- 2-gramas de caracteres: So, oy, y_, _u, un, ..., ma
- 3-gramas de caracteres: Soy, oy_, y_u, ... , ama.

2.2.3. *N*-gramas sintácticos

Debido a que las personas comunican sus ideas, componiendo palabras relacionadas entre sí dentro de unidades más grandes para transmitir significados complejos, necesitamos entender la estructura de las oraciones para interpretar correctamente el lenguaje, para ello necesitamos conocer que está conectado con qué.

Precisamente los n-gramas sintácticos tratan de captar la estructura lingüística[3]. Organizan las palabras en componentes anidados. Muestran la dependencia entre las palabras, esto lo indican a través de flechas (ver figura 2.8):

- Se empieza con unidades: es decir a cada palabra se asocia una etiqueta POS (*part of speech*).
- Las palabras se combinan en oraciones con categorías.
- Las oraciones pueden combinarse en oraciones más grandes de manera recursiva.

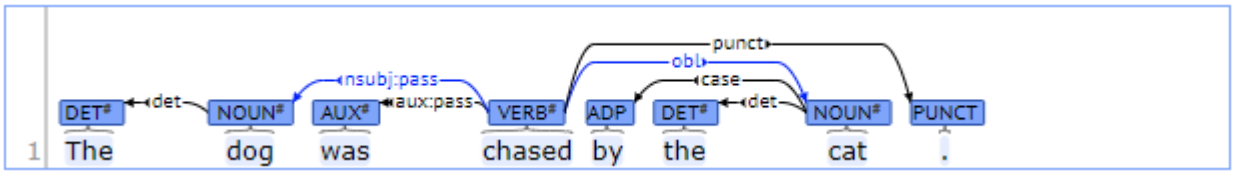

Figura 2.8: N-grama sintáctico. Imagen obtenida de https://universaldependencies.org/ introduction.html

[3]https://web.stanford.edu/class/cs224n/slides/cs224n-2019-lecture05-dep-parsing.pdf. Fecha de consulta:9 de octubre de 2019

2.3. Esquemas de pesado

Para obtener una representación del documento se lleva a cabo un preprocesamiento del texto para verlo como un vector, cada elemento del vector refleja una característica del documento. Para representar cada característica es común asignar algún peso según su relevancia, a este proceso se le llama *esquema de pesado*. A continuación se describen los más relevantes.

2.3.1. Modelo booleano y frecuencia de términos

La forma más natural e intuitiva para asignar pesos es identificar si un término aparece o no, lo que hace el modelo booleano. Otra manera es contar cuantas veces aparece un término en un texto. En ambos casos se asigna a cada término un peso el cual depende del número de ocurrencias del mismo.

2.3.2. Frecuencia inverso de documento

El esquema anterior presenta problemas con la asignación de los pesos, esto es, trata a todos las palabras como si fueran igual de importantes. En este caso los términos muy frecuentes en una colección documentos tendrán mucho peso y los no comunes poco. Para evitar esto, se emplea mecanismos para atenuar el efecto de términos que ocurran con mucha frecuencia en un conjunto de documentos.

Para reducir el peso de la frecuencia del término (TF, *Term Frequency*) se multiplica por un factor que decrece con su frecuencia en el conjunto de documentos. Por lo tanto, las palabras con mayor peso serán las más relevantes para el documento.

Se define el IDF de un término t con frecuencia total en la colección, de la def_t, de la siguiente manera:

$$idf_t = log((N)/df_t) \tag{2.8}$$

2.3.3. TF-IDF

TF-IDF es el producto de TF por IDF. Para asignar un peso a las palabras en un documento, esta medida recibe un conjunto de documentos y con base en la frecuencia de la palabra en el documento y en el conjunto de documentos, se calcula el peso de la siguiente forma:

$$tf - idf_{t,d} = tf_{t,d} \times idf_t \tag{2.9}$$

2.3.4. Vectores de palabras

Un enfoque diferente a los anteriores esquemas de pesado son los *vectores de palabras* (*word embeddings*). Para este método existen dos enfoque, uno *discreto* y otro *distribucional*.

La idea del **enfoque discreto** es representar una palabra en un vector de dimensión n, con un 1 y los demás con 0's, a estos también se les conoce como *one-hot vectors* (ver figura 2.9):

motel	=	[000000010000]
hotel	=	[000001000000]

Figura 2.9: One hot vector

Donde n es el número de palabras en el vocabulario (por ejemplo, $500,000$).

En cambio, en el **enfoque distribucional** se toma en cuenta la similitud entre los propios vectores, se basa en la idea de J. R. Firth 1957: "*You shall know a word by the company it keeps*". En este enfoque cuando una palabra w aparece en un texto, su *contexto* es el conjunto de palabras que aparecen cerca de ella (una ventana de tamaño fijo), como se muestra en la Figura 2.10.

estabamos sentados en un **banco** de madera muy comodos
el **banco** cierra a las seis
tenemos un **banco** de datos muy grande

Figura 2.10: La palabra "banco"se asocia a varios contextos

Se construye un vector denso para cada palabra, procurando que sea similar a los vectores de palabras. Estos vectores son una representación distribuida. Sin embargo estos pueden representarse de varias formas. En [22, 25] se muestran los métodos más usados: *Word2vec y GloVe* (ver figura 2.11).

	0,286
	0,792
	-0,177
banco =	-0,107
	0,109
	-0,542
	0,349
	0,271

Figura 2.11: Vector de palabra de "banco"

2.4. Transferencia de conocimiento

Aunque *deep learning* ha tenido enorme éxito en problemas de aprendizaje supervisado, por ejemplo en clasificación de imágenes, reconocimiento de voz y jugar videojuegos, esos modelos tienen un alto grado de especialización en una sola tarea para la que fueron entrenados. Además, requieren millones de datos para ello.

El aprendizaje profundo empieza desde cero, con una inicialización aleatoria, cuando quiere aprender una nueva tarea. Haciendo una analogía con los humanos, es como si se pretendiera que un bebé se convierta en campeón de ajedrez sin antes siquiera saber como mover una pieza de ajedrez.

Para hacer frente a los problemas antes mencionados surge la *tranferencia de conocimiento* (*transfer learning*), el conocimiento generado por un modelo entrenado en otra tarea, como se aprecia en la figura 2.12. Para ello desarrolla algoritmos que puedan aprender nuevas tareas rápidamente [37]. Es decir, aprovecha la experiencia previa para aprender nuevas tareas con pocos ejemplos.

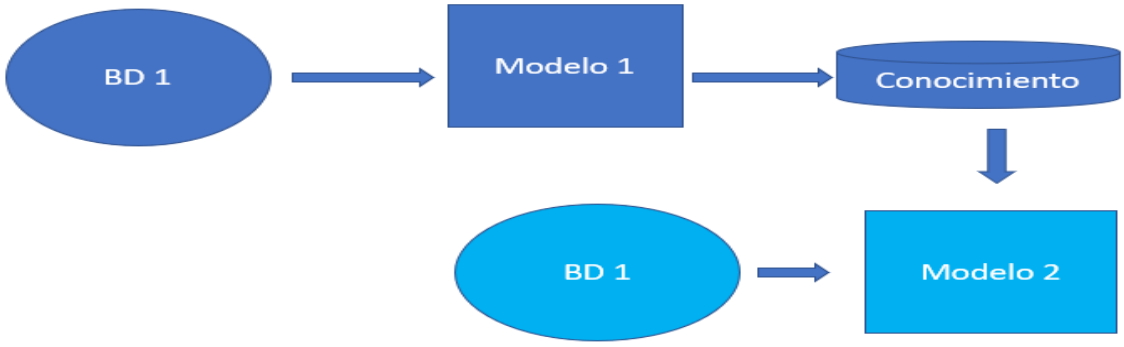

Figura 2.12: Transferencia de conocimiento

La transferencia de conocimiento es un subcampo dentro del aprendizaje automático que existe desde hace mucho tiempo, Caruana [4]. Estudia la habilidad para aprovechar conjuntos de datos anteriores cuando se quiere aprender de nuevos datos. Posiblemente, una de las más grandes historias de éxito de transferencia de conocimiento es la técnica de ***pre-entrenamiento*** en grandes cantidades de datos previamente disponibles y después ajustar finamente (***fine-tuning***) el modelo pre-entrenado en datos provenientes de nuevas tareas [10], aprende con menos ejemplos. Esta técnica ha tenido mucho éxito para entrenar CNN con pre-entrenamiento supervisado en ImageNet [30], y recientemente para entrenar modelos de lenguaje [9, 14] en enormes corpus.

El inconveniente es que diferentes tareas necesitan compartir alguna estructura. Afortunadamente hay muchas tareas que comparten su estructura; está estructura se da en las primeras capas de la red, a las que llaman ***generales*** [17, 37], Yosinski et al. descubrieron que la especialización se da en las últimas capas, a las que llaman ***específicas***. Si las características de la primera capa son generales y las de las últimas son específicas, entonces existe una ***transición*** de lo general a lo específico en algún punto de la red.

En la transferencia de conocimiento [4], primero entrenamos la base de una red en un *dataset* y tarea específica, luego reutilizamos las características aprendidas, las transferimos, a una segunda red para que sea entrenada en otra tarea y *dataset*.

La transferencia de conocimiento consiste, según Schmidhuber, en aprovechar los pesos de una red ya entrenada y ajustarlos (*fine tuning*) para resolver otro tipo de tareas con muy pocos ejemplos [33, 36].

Los tipos de estrategia para realizar la transferencia de conocimiento en un nuevo *dataset* son [4]:

1. **Extractor de pesos fijo**. Consiste en remover la última capa densa de la red y extraer las características de la red para nuevas tareas. En la figura 2.13 se observa como se elimina a partir de la capa fc2 y se aplican sus pesos para reentrenar un clasificador, en este caso un SVM.

[4]http://cs231n.github.io/transfer-learning/. Fecha de consulta: de 12 de octubre de 2019

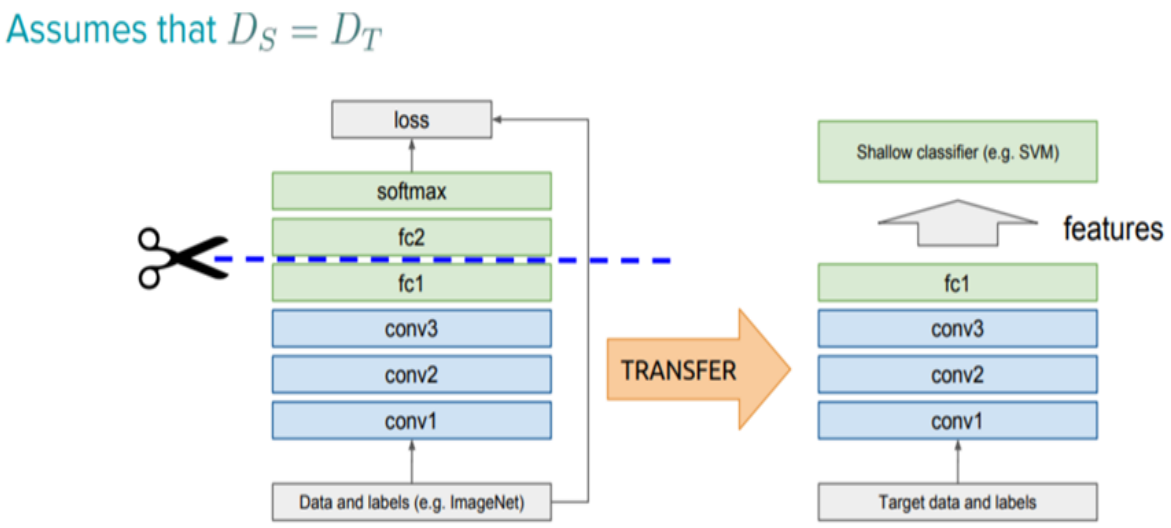

Figura 2.13: Transferencia de conocimiento utilizando un extractor de pesos. Imagen tomada de *https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a*

2. **Ajuste fino**. En esta estrategia se ajustan los pesos en la red original en capas seleccionadas a través de diversos entrenamientos, por ejemplo *backpropagation*. Existen dos formas de realizarlo, una se hace **congelando** las capas y por lo tanto congelando los pesos de las mismas y la otra es **ajustando** los hiperparámetros a partir de una capa elegida y por ende actualizando sus pesos. En la figura 2.14 se aprecia como se eligió la última capa para ajustarla con nuestra propia capa my_fc2 y con una función softmax.

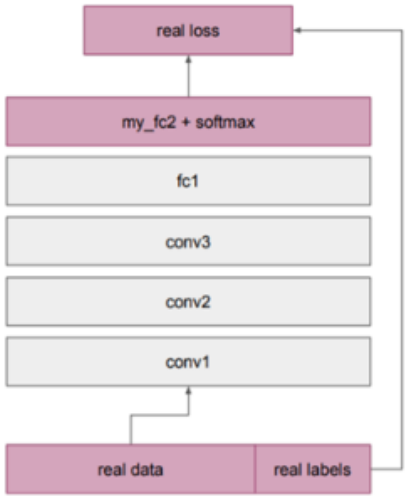

Figura 2.14: Transferencia de conocimiento por ajuste. Imagen tomada de *https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a*

3. **Modelos pre-entrenados**. Consiste en aprovechar los pesos de la red entrenados por otras personas. Por ejemplo en visión computacional se tiene inception V3, RestNet 50. En el procesamiento del lenguaje natural se tiene USE, FaxText, GloVe y word2Vec.

Para saber qué tipo de transferencia de conocimiento se tiene que hacer se toman en cuenta los siguientes criterios [4]:

- *El nuevo dataset es pequeño y similar al dataset original.* Como se tiene pocos datos en la nueva tarea se corre el riesgo de que haya sobreajuste, por lo que es mejor reentrenar el clasificador.
- *El nuevo dataset es grande y similar al dataset original.* A diferencia del supuesto anterior, como se tienen muchos datos el riesgo de sobreajuste disminuye, por lo que se puede aplicar el ajuste congelando algunas de la capas para aprovechar sus pesos.
- *El dataset es pequeño pero muy diferente al dataset original.* Como se tienen pocos datos y son diferentes al original lo mejor es reentrenar el clasificador en las primeras capas, que son las generales, pues si se entrena en las últimas capas se tendrán características diferentes a las de nuestra tarea.
- *El nuevo dataset es grande y muy diferente al original.* Gracias a que se tiene muchos datos no se corre el riesgo de sobreajuste, por lo que se puede ajustar cualquier capa de la red original actualizado sus pesos.

De acuerdo con [9] existen dos estrategias para la transferencia de conocimiento para texto:

(a) **Basado en características**: consiste tener vectores entrenados en otras tareas que capturan el contexto adicional de las palabras. Se obtienen nuevos vectores por cada capa que son usados después como características, concatenados con los vectores de palabras o con las capas intermedias, ejemplo de ello es el trabajo presentado en [26].

(b) **Ajuste fino**: consiste en entrenar un modelo de lenguaje en alguna arquitectura de red para captar el contexto de las palabras, para después ajustar esta arquitectura para otra tarea. Se ajustan un mínimo de hiperparámetros de la nueva tarea y se entrena afinando los hiperparámetros de la arquitectura original. Un ejemplo el trabajo de Howard et al [17].

Debido a que para nuestros experimentos tenemos una *dataset* pequeño decidimos aplicar la estrategia de modelos pre-entrenados y utilizar el método descrito en la Sección 2.5.

2.5. Universal Sentence Encoder

En esta sección se describe a detalle el algoritmo utilizado para extraer características basadas en transferencia de conocimiento y que posteriormente se utilizaran para realizar el perfilado de autor. El algoritmo que utilizaremos en este trabajo es el *Universal Sentence Encoder* (USE) [5]. Aunque este método no está diseñando para realizar perfilado de autor, posee ciertas características que pueden ser utilizadas para ésta tarea. El marco de trabajo presentado más adelante en el capítulo 4 adapta USE, entrenándolo con un clasificador lineal.

Universal Sentence Encoder codifica el texto en vectores de alta dimensión para que se puedan usar para la clasificación de textos, la similitud semántica, el agrupamiento y otras

tareas de lenguaje natural. El modelo es entrenado y optimizado para texto de una longitud mayor que la palabra, como oraciones, frases o párrafos cortos. Está entrenado en una variedad de fuentes de datos y una variedad de tareas con el objetivo de acomodar dinámicamente una amplia variedad de tareas de comprensión del lenguaje natural.

En concreto, USE tiene dos modelos para codificar documentos en vectores de palabras, uno hace uso de la arquitectura basada en promedios *Deep Averanging Network (DAN)* [18], mientras el otro se basa en una red neuronal convolucional para clasificación de documentos [19]. A continuación se detallan las arquitecturas del método propuesto para una mejor comprensión.

2.5.1. Red profunda de promedios

Esta arquitectura funciona en tres simples pasos [18]. En la figura 2.15 se puede apreciar como las C_i se promedian en un sólo vector que se ingresa a la red, de ahí su nombre de red profunda por promedios. A continuación se describe el proceso:

1. Promedia los vectores asociados a una secuencia de *tokens*.
2. Pasa el promedio a través de una o más capas de una red densa.
3. Realiza la clasificación lineal en la última capa.

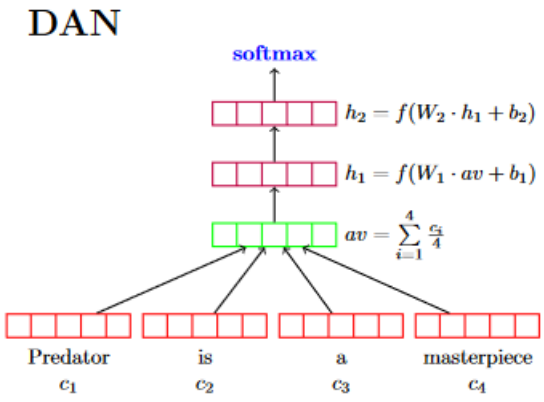

Figura 2.15: Red profunda de promedios. Imagen tomada de [18]

2.5.2. Red neuronal convolucional para clasificación de documentos

Este tipo de red toma una oración, convierte cada palabra en un vector denso y los ingresa en forma secuencial a la red para convertirlos en un sólo vector de longitud fija. Lo anterior lo consigue aplicando un muestreo promedio, esto es, los vectores finales de las oraciones se obtienen después de promediar los vectores de entradas a través de una o más capas de un red densa. En la figura 2.16 se aprecia como las palabras se ingresan a la red, después son pasadas a través de varias capas convolucionales para después promediarlas y obtener un vector que se pasa a través de una red densa.

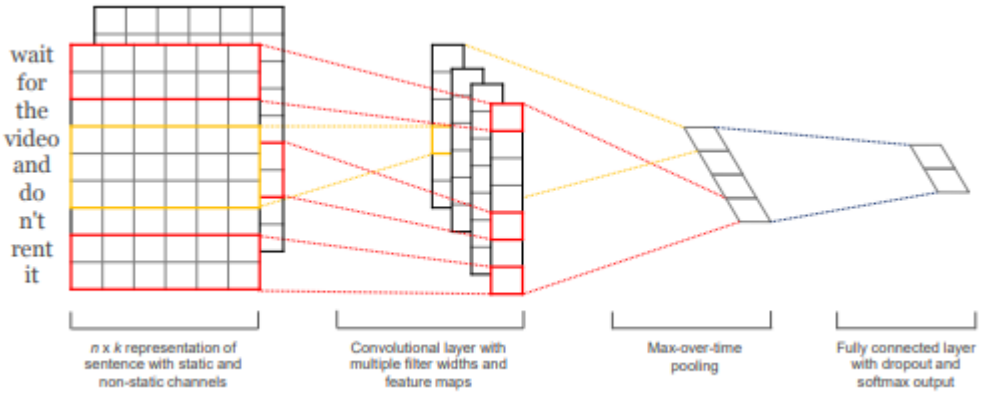

Figura 2.16: Red convolucional para clasificación de documentos. Imagen tomada de [19]

2.6. Resumen

En este capítulo se expusieron la terminología y conceptos relacionados con la transferencia de conocimiento. Además se presentaron los conceptos y modelos relacionados con este trabajo para mejorar la comprensión del desarrollo posterior de esta tesis.

Capítulo 3

Estado del arte

> "If I have seen further it is by standing on the sholders of Giants."
>
> *Sir Isaac Newton , Letter to Robert Hooke (15 February 1676)*

La tarea de perfilado de autor es un campo de constante desarrollo que ha tenido gran impacto en diversas áreas, por ejemplo en seguridad. En la actualidad existen grandes repositorios de datos donde se puede aplicar el perfilado. Para poder aplicar ésta tarea es necesario contar con herramientas automáticas de análisis y procesamiento de textos. Por lo que se realizan investigaciones orientadas a crear y mejorar técnicas usadas en el perfilado de autor.

Cuando se realiza el perfilado de autor en textos provenientes de las redes sociales, se enfrentan a características propias que no pueden compararse con los textos literarios, ensayos o cualquier otro tipo de texto formal. Esto es debido a la rapidez con se produce la comunicación en estas redes y a la libertad que se tiene para publicar contenido que no es revisado.

En este capítulo se presenta un panorama global del estado del arte en el área de perfilado de autor. También se describen las etapas de preprocesamiento, extracción de características y clasificadores para tener un mejor entendimiento del presente.

A continuación se realiza un análisis de algunos trabajos, que han servido de base y fundamento para la presente tesis.

3.1. Técnicas de preprocesamiento para perfilado de autor

Debido a que el lenguaje humano es diferente al lenguaje de las computadoras es necesario transformar el primero para que el segundo pueda entenderlo. Es decir, los datos no están condicionados para cumplir los objetivos de la tarea de aprendizaje.Lo anterior da lugar a la necesidad de preparar los textos para que la computadora pueda trabajar con ellos. Para poder realizarlo es necesario preprocesar la información para realizar la tarea de PA eficazmente. A continuación describiré las técnicas más importantes que se utilizan para tal propósito.

En la etapa de preprocesamiento de textos, estos se transforman a algún tipo de representación estructurada o semi-estructurada que facilite su posterior análisis. Su objetivo es eliminar información que no ayuda al proceso del perfilado de autor, realizándose alguna de las siguientes técnicas: análisis léxico, tratamiento y separación de palabras vacías (artículos,

preposiciones, conjunciones), términos relacionados morfológicamente, variaciones de género, número o tiempo verbal, palabras compuestas, normalización de palabras y etiquetas de palabras. Asimismo, se corrigen algunos problemas que presentan los documentos como: problemas de formato, polisemia, homonimia y sinonimia. [34]. También tiene como objetivo preparar a los textos para que puedan ser procesadas por lo

Esta fase se basa principalmente en la lingüística computacional, a través del análisis morfológico y sintáctico). Su principal objetivo es facilitar la selección de características deseadas para identificar patrones claves en la tarea del perfilado de autor.

Las implementaciones fueron realizadas en Python(2.7+) y utilizan la biblioteca de Python Scikit-learn [1] para la parte de los algoritmos de clasificación. Para la parte de preprocesamiento se utiliza la biblioteca *Natural Language Toolkit* (NLTK), para hacer el *stemming*, *tokenización*, etiquetado gramatical (POS) y *lematización*.

Varios autores limpian el *tweet* para obtener un texto plano, en [3] (Arroju et al.) extraen los *tweets* que pertenecen a un usuario y *tokenizan* sus palabras, remueven las etiqueta html, signos de puntuación, menciones de usuario, *hashtags* y *emoticones*. En cambio en Daneshvar et al. [8] además utilizan *TweetTokenizer* para reemplazar los caracteres que representan un salto de línea con un salto de línea, concatenan todos los *tweets* de cada uno de los autores en una sola cadena de caracteres, con una etiqueta *<EndOfTweet>* al final de cada uno de ellos, después pasan todas las palabras a minúsculas, cortan las caracteres repetidos, reemplazan las URLs, @username menciones con una etiqueta, remueven las *stopwords* y cualquier n-grama que ocurra en todos los documentos que se considere *stopword* es ignorada. Por otra parte en Martinc et al. [21] los reemplazan con un específico *placeholder* #HASHTAG, @MENCION, HTTP, URL, respectivamente, después aplican etiquetas POS.

Otro enfoque lo propone en [24] (Modaresi et al.), preprocesan cada texto quitando la mayoría de la información que puede ser de un genero específico por ejemplo las URLs. También utilizan los *hashtags* para categorizar los *tweets*.

La tabla 3.1 presenta los resultados del preprocesamiento usual para la identificación de género con la metodología tradicional.

Autor|	*Stopword*	*Emoticones*	*Emojis*	Menciones	*Hastags*	URLs	*Slangs*	Puntuación y números
Arroju et al.	X	X		X	X			X
Danveshvar et al.	X			X		X		X
MartinC et al.	X			X	X			X
Moderasi et al.	X			X	X	X		X

Tabla 3.1: Comparación de las técnicas de preprocesamiento del estado del arte para identificación de género

3.2. Extracción de características para perfilado de autor

En esta etapa se realiza la extracción y análisis de características, con el fin de encontrar estructuras existentes. Durante esta etapa los textos se pueden representar a través del modelo de espacio vectorial [31], donde cada texto es modelado como un vector de dimensión n y representado por el vector $D_i = (d_{i1}, d_{i2}...d_{in})$, cada d_{ij} representa el número de repeticiones de términos en el documento i. Éste modelo busca mapear cada término de cada texto en un índice, lo que permite contabilizar los términos del documento.

[1] https://scikit-learn.org/stable/

La indexación se puede aplicar a diferente tipos de extracciones de un texto dado, se puede hacer a nivel carácter, n-gramas de caracteres, o a nivel palabra, n-gramas de palabras. Está técnica permite construir un histograma a partir de un texto donde se ven reflejada las repeticiones de cada índice en un documento, conocida como frecuencia de término. Los datos obtenidos en esta etapa son representados en bolsa de palabras o n-grama de caracteres.

Debido a la gran cantidad de información textual que se genera a diario necesitamos métodos que extraigan patrones útiles a partir de la información utilizada, para tal tarea ocupamos las características lingüísticas para realizar el perfilado de autor. Las más usuales son los n-gramas, n-gramas sintácticos y la bolsa de palabras.

El trabajo de Daneshvar et al. [8] fue presentado en el concurso PAN 2018. Los participantes tuvieron que predecir el género de los usuarios de *Twitter* en tres idiomas diferentes (inglés, español y árabe). El enfoque tiene en cuenta las características estilísticas representadas por n-gramas de caracteres y de palabras.

Para el idioma inglés ocuparon bolsa de palabras, bigramas y trigramas de palabras. Para el español y árabe utilizaron bolsa de palabras y bigramas, y en los tres casos utilizaron 3-5-gramas de caracteres, obteniendo su mejor resultado para el idioma inglés con un 82.21 %.

Argamon et al. [2] demostraron la variación de estilos de escritura entre hombres y mujeres en un corpus en inglés obtenido de periódicos y libros. Ellos encontraron que los pronombres (*I, you,she, her, their, myself, yourself, herself*) son fuertes indicadores del género femenino. y que los determinantes (*a, the, that, these) y los cuantificadores (one, two, more, some*) son indicadores del género masculino.

Scheler et al. [32] obtuvieron una precisión del 80 % en la clasificación de género en un corpus de 71,000 blogs de blogger.com. Ellos utilizaron algunas características estilísticas, incluyendo POS e *hyperlinks*, junto con 1,000 unigramas. Ellos reportaron que los *bloggers* másculinos son sobre política, tecnología y dinero, mientras que las mujeres comparten más sobre sus vidas personales.

En [16] (Gómez) se realiza un profundo estudio sobre la extracción de características lingüísticas, y se proponen métodos basados en grafos sintácticos integrados. Gómez propone como primer métodos la extracción de unigramas para encontrar características en el grafo a través del conteo de lemas de palabras y etiquetas POS a medida que se recorre el grafo. El otro método que propone es la extracción de n-gramas sintácticos, a diferencia del método anterior, se construyen n-gramas sintácticos a partir del recorrido de caminos cortos en ambos grafos.

Otro enfoque es el propuesto por Rangel, quien propone *EmoGraph* que modela el modo en que las emociones de los autores son expresadas para obtener información sobre su importancia relativa por su posición con y en relación con el resto de los elementos del discurso. Es decir, construye un grafo con las diferentes categorías morfosintácticas del texto y le agrega información semántica cómo las emociones y sentimientos que expresa. Con esto obtiene los pesos relativos a cada una de las características utilizadas [29].

La tabla 3.2 presenta los resultados de la extracción de características para la identificación de género con la metodología tradicional.

Autor	N-grama palabras	N-grama de caracteres	N-grama sintáctico	Características estilística	POS	BoW	Grafos
Danveshvar et al.	X	X				X	
Argamon et al.				X			
Scheler				X	X	X	
Gómez			X				X
Rangel			X				X

Tabla 3.2: Comparación de las técnicas de extracción de características del estado del arte para identificación de género

3.3. Clasificadores utilizados en perfilado de autor

En el aprendizaje supervisado, los algoritmos trabajan con datos etiquetados para producir una función a partir de datos de entrenamiento que, dada los datos de entrada, les asigne una etiqueta de salida que representa la clase a la que pertenece. Se entrena el modelo en una conjunto de entrenamiento para después probarlo en un conjunto de prueba y ver que efectivamente predice las etiquetas correctas. A continuación describiré los métodos más importantes que se utilizan para la clasificación del perfilado de autor.

En [12] (Posadas et al.) proponen tratar al PA como una problema de clasificación multietiqueta donde una instancia es asociada a varias etiquetas. Utilizan máquinas de vectores de soporte para cada una de las etiquetas, en donde la predicción del perfilado de autor es la intersección de cada uno de los clasificadores. Para ello utilizan modelos de espacio vectorial enfocados en n-gramas sintácticos.

En Aragon et al. [1] clasifican *tweets*, para ello seleccionan los mejores n-gramas, de tamaño 2 a 5, utilizando la distribución chi cuadrada para cada grupo de n-gramas y los concatenan. Después utilizan máquinas de vectores de soporte para el entrenamiento y validación.

Respecto a los enfoques del aprendizaje profundo, los autores que obtuvieron las puntuaciones más altas en el concurso PAN 2018 fueron *Takahashi et al.*, usando redes neuronales recurrentes. Su metodología consistió en utilizar como valores de entrada vectores de palabras para codificar los *tweets* de usuarios en representaciones de textos, integrando los 100 *tweets* de cada usuario en una representación. Para ello pasaban los vectores de palabras a través de una red neuronal recurrente, que manejaba cada oración palabra por palabra en el tiempo t, con estados bidireccionales GRU, después lo pasan a un capa de muestreo por promedios por palabras en *tweets*, y éste a su vez es pasado por una capa de muestreo por promedio de *tweets* de un usuario [35]. Esta arquitectura se puede ver en la figura 3.1.

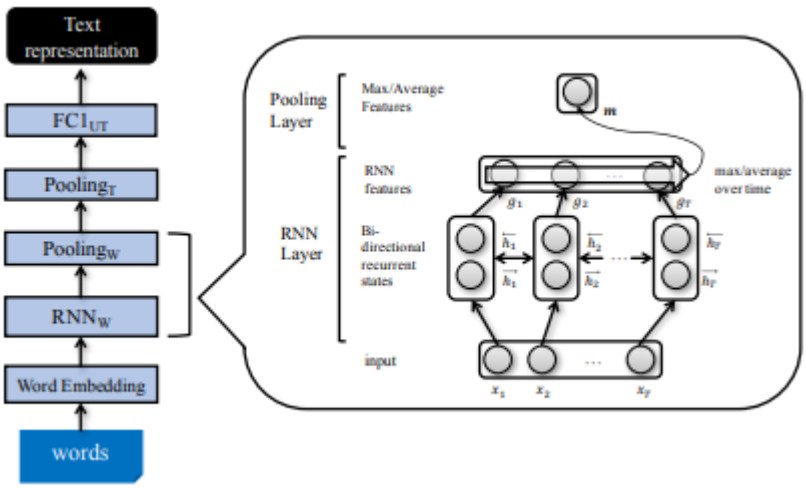

Figura 3.1: Descripción del componente de texto para perfilado de autor. Imagen tomada de [35]

La tabla 3.3 presenta los resultados de los algoritmos de aprendizaje usados para la identificación de género con la metodología tradicional.

Autor	SVM	RL	Redes neuronales
Posadas et al.	X		
Aragon et al.	X		
Markov et al.	X	X	
Takanashi et al.			X

Tabla 3.3: Comparación de los algoritmos de aprendizaje para clasificación del estado del arte para la identificación de género

3.4. Resumen

En esta capítulo se resumen las técnicas y métodos que han sido desarrollados hasta el momento para realizar el perfilado de autor. En especial aquellos relacionados con el preprocesamiento, extracción de características y clasificación.

Capítulo 4

Metodología propuesta

En este capítulo se describe el marco metodológico de este trabajo de tesis que busca encontrar el perfil del autor de un documento de origen desconocido. También se resumen y analizan los resultados obtenidos a partir de dichos experimentos.

Se describen dos métodos para realizar dicha tarea: el sistema base y nuestro método propuesto que utiliza la transferencia de conocimiento. En este trabajo de tesis desarrollamos un método que utiliza el algoritmo *Universal Sentence Encoder* (USE) [5] para obtener vectores de de características que representen a los *tweets* y utilizarlos como características para entrenar un modelo de PA. Como se explicó en el capítulo 2, el algoritmo USE no fue pensando *ex profeso* para realizar PA. Sin embargo, permite convertir documentos completos en representaciones vectoriales, las cuales capturan información semántica que se puede utilizar para entrenar algoritmos para una amplia gama de problemas de clasificación.

4.1. Perfilado de autor utilizando máquinas de vectores de soporte y regresión logística

En la etapa inicial se reproducirá un sistema base . Al final, compararemos los resultados obtenidos por nuestra metodología propuesta con los resultados obtenidos por el sistema base. Éste se divide en tres secciones: preprocesamiento, extracción de características y clasificación. Se evalúan con validación cruzada estratificada, repetida 5 veces.

4.1.1. Preprocesamiento

En primer lugar se procede a transformar todos los archivos del corpus que contienen los *tweets* del formato XML a *txt* para facilitar el procesamiento de los mismos. Después, realizamos el preprocesamiento del corpus PAN 2017 quitando los caracteres especiales, las *stopwords* y pasando todas las palabras a minúsculas utilizando las técnicas descritas en el capítulo 3 .

Adicionalmente, realizamos distintos tipos de preprocesamiento sobre los *tweets* para remover características asociadas con el dominio de esta red social como lo son las menciones (@user), etiquetas (#hashtags), *links* (urls), *emoticones*, *emojis* y *slangs*.

4.1.2. Extracción de características

Posteriormente realizamos la extracción de características sobre el corpus preprocesado. Tradicionalmente las características más relevantes para realizar perfilado de autor son las

bolsa de palabras y los N-gramas de caracteres. Por lo tanto, utilizamos estas características para el sistema base.

Sin embargo, estas características no se pueden alimentar directamente a los algoritmos de clasificación, ya que la mayoría de ellos esperan vectores de características numéricas con un tamaño fijo en lugar de los documentos de texto sin formato con longitud variable.

Para resolver este problema utilizamos el proceso de vectorización. Primero se separa cada palabra mediante un proceso de tokenización, luego contamos la frecuencia de cada una de las palabras en el texto. Esta estrategia es lo que se denomina representación de la bolsa de palabras. Los documentos se describen por las ocurrencias de palabras mientras se ignora por completo la información de posición relativa de las palabras en el documento. Con el objetivo de tomar en cuenta la posición de las palabras en el documento, utilizamos también los 3-gramas de caracteres. De igual manera, se obtienen secuencias de 3 caracteres contiguos en el texto y se cuentan las ocurrencias totales.

Para realizar este proceso de extracción de características y vectorización utilizamos la biblioteca Scikit-learn[1] de Python.

4.1.3. Clasificación

Con las características obtenidas entrenamos un algoritmo de clasificación para generar un modelo de perfilado de autor que pueda identificar el género de la persona que escribió un *tweet* y su variación del lenguaje. Para realizar tal tarea utilizamos el algoritmo Máquinas de Vectores de Soporte (SVM) y el de regresión logística, implementados en la biblioteca Scikit-learn[1] de Python.

Finalmente, evaluamos el modelo entrenado con datos que no estén el conjunto de entrenamiento utilizando validación cruzada para verificar el rendimiento de este modelo. Es decir, dividimos el corpus en 5 partes y realizamos 5 iteraciones. En cada iteración utilizamos 4 partes para entrenar y 1 parte para evaluar, para esto utilizaremos el método *RepeatedStratifiedKFold* de Scikit-learn.

4.2. Perfilado de autor utilizando transferencia de conocimiento

En capítulos anteriores hemos descrito a detalle USE, los métodos de clasificación y las redes neuronales. A continuación detallamos cómo utilizamos USE para obtener características de los usuarios de *Twitter* con base en muestras de *tweets* y mediante las cuales se determinará el perfil de cada uno de los usuarios.

USE está compuesto de una red neuronal convolucional que se entrenó inicialmente para resolver problemas de similitud textual y clasificación de textos, y genera un vector de salida de 512 dimensiones. Ocuparemos la transferencia de conocimiento para trasladar el conocimiento adquirido por esta red en las tareas antes mencionadas y utilizarlo para el perfilado de autor.

Para este trabajo de tesis utilizaremos un modelo de USE entrenado en múltiples tareas a través de 16 lenguajes, incluyendo el Español. USE recibe como entrada un texto de longitud variable en cualquiera de los idiomas en el que fue entrenado y la salida es un vector de 512 dimensiones. El modelo de USE que utilizamos está disponible en la página de *TensorFlowHub*[2] y puede ser descargado libremente. Además de este modelo, existen varias versiones de modelos de USE entrenados con diferentes objetivos, incluidos el tamaño/rendimiento multilingüe y la recuperación de preguntas y respuestas.

[1] https://scikit-learn.org/

[2] https://tfhub.dev/google/universal-sentence-encoder-multilingual/1

En nuestra propuesta para el perfilado de autor, nosotros pasamos a USE los 100 *tweets* de cada usuario, de esta manera la red convolucional los transformará en un vector de 512 dimensiones, utilizando el modelo del lenguaje que ya tenía aprendido y actualizando con las nuevas muestras textuales de *Twitter*.

Dado que el lenguaje de redes sociales es bastante diferente al lenguaje formal en el cual fue entrenado originalmente USE, nosotros evaluaremos diferentes enfoques para hacer el ***pre-procesamiento***. A continuación definimos las diferentes estrategias de pre-procesamiento:

1. **Eliminación de *emoticones*:** Se hará la clasificación quitando los *emoticones* de los textos. Utilizamos una lista de *emoticones* recolectadas en [27] y mediante expresiones regulares se identifican en los tweets cuando aparecen dichos emoticones y se reemplazan por un espacio vacío.

2. **Eliminación de *emojis*:** Se hará la clasificación quitando los *emojis* de los textos. Utilizamos una lista de *emojis* recolectadas en [27] y mediante expresiones regulares se identifican en los tweets cuando aparecen dichos *emojis* y se reemplazan por un espacio vacío.

3. **Eliminación de menciones:** Se hará la clasificación quitando los menciones(@usuario) de los *tweets*. Mediante expresiones regulares se identifican en los tweets cuando aparecen dichas menciones a usuarios y se reemplazan por un espacio vacío.

4. **Eliminación de URLs:** Se hará la clasificación quitando los URLs que comparten los usuarios en los *tweets*. Mediante expresiones regulares se identifican en los tweets cuando aparecen dichos *links* y se reemplazan por un espacio vacío.

5. **Eliminación de *hashtags*:** Se hará la clasificación quitando los *hastags* (#tópico) de los textos. Mediante expresiones regulares se identifican en los tweets cuando aparecen dichas etiquetas y se reemplazan por un espacio vacío.

6. **Reemplazo de *slangs*:** Se hará la clasificación reemplazando los *slangs* de los *tweets* por su versión normalizada. Utilizamos una lista de *slangs* recolectada en [27] y mediante expresiones regulares se identifican en los tweets cuando aparecen dichos *slangs* y se reemplazan por la palabra o frase a la que hace referencia.

7. **Todos los preprocesamientos:** Por último, se hará la clasificación realizando todos los preprocesamientos mencionados previamente.

Finalmente se entrenará el algoritmo de clasificación para obtener un modelo que identifique el perfil de usuarios de *Twitter*. Los algoritmos de clasificación que utilizaremos son las Máquinas de Vectores de Soporte y Regresión Logística. Generaremos varios modelos de clasificación con cada uno de los algoritmos para cada uno de los tipos de preprocesamiento.

Para la implementación de los algoritmos de aprendizaje utilizaremos el módulo *Scikit-learn* de *Python*. En el caso de la regresión logística se hará con el módulo *LogisticRegression*[3], que implementa regresión logística estandarizada, con regularización *L2*, con un algoritmo de optimización quasi-Newton de funciones con un gran número de parámetros (lbfgs).

Para el algoritmo Máquinas de Vectores de Soporte se utilizará el módulo *SVC* [4] con un kernel de base radial (*RBF*) y con hiperparámetros por defecto.

[3]https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html
[4]https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

4.3. Métricas de evaluación de modelos

Las métricas de evaluación miden el desempeño de un modelo, algo importante del aprendizaje automático para comprender la calidad del modelo y para ajustar los hiperparámetros en el proceso para seleccionar el modelo o técnica más aceptable. La elección de la métrica depende del tipo de modelo y el plan de implementación del mismo. La evaluación de los modelos de aprendizaje se realiza con el siguiente esquema:

- Se utilizará **RepeatedStratifiedKFold**, para aplicar validación cruzada en el conjunto de datos de entrenamiento. Se harán 5 particiones y repetirlas 5 veces cada una con diferentes particiones aleatorias. Éste método realiza un muestreo estratificado para producir particiones que contengan una razón representativa de cada clase. En cada iteración el código genera un clon del clasificador, entrena el clon en la partición de entrenamiento, y hace predicciones en la partición de prueba. Después cuenta el número correcto de predicciones y los resultados de la razón de predicciones correctas [11].

- Luego se utilizarán las métricas de evaluación: F1, precisión, exactitud y exhaustividad para cada iteración. Se obtendrá el promedio de todas las evaluación y esto se presentará como resultado final en el capítulo 5.

A continuación se explican a detalle las métricas de evaluación utilizadas:

4.3.1. Exactitud

Es una métrica para evaluar modelos de clasificación y se define como el cociente del número total de predicciones correctas sobre el total de predicciones. Se usa cuando el número de elementos de cada clase está bien balanceado.

$$Exactitud = \frac{número\ de\ predicciones\ correctas}{número\ total\ de\ predicciones} \tag{4.1}$$

4.3.2. Precisión

Está métrica evalúa el rendimiento de un modelo de clasificación con el cociente de las predicciones positivas que realizó nuestro modelo sobre la suma de las predicciones correctas más las incorrectas. Por ejemplo, si Se usa cuando el número de elementos de cada clase esta desbalanceado y existe una gran disparidad entre el número de etiquetas positivas y negativas en clasificación binaria.

Para la clasificación binaria:

$$Precisión = \frac{VP}{VP\ + FP} \tag{4.2}$$

Para la clasificación multiclase se obtiene el promedio de la precisión.

$$Precisión = \frac{1}{n}\sum_{i=0}^{n}(\frac{VP}{VP\ + FP}) \tag{4.3}$$

4.3.3. Exhaustividad

También se le conoce como tasa de verdaderos positivos. Evalúa el rendimiento de un modelo de clasificación a través de un cociente entre los verdaderos positivos sobre el total de los mismo, más los falsos negativos (llamados elementos relevantes). Responde a la pregunta: ¿Dada una clase, qué tan probable es que el modelo la detecte correctamente?

Para la clasificación binaria:

$$Exhaustividad = \frac{VP}{VP\ +FN} \tag{4.4}$$

Para la clasificación multiclase se obtiene el promedio de la precisión.

$$Exhaustividad = \frac{1}{n}\sum_{i=0}^{n}(\frac{VP}{VP\ +FN}) \tag{4.5}$$

4.3.4. Valor F1

Es una métrica ampliamente utilizada en clases desbalanceadas y combina los dos conceptos anteriores para evaluar el rendimiento de un modelo de clasificación. El valor F1 es un promedio armónico entre las métricas de precisión y exhaustividad. Mientras que el promedio regular trata a todos los valores de la misma forma, el promedio armónico da mucho más peso a los valores bajos. Como consecuencia, el clasificador sólo tendrá un valor F1 alto, si la precisión y exhaustividad son también altos [11].

$$F_1 = 2 \times \frac{precisión\ X\ exhaustividad}{precisión\ +\ exhaustividad} = \frac{2}{\frac{1}{precisión} + \frac{1}{exhaustividad}} = \frac{TP}{TP + \frac{FN+FP}{2}} \tag{4.6}$$

4.4. Resumen

En este capítulo se describió la metodología propuesta para realizar el perfilado de autor de *tweets* en español utilizando USE como transferencia de conocimiento.

Se describieron los tipos de preprocesamiento propuestos para conocer cuál de todos influye más en la tarea de PA. Se probaron los dos modelos, el sistema base, y nuestra propuesta. Se utilizó los algoritmos de regresión logística y máquinas de vectores de soporte para la clasificación. Se utilizó la validación cruzada estratificada repetida para el entrenamiento y prueba de los modelos.

Por último se describieron las métricas F1, precisión, exhaustividad y exactitud.

Capítulo 5

Resultados experimentales

En este capítulo se mostrarán y discutirán los resultados obtenidos en la tarea de perfilado de autor. En la primera parte se describen las características del corpus utilizado, después, en la segunda parte se presentan los resultados del *baseline* y por último se muestran los resultados con la metodología propuesta. De esta manera se busca verificar que las experimentaciones realizadas se cotejen con la hipótesis planteada. En especial, se describen los algoritmos usados para el proceso de clasificación de textos.

Para mostrar los resultados se presentarán cuatro secciones donde se aprecia la combinación de los distintos tipos de características para cada algoritmo: Bolsa de palabras y N-gramas. Dependiendo del algoritmo de clasificación se presentan los resultados de experimentos con diferentes variantes de preprocesamiento: sin *emojis*, sin *slangs*, sin *emoticones*, sin *hashtags*, sin url, sin menciones y realizando todos los preprocesamientos.

5.1. Descripción del corpus

El corpus con el cual se trabajará es el de la competencia [28] (PAN 2017), el cual fue recopilado de *Twitter* en español. La información de género y edad la han proporcionado los propios usuarios en un cuestionario realizado *online*. Hay 600 usuarios de cada nacionalidad: mexicana, colombiana, peruana, argentina, chilena y venezolana. El 50 % es de hombres y el otro de mujeres. El corpus está balanceado por género y nacionalidad. Para los experimentos de este trabajo de tesis utilizaremos solamente las etiquetas de género.

Género	**Autores**	**Tweets**
Hombres	2,100	21k
Mujeres	2,100	21k
Total	**4,200**	**42,000**

Variación	**Autores**	**Tweets**
Colombia	600	6k
Argentina	600	6k
España	600	6k
Venezuela	600	6k
Perú	600	6k
Chile	600	6k
México	600	6k
Total	**4,200**	**42,000**

Tabla 5.1: Estadísticas de distribución de clases del corpus de Perfilado de autor del PAN 2017.

5.2. Resultados obtenidos con el método basado en regresión logística

En este primer experimento se consideran las características de BOW y N-gramas de caracteres. Se reproducen los resultados con el *baseline* y se prueban con todos los preprocesamientos propuestos. En particular, los resultados que se presentan se evalúan con las métricas F1, precisión, exhaustividad y exactitud.

Durante la primera parte de este experimento se evaluaron los diferentes preprocesamiento con las bolsas de palabras, la tabla 5.2 muestran los resultados para el algoritmo de clasificación de regresión logística. La columna *Características* indica cuál es la utilizada en el experimento, mientras que la columna *Dim.* indica la cantidad de características extraídas, y por tanto, la dimensionalidad del vector de características. Presentamos las medidas de evaluación de exactitud (con la STD, la desviación estándar de la exactitud), exhaustividad, precisión y valor F1. La columna *Preprocesamiento* indica cuál estrategia se siguió en cada experimento. Por ejemplo, *NINGUNO* indica que no se realizó preprocesamiento en ese experimento, *Sin Emojis* indica que se eliminaron los *emojis* y así sucesivamente con todos los experimentos.

Se observa que el preprocesamiento con el cual se obtienen los mejores resultados es cuando se quitan las menciones de usuarios, esto nos permite inferir que estas son las que aportan menos información con respecto al género de la persona que escribió el *tweet*.

Característica	Dim.	Exactitud	STD (Exactitud)	F-1	Precisión	Exhaustividad	Preprocesamiento
BoW	199114	0.6923	0.0287	0.6927	0.6947	0.6923	NINGUNO
BoW	199114	0.6926	0.0289	0.6930	0.6951	0.6926	Sin Emoticones
BoW	186798	0.6909	0.0318	0.6913	0.6934	0.6909	Sin Hashtags
BoW	**166070**	**0.6976**	**0.0299**	**0.6979**	**0.6994**	**0.6976**	**Sin Menciones**
BoW	43089	0.6090	0.0680	0.6409	0.7273	0.6090	Sin Slangs
BoW	136731	0,6926	0,0289	0,6930	0,6951	0,6926	Sin URLs
BoW	199226	0.6925	0.0288	0.6929	0.6950	0.6925	Sin Emojis
BoW	27137	0.6497	0.0533	0.6830	0.7836	0.6497	TODOS

Tabla 5.2: Resultados de los experimentos realizados para predecir género utilizando bolsa de palabras y el clasificador de regresión logística.

La tabla 5.3 presenta los resultados de la identificación de género utilizando como características los 3-gramas de caracteres, y como algoritmo de clasificación la regresión logística. Se observa que los mejores resultados se obtienen igualmente al quitar las menciones de usuarios, en cambio cuando se quitan los *slangs*, el rendimiento de los modelos baja considerablemente.

Característica	Dim.	Exactitud	STD (Exactitud)	F-1	Precisión	Exhaustividad	Preprocesamiento
N-char	2550956	0.6833	0.0248	0.6837	0.6858	0.6833	NINGUNO
N-char	2545770	0.6836	0.0243	0.6841	0.6862	0.6836	Sin Emoticones
N-char	**2309400**	**0.6874**	**0.0279**	**0.6878**	**0.6896**	**0.6874**	**Sin Menciones**
N-char	2470050	0.6811	0.0289	0.6815	0.6839	0.6811	Sin Hashtags
N-char	613817	0.5677	0.0537	0.6284	0.7685	0.5677	Sin Slangs
N-char	2324031	0.6817	0.0251	0.6821	0.6842	0.6817	Sin Emojis
N-char	1461457	0,6836	0,0243	0,6841	0,6862	0,6836	Sin URLs
N-char	276872	0.6240	0.0454	0.6717	0.7990	0.6240	TODOS

Tabla 5.3: Resultados de los experimentos realizados para predecir género utilizando N-gramas de caracteres y el clasificador de regresión logística.

5.3. Resultados obtenidos con el método basado en máquinas de vectores de soporte

En esta sección se muestran los resultados de la segunda parte de los experimentos utilizando como referencia el *baseline*. Para esta parte, se utilizó como algoritmo de clasificación las máquinas de vectores de soporte y se le aplicaron como características las bolsa de palabras y los N-gramas de caracteres.

Para estos experimentos también se aplicaron todos los preprocesamientos que se realizaron en el experimento anterior para ver cómo afectan a las características para realizar el perfilado de autor.

La tabla 5.4 presenta los resultados de la identificación de género utilizando como características bolsa de palabras, y como algoritmo de clasificación las Máquinas de Vectores de Soporte. La tabla presenta la misma estructura que la anterior. De igual manera, presentamos las medidas de evaluación de exactitud, exhaustividad, precisión y valor F1. Se observa que los mejores resultados se obtienen de quitar las menciones de usuarios y los peores cuando se quitan los *slangs*, con una diferencia entre ambos de aproximadamente 10 %.

Característica	Dim.	Exactitud	STD (Exactitud)	F-1	Precisión	Exhaustividad	Preprocesamiento
BoW	199114	0.6926	0.0294	0.6933	0.6964	0.6926	NINGUNO
BoW	199226	0.6925	0.0293	0.6931	0.6963	0.6925	Sin Emojis
BoW	186798	0.6933	0.0290	0.6939	0.6971	0.6933	Sin Hashtags
BoW	**166070**	**0.6981**	**0.0294**	**0.6986**	**0.7009**	**0.6981**	**Sin Menciones**
BoW	199114	0.6925	0.0296	0.6932	0.6963	0.6925	Sin Emoticones
BoW	43089	0.5939	0.0595	0.6292	0.7236	0.5939	Sin Slangs
BoW	136731	0.6925	0.0296	0.6932	0.6963	0.6925	Sin URLs
BoW	276872	0.6219	0.0391	0.6753	0.8157	0.6219	TODOS

Tabla 5.4: Resultados de los experimentos realizados para predecir género utilizando bolsa de palabras y el clasificador maquinas de vectores de soporte.

La tabla 5.5 presenta los resultados de la identificación de género utilizando como características 3-gramas de caracteres, y como algoritmo de clasificación las Máquinas de Vectores de Soporte. De igual manera, se observa que los mejores resultados se obtienen de quitar las menciones de usuarios y los peores resultados cuando se quitan los *slangs*. Sin embargo, para el caso de los 3-gramas de caracteres la diferencia de exactitud entre ambos es aproximadamente del 15 %.

Característica	Dim.	Exactitud	STD (Exactitud)	F-1	Precisión	Exhaustividad	Preprocesamiento
N-char	2550956	0.6817	0.0269	0.6824	0.6858	0.6817	NINGUNO
N-char	2545770	0.6822	0.0270	0.6830	0.6863	0.6822	Sin Emoticones
N-char	**2309400**	**0.6930**	**0.0296**	**0.6938**	**0.6975**	**0.6930**	**Sin Menciones**
N-char	2470050	0.6814	0.0261	0.6821	0.6856	0.6814	Sin Hashtags
N-char	2324031	0.6822	0.0270	0.6830	0.6863	0.6822	Sin Emojis
N-char	613817	0.5417	0.0314	0.6354	0.8286	0.5417	Sin Slangs
N-char	1461457	0,6822	0,0270	0,6830	0,6863	0,6822	Sin URLs
N-char	27137	0.6517	0.0547	0.6812	0.7723	0.6517	TODOS

Tabla 5.5: Resultados de los experimentos realizados para predecir género utilizando N-grama de caracteres y el clasificador máquinas de vectores de soporte.

5.4. Resultados obtenidos con el método basado en transferencia de conocimiento

La tabla 5.6 presenta los resultados de la identificación de género utilizando USE para obtener los vectores de características de 512 dimensiones por cada usuario, es decir, los 100 *tweets* se reducen a un vector de 512 dimensiones, y como algoritmo de clasificación la regresión logística. La estructura de la tabla es igual que las anteriores y en este caso la dimensionalidad del vector de características es siempre 512. Presentamos las medidas de evaluación de exactitud, exhaustividad, precisión y valor F1. Se observa que los mejores resultados en términos de exactitud se obtienen al quitar las menciones y los peores al reemplazar los *slangs*.

Característica	Dim.	Exactitud	STD (Exactitud)	F-1	Precisión	Exhaustividad	Preprocesamiento
USE	512	0.6986	0.0222	0.6989	0.6854	0.6998	NINGUNO
USE	512	0.6977	0.0263	0.6981	0.6808	0.7002	Sin Emojis
USE	512	0.6989	0.0213	0.6991	0.6854	0.7001	Sin Emoticones
USE	512	0.7041	0.0241	0.7042	0.6946	0.7036	Sin Hashtags
USE	512	**0.7156**	**0.0255**	**0.7158**	**0.6972**	**0.7198**	**Sin Menciones**
USE	512	0.6794	0.0459	0.6856	0.7745	0.6864	Sin Slangs
USE	512	0.7001	0.0265	0.7004	0.6895	0.7005	Sin URLs
USE	512	0.6864	0.0489	0.6900	0.7598	0.7029	TODOS

Tabla 5.6: Resultados de experimentos basados en transferencia de conocimiento bajo el clasificador de regresión logística para identificar género

La tabla 5.7 presenta los resultados de la identificación de género utilizando USE para obtener los vectores de palabras de 512 dimensiones y como algoritmo de clasificación las máquinas de vectores de soporte. Presentamos las medidas de evaluación de exactitud, exhaustividad, precisión y valor F1. De igual manera que con el clasificador anterior, se observa que los mejores resultados en términos de exactitud se obtienen al quitar las menciones y los peores al reemplazar los *slangs*. Si bien los resultados van de acuerdo con los obtenidos con características tradicionales en términos de mejor y peor preprocesamiento, podemos observar que con USE la diferencia entre ellos no supera el 3 %.

Característica	Dim.	Exactitud	STD (Exactitud)	F-1	Precisión	Exhaustividad	Preprocesamiento
USE	512	0.7068	0.0218	0.7072	0.6838	0.7124	NINGUNO
USE	512	0.7037	0.0281	0.7043	0.6705	0.7137	Sin Emojis
USE	512	0.7061	0.0213	0.7064	0.6833	0.7115	Sin Emoticones
USE	512	0.7080	0.0211	0.7084	0.6856	0.7134	Sin Hashtags
USE	512	**0.7198**	**0.0267**	**0.7201**	**0.6951**	**0.7270**	**Sin Menciones**
USE	512	0.6955	0.0408	0.6974	0.7340	0.7207	Sin Slangs
USE	512	0.7009	0.0277	0.7010	0.6987	0.6974	Sin URLs
USE	512	0.6992	0.0485	0.7020	0.7563	0.7201	TODOS

Tabla 5.7: Resultados de experimentos basados en transferencia de conocimiento bajo el clasificador máquinas de vectores de soporte para identifica género

5.5. Resumen

A partir de los resultados presentados en éste capítulo podemos resumir que nuestra metodología funciona mejor para la clasificación binaria cuando se quitan las menciones. Este

supuesto se obtiene a partir de los experimentos con el corpus del PAN 2017, donde obtuvimos los mejores resultados cuando identificamos el género de los autores de los *tweets*.

En este capítulo presentamos un compilado de experimentos utilizando una metodología tradicional para perfilado de autor, y nuestra metodología propuesta con transferencia de conocimiento. Presentamos resultados de experimentos utilizando varias estrategias de preprocesamiento de textos de redes sociales tanto para la metodología tradicional, como para la metodología con transferencia de conocimientos.

Observamos que ambas metodologías tienen resultados similares, sin embargo, nuestra metodología utiliza un vector de características de menor dimensionalidad, en contraste con los miles de características que se necesitan para entrenar un algoritmo de clasificación utilizando bolsa de palabras o n-gramas de caracteres.

En cuanto al impacto de las estrategias de preprocesamiento, observamos que en los experimentos con bolsa de palabras y n-gramas de caracteres, los resultados fueron mas altos al remover las menciones. De igual manera cuando utilizamos USE, las estrategias de preprocesamiento que ayudan a mejorar los resultados de clasificación son la eliminación de menciones y *hashtags*.

Capítulo 6

Conclusiones y trabajo futuro

En éste capítulo se presentan las conclusiones finales del presente trabajo de tesis. También se proponen posibles direcciones para trabajo futuro.

6.1. Conclusiones finales

El principal objetivo de la presente tesis es poder realizar el perfilado de autor a partir de los textos escritos en *Twitter* en Español usando la metodología USE y, evaluando diferentes estrategias de preprocesamiento tales como eliminación de emoticones, emojies, URLs, menciones, *hashtags* y sustitución de *slangs*. Para lograr tal objetivo se propuso una metodología basada en la transferencia de conocimiento de una red entrenada para realizar otras tareas.

Se ha desarrollado una metodología basada en *Universal Sentence Encoder* que utiliza una red neuronal convolucional para obtener vectores de baja dimensionalidad y utilizarlos como características para realizar el perfilado de autor.

Para evaluar la calidad de los vectores (que representan a todos los *tweets* de un usuario) obtenidos por USE se usaron como características para entrenar dos algoritmos de clasificación que generalmente se obtienen buenos resultados en perfilado de autor [1]. Mediante los experimentos, mostramos que estos vectores permiten identificar el género del autor con una exactitud del 71.98 %, cuando se quitan las menciones a usuarios, con un clasificador SVM para el corpus del PAN 2017. Podemos observar que este resultado es mejor que el obtenido con el *baseline* para la clasificación de género.

A partir de los resultados obtenidos, podemos contestar las preguntas de investigación que se plantearon al inicio de este trabajo de tesis. A continuación respondemos brevemente dichas preguntas:

1. ¿Qué ventajas y desventajas tiene aplicar las técnicas de transferencia de conocimiento en lugar de los métodos tradicionales?

   Una ventaja de utilizar USE es el bajo costo computacional que representa entrenar un algoritmo de aprendizaje automático, pues utiliza vectores densos de baja dimensionalidad (512) lo cual es mucho menor a los vectores dispersos de alta dimensionalidad de la bolsa de palabras (BOW) o los N-gramas de caracteres. También, cuando se quitan los slangs, USE codifica la información de mejor manera, y por ende, los rasgos proporcionados por los caracteres especiales como *emojies*, emoticones y *hashtags* se ven potenciados, como se puede ver en las tablas del capítulo 5. Además, cuando se realiza

éste preprocesamiento se obtiene la mayor diferencia en el rendimiento de los clasificadores del *baseline* y USE. Es decir, USE capta mejor la información proporcionada por los signos no estándares del lenguaje.

2. ¿El preprocesamiento afecta el desempeño del PA utilizando USE?

   El preprocesamiento sí afecta la tarea de PA utilizando USE. Cuando se aplica la mayoría de las técnicas de preprocesamiento el rendimiento de USE mejora (ver Capítulo5), en comparación con el método *baseline*. Sólo cuando se sustituyeron los *slangs*, el rendimiento disminuye en comparación de cuando no se realizó ningún tipo de preprocesamiento.

   Además, USE no es tan sensible al preprocesamiento, es decir, su cambio de un preprocesamiento a otro varía muy poco. Por otra parte, no todos los preprocesamientos afectan de la misma manera al perfilado de autor.

3. ¿Qué características específicas del corpus de *Twitter* influyen más para realizar PA?

   Las características que más afectan al tarea de perfilado de autor son los *slangs*, cuando estos se sustituyeron, en todos los experimentos se obtuvieron los resultados más bajos, esto es, son los que aportan más información al texto. En cambio cuando se quitaron la menciones en todos los experimentos los resultados fueron los más altos. Esto se debe a que las menciones son las que menos aportan información a los textos, ya no conllevan ninguna carga semántica.

6.2. Contribuciones del trabajo

1. Se propuso una metodología para realizar el perfilado de autor de *tweets* en español utilizando transferencia de conocimiento de una red neural que fue entrenada para resolver otro problema.

2. Se evaluó el vector de palabra obtenido de USE, para realizar el perfilado de autor obteniendo resultados mejores a los del sistema base utilizando un vector de características de menor dimensionalidad, lo que disminuye sustancialmente el tiempo de entrenamiento del modelo de aprendizaje.

3. Se evaluó el impacto de las distintas técnicas de preprocesamiento de textos que se encuentran en la literatura, tanto en el sistema base como en el método basado en transferencia de conocimiento. De esta manera, logramos identificar cuáles son las técnicas de preprocesamiento más relevantes para PA utilizando USE.

6.3. Trabajo futuro

Se considera que una posible extensión del presente trabajo considere los siguientes aspectos:

1. Probar USE en un corpus donde la mayoría de los usuarios sean adolescentes, tipo *facebook* o *instagram* para probar el rendimiento en textos donde el lenguaje informal sea más predominante.

2. Probar las otras versiones de modelos de USE entrenados con datos diferentes y conjuntos de mayores dimensiones.

3. Evaluar otras técnicas de transferencia de conocimiento, como el *Universal Language Model Fine-tuning (ULMFit)* [17], que ha logrado muy buenos resultados en problemas de clasificación de textos.

4. Desarrollar una red neuronal que incluya codificadores y decodificadores para entrenar los vectores obtenidos de USE y realizar diversas tareas del procesamiento del lenguaje natural.

Bibliografía

[1] M. E. Aragón and A. P. López-Monroy. Author profiling and aggressiveness detection in spanish tweets: Mex-a3t 2018. In *IberEval@ SEPLN*, pages 134–139, 2018.

[2] S. Argamon, M. Koppel, J. Fine, and A. R. Shimoni. Gender, genre, and writing style in formal written texts. *Text-The Hague Then Amsterdam Then Berlin-*, 23(3):321–346, 2003.

[3] M. Arroju, A. Hassan, and G. Farnadi. Age, gender and personality recognition using tweets in a multilingual setting. In *6th Conference and Labs of the Evaluation Forum (CLEF 2015): Experimental IR meets multilinguality, multimodality, and interaction*, pages 22–31, 2015.

[4] R. Caruana. Multitask learning: A knowledge-based source of inductive bias. In *Machine Learning, Proceedings of the Tenth International Conference, University of Massachusetts, Amherst, MA, USA, June 27-29, 1993*, pages 41–48, 1993.

[5] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, et al. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*, 2018.

[6] J. Cervantes Canales. *Clasificación de grandes conjuntos de datos vía Máquinas de Vectores Soporte y aplicaciones en sistemas biológicos.* PhD thesis, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, 2009.

[7] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[8] S. Daneshvar and D. Inkpen. Gender identification in twitter using n-grams and lsa. In *Proceedings of the Ninth International Conference of the CLEF Association (CLEF 2018)*, 2018.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[10] C. B. Finn. *Learning to Lear with Gradients.* PhD thesis, University of California,Berkeley, 2018.

[11] A. Géron. *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems.* .ºReilly Media, Inc.", 2019.

[12] H. Gómez-Adorno, I. Markov, G. Sidorov, J.-P. Posadas-Durán, M. A. Sanchez-Perez, and L. Chanona-Hernandez. Improving feature representation based on a neural network for author profiling in social media texts. *Computational intelligence and neuroscience*, 2016:2, 2016.

[13] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[14] J. Gu, Y. Wang, Y. Chen, K. Cho, and V. O. Li. Meta-learning for low-resource neural machine translation. *arXiv preprint arXiv:1808.08437*, 2018.

[15] D. Gusfield. *Algorithms on strings, trees, and sequences: computer science and computational biology*. Cambridge university press, 1997.

[16] H. M. Gómez Adorno. *Extracción de características de texto basada en grafos sintácticos integrados*. PhD thesis, Instituto Politécnico Nacional, 2018.

[17] J. Howard and S. Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

[18] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé III. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1681–1691, 2015.

[19] Y. Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[20] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[21] M. Martinc, I. Skrjanec, K. Zupan, and S. Pollak. Pan 2017: Author profiling-gender and language variety prediction. In *CLEF (Working Notes)*, 2017.

[22] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[23] T. M. Mitchell. *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research ..., 1980.

[24] P. Modaresi, M. Liebeck, and S. Conrad. Exploring the effects of cross-genre machine learning for author profiling in pan 2016. In *CLEF (Working Notes)*, pages 970–977, 2016.

[25] J. Pennington, R. Socher, and C. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[26] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[27] J.-P. Posadas-Durán, I. Markov, H. Gómez-Adorno, G. Sidorov, I. Batyrshin, A. Gelbukh, and O. Pichardo-Lagunas. Syntactic n-grams as features for the author profiling task. *Working Notes Papers of the CLEF*, 2015.

[28] F. Rangel, P. Rosso, B. Verhoeven, W. Daelemans, M. Potthast, and B. Stein. Overview of the 4th author profiling task at pan 2016: cross-genre evaluations. In *Working Notes Papers of the CLEF 2016 Evaluation Labs. CEUR Workshop Proceedings/Balog, Krisztian [edit.]; et al.*, pages 750–784, 2016.

[29] F. M. Rangel Pardo. *Author Profiling en Social Media: Identificación de Edad, Sexo y Variedad del Lenguaje*. PhD thesis, Universidad Politécnica de Valencia, 2016.

[30] S. Reed, Y. Chen, T. Paine, A. v. d. Oord, S. Eslami, D. Rezende, O. Vinyals, and N. de Freitas. Few-shot autoregressive density estimation: Towards learning to learn distributions. *arXiv preprint arXiv:1710.10304*, 2017.

[31] G. Salton. Automatic text processing: The transformation, analysis, and retrieval of. *Reading: Addison-Wesley*, 169, 1989.

[32] J. Schler, M. Koppel, S. Argamon, and J. W. Pennebaker. Effects of age and gender on blogging. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, volume 6, pages 199–205, 2006.

[33] J. Schmidhuber. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. PhD thesis, Technische Universität München, 1987.

[34] M. Sukanya and S. Biruntha. Techniques on text mining. In *2012 IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, pages 269–271. IEEE, 2012.

[35] T. Takahashi, T. Tahara, K. Nagatani, Y. Miura, T. Taniguchi, and T. Ohkuma. Text and image synergy with feature cross technique for gender identification. *Working Notes Papers of the CLEF*, 2018.

[36] S. Thrun. Is learning the n-th thing any easier than learning the first? In *Advances in neural information processing systems*, pages 640–646, 1996.

[37] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pages 3320–3328, 2014.