



**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO®

# INSTITUTO TECNOLÓGICO DE CULIACÁN



## **Creación de un corpus y clasificación de textos en español para reconocimiento de personalidad**

TESIS

PRESENTADA ANTE EL DEPARTAMENTO ACADÉMICO DE ESTUDIOS DE POSGRADO  
DEL INSTITUTO TECNOLÓGICO DE CULIACÁN EN CUMPLIMIENTO PARCIAL DE LOS  
REQUISITOS PARA OBTENER EL GRADO DE

**MAESTRO EN CIENCIAS DE LA COMPUTACIÓN**

POR:

**Jonathan Iván Roldan Arana**

Ingeniero Mecánico

DIRECTOR DE TESIS:

Dr. Ramón Zatarain Cabada

CULIACÁN, SINALOA

Noviembre de 2022

# AUTORIZACIÓN DE IMPRESIÓN

(Dejar esta hoja en blanco)

El tipo de texto que se debe usar en cada parte de este documento ya está establecido. Cada nota o explicación de este documento ya cuenta con las configuraciones necesarias como: tipo de letra, interlineado, tamaño, color, tipo de título, formato, espaciado, sangría, etc., que deben ser usadas para la redacción de cada apartado.

Para la cuestión de títulos, subtítulos, referencias de tablas y referencias de figuras se presenta información que describe cómo se utiliza cada una de ellas y por qué el respectivo uso.

Todo el texto a excepción del título debe ser borrado antes de imprimir.

# Hoja de firmas

En esta hoja vendrán las firmas. Dejar en blanco

# Dedicatoria

Este nuevo logro en mi vida, sin pensarlo se lo dedico a mi mamá, Yolanda Arana Reyes, quién es mi ejemplo para seguir como persona, como madre y como ser humano. Soy muy afortunado por la madre que Dios me dio, eres amiga, mi consejera, mi guía. Gracias por darme esa seguridad, respaldo y apoyo en los momentos más difíciles cuando me siento perdido o desubicado, como lo fue esta etapa de mi vida, donde estuve a punto de tirar la toalla y con tus consejos y motivación influiste en mí para a dar todo y luchar hasta el último. Gracias por ser exigente y creer en tus hijos, dándonos en todo momento tu amor y apoyo incondicional, por eso te digo que: Nuestros logros son tus éxitos.

Gracias a mi hermana Ivonne Lizbeth Roldan Arana, quien, para mí, es un ejemplo claro de que no hay que quitar de la mira el objetivo ni las metas que uno se propone, que todo se puede lograr incluso cuando uno se cae o se equivoca. Me enseñó que debes seguir luchando y ser persistente hasta conseguir tu objetivo. Eres muy inteligente hermana gracias por tu apoyo moral y cariño.

A mi padre Gregorio Alfredo Roldán Cisneros que a pesar de la distancia siempre ha estado al pendiente de sus hijos, dándome su apoyo moral y económico para poder solventar mis estudios.

Este trabajo se lo dedico a mi abuela Leonila Reyes Hernández, ya que ha sido una excelente madre que supo cuidar a sus hijas e hijo, supo apoyarlos y darles estudio a todos, alejándolos de distracciones o vicios. Gracias abuela porque siempre ha sido muy linda con sus nietos, gracias por consentirnos y apoyarnos en la medida que puede.

Se la dedico a mi abuelo que descansa en paz José Arana Morales (1926-2015) quién influyó en mí para ser ingeniero. Me hubiera gustado que viera mis logros, pero estoy seguro me sigue la pista desde el cielo.

También dedico este éxito alcanzado a mi Familia que me apoya, me quiere y acepta como soy, gracias por su apoyo, me alientan a seguir preparándome. Son mi motor para seguir siendo ese nieto, primo, sobrino del cuál estén orgullosos.

# Agradecimientos

Quiero agradecer en primera instancia al Instituto Tecnológico de Culiacán, particularmente al programa de Maestría en Ciencias Computacionales, así como a sus colaboradores por las facilidades que me brindaron para alcanzar este nuevo objetivo en mi formación académica. De mismo modo agradezco a CONACYT que gracias a su apoyo económico solventó los gastos personales y académicos para poder dedicarme exclusivamente a este reto académico ya que sin esta ayuda hubiera sido casi imposible de lograr obtener el grado, debido a la complejidad y demanda de tiempo que solicita el programa.

De igual forma doy gracias a mi director de tesis, el Dr. Ramón Zatarain Cabada, quién es un docente e investigador de primer nivel del Instituto Tecnológico de Culiacán que, con su experiencia y visión profesional me apoyo como guía para no perder de vista el objetivo de la tesis en este trabajo, brindándome una buena orientación por medio sus consejos, gestionando los recursos necesarios para culminar mi investigación.

También agradezco a mis revisores externos, la Dra. Lucía Barrón Estrada y a mi querido amigo MC. Víctor Manuel Bátiz Beltrán por sus observaciones, recomendaciones y resolución de dudas a mi proyecto de tesis.

# **Declaración de autenticidad**

Por medio de la presente hago constar que el trabajo de esta tesis de maestría es original y no copia de otras tesis o artículos o congresos etc., ya que es el resultado de mi esfuerzo profesional. Por otro lado, los textos, figuras, tablas, gráficas mostradas en el trabajo y cuyo origen no son de mi creación intelectual, están protegidos por derechos de autor por lo que se hace una referencia explícita a la fuente o al autor.

Jonathan Iván Roldan Arana

Culiacán, Sinaloa, México, noviembre de 2022

# Resumen

Existen diferentes teorías o modelos utilizados para el reconocimiento de personalidad dentro del mundo de la psicología, dentro de estas teorías destaca el modelo llamado “los cinco grandes” (*Big Five* en inglés). Esta teoría divide la personalidad en 5 rasgos llamados: apertura, conciencia, extroversión, amabilidad y neuroticismo. Actualmente hay investigaciones que buscan reconocer automáticamente la personalidad con diferente enfoque. Hay investigaciones que tienen como objetivo detectar la personalidad por medio de reconocimiento facial. También hay enfoques que hacen uso del reconocimiento de emociones y de sentimientos, así como a través de reconocimiento de tonos de la voz de los individuos. Dentro de estos tipos de vertientes se pueden mencionar algunos donde se utilizan textos provenientes de transcripciones de voz o textos escritos por la persona de los cuales se puede determinar si una persona presenta o no un determinado rasgo de personalidad. Dichos métodos se basan en algoritmos de *Machine Learning* o *Deep Learning* los cuales requieren de una base de datos para su entrenamiento. En este trabajo se presenta la creación del corpus de textos llamado *PersonText*, el cual contiene 213 textos en español con sus respectivas etiquetas, así como los inventarios de resultados de prueba IPIP. Esta información se obtuvo de una plataforma *PersonApp*, que es utilizada para reconocimiento de personalidad mediante textos basado en el modelo Big Five. Adicional, se realizaron pruebas con un modelo *Deep Learning*, con la finalidad de evaluar este corpus y conocer si el resultado obtenido es confiable o no, donde el mejor resultado obtenido fue de 74% de precisión.

# Palabras clave

Aprendizaje máquina

Aprendizaje profundo

Computación afectiva

Corpus

Emociones

*Ipip*

Métricas de evaluación

*Ocean*

Personalidad

Preprocesamiento

Procesamiento natural de lenguaje

Pruebas estandarizadas de personalidad

Reconocimiento automático de personalidad

Redes neuronales



## Índice general

<b>1.</b>	<b>Introducción.....</b>	<b>1</b>
1.1.	Planteamiento del problema.....	2
1.2.	Objetivo general .....	3
1.3.	Objetivos específicos .....	3
1.4.	Hipótesis .....	3
1.5.	Justificación y contribuciones .....	4
1.6.	Estructura de las tesis .....	4
<b>2.</b>	<b>Marco teórico .....</b>	<b>6</b>
2.1.	Computación Afectiva.....	6
2.1.1.	Emociones .....	7
2.1.1.1.	¿Qué son las Emociones? .....	7
2.1.1.2.	Técnicas de reconocimiento de Emociones .....	9
2.1.2.	Sentimientos .....	11
2.1.2.1.	¿Qué son los Sentimientos? .....	11
2.1.2.2.	Técnicas de reconocimiento de Sentimientos .....	11
2.1.3.	Personalidad .....	14
2.1.3.1.	Teorías de la personalidad.....	15
2.1.3.2.	Modelo de los cinco grandes ( <i>Big-Five</i> ) .....	18
2.1.3.3.	International Personality Item Pool (IPIP) .....	19
2.1.3.4.	Reconocimiento automático de personalidad .....	20
2.2.	Procesamiento Natural de Lenguaje.....	21
2.3.	Aprendizaje máquina.....	21
2.3.1.	Aprendizaje no supervisado .....	22
2.3.2.	Aprendizaje supervisado .....	23
2.3.2.1.	Árboles de decisión.....	23
2.3.2.2.	Algoritmos Basados en Reglas .....	24
2.3.2.3.	Redes Bayesianas .....	24
2.3.2.4.	Support Vector Machine .....	25
2.3.2.5.	Redes Neuronales .....	25
2.3.3.	Aprendizaje profundo.....	27
2.3.3.1.	Perceptrón Multicapa .....	29
2.3.3.2.	Redes Recurrentes.....	30

2.3.3.3.	Redes de memoria de corto plazo .....	32
2.3.3.4.	Redes Convolucionales.....	33
2.3.3.5.	Aprendizaje profundo con datos en texto .....	37
2.4.	Conjunto de datos.....	39
3.	Estado del arte .....	41
3.1.	Corpus de texto utilizados en el reconocimiento de personalidad .....	41
3.1.1.	Essays .....	42
3.1.2.	MyPersonality .....	42
3.1.3.	B5 .....	42
3.1.4.	Pan2015 .....	43
3.1.5.	Otros conjuntos de datos de Redes sociales y plataformas de internet .....	43
3.2.	Modelos de reconocimiento de personalidad con base en texto .....	43
4.	Desarrollo del proyecto .....	49
4.1.	Una metodología para la creación del corpus PersonText .....	49
4.1.1.	Plataforma PersonApp .....	50
4.1.2.	Transformación de video a texto.....	53
4.1.3.	Extracción de resultados de pruebas IPIP .....	55
4.1.4.	Relación de textos y etiquetas de valores de rasgos de personalidad.....	56
4.1.5.	Creación de etiquetas de presencia / ausencia de rasgo de personalidad.....	57
4.1.6.	Estructura final del corpus PersonText .....	58
4.2.	Metodología para crear y entrenar Redes Neuronales DL para reconocimiento de personalidad mediante texto .....	59
4.2.1.	Datos de entrenamiento .....	61
4.2.2.	Preprocesamiento del corpus .....	61
4.2.3.	Extracción de características / Tokenización.....	63
4.3.	Creación de las redes neuronales de aprendizaje profundo .....	66
4.3.1.	Configuración de modelos de Aprendizaje Profundo .....	67
4.3.2.	Perceptrón Multicapa .....	68
4.3.3.	<i>Long Short-Term Memory Simple (LSTM)</i> .....	69
4.3.4.	LSTM Bidireccional .....	70
4.3.5.	Red Neuronal Convolucional .....	70
5.	Pruebas y Resultados .....	72
5.1.	Métrica de evaluación .....	72

5.2.	Separación de datos de Entrenamiento y Validación.....	72
5.3.	Entrenamiento y Evaluación de modelos DL.....	73
5.3.1.	Entrenamiento y Evaluación del Corpus PersonText.....	74
5.3.2.	Entrenamiento y Evaluación del Corpus MyPersonaliy.....	75
5.3.3.	Entrenamiento y Evaluación del Corpus Essays.....	77
5.4.	Resultados Finales.....	77
6.	Conclusiones y trabajo futuro.....	79
6.1.	Conclusiones.....	79
6.2.	Trabajo a futuro.....	82
	Referencias.....	83

# Índice de figuras

Figura 2-1. Rueda de las emociones o modelo circular de Plutchik. (Plutchik, 1979).....	8
Figura 2-2. Modelo cónico o tridimensional de Plutchik. (Plutchik, 1979) .....	8
Figura 2-3. Arquitectura de un sistema típico de análisis de sentimientos. ....	14
Figura 2-4. Ejemplo de un árbol de decisión.....	24
Figura 2-5. Ejemplo de clasificación con SVM. ....	25
Figura 2-6. Arquitectura de una Red neuronal artificial. ....	26
Figura 2-7. Relación entre la inteligencia artificial y el aprendizaje profundo. ....	27
Figura 2-8. Perceptrón multicapas con dos capas ocultas. ....	29
Figura 2-9. Red Neuronal Recurrente. ....	31
Figura 2-10. Red LSTM. ....	32
Figura 2-11. Arquitectura de una red CNN general de alto nivel (Patterson & Gibson, 2017). ....	34
Figura 2-12. Desplazamiento el filtro sobre una entrada determinada para realizar la convolución (Albawi, S. et al., 2018). ....	35
Figura 2-13. Ejemplo de una capa de pooling (Max pooling).....	36
Figura 2-14. Vectorización de texto. Imagen de Chollet (2017).....	38
Figura 4-1. Interfaz de bienvenida de la plataforma PersonApp (Bátiz et al., 2022) .....	50
Figura 4-2. Cuestionario IPIP de 50 ítems, imagen obtenida de (Bátiz et al., 2022) .....	51
Figura 4-3. Información de usuarios recolectada por plataforma PersonApp y almacenada en nube de Firebase. ....	52
Figura 4-4. Almacenamiento de videos en la nube de Firebase. ....	53
Figura 4-5. Metodología para creación de corpus de texto. ....	54
Figura 4-6. Primer prototipo del corpus PersonText.....	55
Figura 4-7. Evaluaciones de los cinco rasgos de personalidad de varios usuarios.....	56
Figura 4-8. Imagen del segundo prototipo del corpus PersonText.....	56
Figura 4-9. Asignación de valores numéricos a etiquetas. ....	58
Figura 4-10. Vista parcial del corpus PersonText versión Final. ....	59
Figura 4-11. Metodología para crear redes neuronales utilizando el corpus de texto. ....	60
Figura 4-12. Proceso de Tokenización de texto. ....	64
Figura 4-13. Orden de tokens por la frecuencia de palabras. ....	65
Figura 4-14. Fragmento del código para tokenizar el corpus PersonText.....	66
Figura 4-15. Modelo Perceptrón multicapa.....	68
Figura 4-16. Modelo LSTM simple. ....	69
Figura 4-17. Modelo LSTM Bidireccional.....	70
Figura 4-18. Modelo de CNN. ....	71
Figura 5-1. Validación de los modelos.....	73
Figura 5-2. Técnica early-stopping. ....	74

# Índice de tablas

Tabla 2-1. Enfoques básicos de la psicología de la personalidad (Funder, 2013).....	15
Tabla 2-2. 16 factores primarios y los cinco rasgos de personalidad globales.....	16
Tabla 2-3. Tres Gigantes y rasgos primarios propuestos por Eysenck.....	17
Tabla 2-4. Funciones de activación de las neuronas. ....	27
Tabla 4-1. Preguntas o instrucciones para las grabaciones de los participantes.....	52
Tabla 5-1. Exactitud (accuracy) obtenida de pruebas con diferentes corpus. ....	77

# Capítulo 1

## 1. Introducción

La personalidad es un conjunto de rasgos que determinan la manera de actuar o tomar decisiones de una persona. Estos rasgos son conceptos como las emociones, los sentimientos y características de patrón de actitudes y pensamientos (Funder, 2013).

Es importante mencionar que cada ser humano posee una personalidad única en el mundo y es lo que hace especial a cada persona. La personalidad es lo que induce a la persona a tomar decisiones como lo son el elegir el tipo de vestimenta que usa a diario, los artículos que compra en el supermercado, pasatiempos que tiene, lugares que visita o incluso, los eventos sociales a los que asiste.

Por lo anterior, reconocer la personalidad es importante y para ello existen teorías de reconocimiento de personalidad las cuales clasifican o reconocen la personalidad de acuerdo con la anatomía, fisiología, genética o incluso mediante un psicoanálisis de las personas (Funder, 2013); dichas teorías han sido la base para el desarrollo de métodos o técnicas de aprendizaje máquina, que son utilizadas para el reconocimiento de personalidad automáticamente. Dentro de estas técnicas se incluye el reconocimiento facial, reconocimiento de personalidad a partir de voz y texto, reconocimiento de emociones y de sentimientos los cuales son acercamientos al reconocimiento de personalidad.

Un ejemplo de estos trabajos es una investigación donde reconocen la personalidad a partir de publicaciones de texto que aparecen en las redes sociales, utilizando métodos de aprendizaje profundo *Deep Learning* (DL por sus siglas en inglés), como son algunos algoritmos de regresión tradicionales. El trabajo de Xue et al. (2018) tiene como objetivo principal reconocer la personalidad desglosada en valores numéricos de acuerdo con la teoría de los cinco grandes conocida como *Big Five* (Goldberg, 1992) y mediante el entrenamiento del algoritmo propuesto obtienen el valor correspondiente a cada uno de los rasgos de personalidad en una escala establecida.

Para la realización de dichas investigaciones se requieren bases de datos o conjuntos de datos que contengan textos, con etiquetas de los rasgos de personalidad y sus respectivos valores que puedan ser utilizados para el entrenamiento de los métodos de aprendizaje máquina.

La principal contribución de este trabajo es la creación de un corpus de textos en idioma español que incluye etiquetas de personalidad con base en el modelo de los cinco grandes. El contenido del conjunto de datos fue extraído de una plataforma desarrollada internamente por nuestro grupo de investigación. Esta plataforma tiene como objetivo la recolección de datos como videos y pruebas estandarizadas de personalidad de los participantes. Estas pruebas se crearon con base a las preguntas disponibles en el *International Personality Item Pool* (IPIP, International Personality Item Pool, 2021), que es un laboratorio de desarrollo de preguntas y escalas avanzadas en la medición de la personalidad. Por otra parte, se desarrollaron y optimizaron diversos modelos de aprendizaje máquina para probar el corpus y hacer el reconocimiento de personalidad.

### **1.1. Planteamiento del problema**

Actualmente se lleva a cabo mucho trabajo de investigación para que las computadoras logren reconocer las emociones, sentimientos y personalidad humana, para tareas como ayudar a los alumnos en el aprendizaje de algún tema, mediante la implementación de Sistemas Tutores Inteligentes (STI). Los STI son programas de computadora que enseñan contenido educativo simulando a un tutor humano (Nwana, 1990), brindando explicación del tema a tratar, adaptando los contenidos de acuerdo con el nivel de conocimiento de los educandos, haciendo cálculos y ajustes correspondientes. Uno de los objetivos de los STI es identificar la dificultad de los ejercicios para los usuarios tomando en cuenta tanto sus conocimientos como su nivel de interés en el tema, con el propósito de mejorar su aprendizaje. También se busca adecuar estos programas educativos de acuerdo con su personalidad, por lo tanto, se están desarrollando métodos de aprendizaje máquina que reconozcan la personalidad del usuario de un STI.

En esta clase de métodos de reconocimiento de personalidad se usan modelos de aprendizaje máquina tradicionales: *Support Vector Machine (SVM)*, *Logistic Regression*, *Naïve Bayes*, *Gradient Boosting* y *Linear Discriminant Analysis (LDA)*, así como modelos de aprendizaje

profundo: *Multi Layer Perceptron* (MLP), *Gated Recurrent Units* (GRU), *Long Short-Term Memory* (LSTM), o *Convolutional Neural Network* (CNN). Dichos modelos requieren de bases de datos o conjuntos de datos para su entrenamiento. El problema principal es que la mayoría de los conjuntos de datos construidos actualmente están en idiomas diferentes al español, algunos ejemplos son: el corpus *myPersonality* y el corpus *essays*.

## 1.2. Objetivo general

Esta investigación tiene como propósito principal diseñar y desarrollar un conjunto de datos de textos en español para su uso en métodos de reconocimiento automático de personalidad, mediante la extracción de textos de los videos que se encuentran en la plataforma llamada *PersonApp*. Dichos textos servirán para el entrenamiento de modelos de aprendizaje máquina con el fin de crear un reconocedor automático de personalidad a través de texto.

## 1.3. Objetivos específicos

- Diseñar y desarrollar un sistema que descargue datos en formato de video de la nube y transformarlos automáticamente en textos en español.
- Definir la estructura del corpus de texto y el tipo de datos o información personal requerida de los autores de los textos.
- Desarrollar un proceso filtrado y extracción de características de textos.
- Crear una metodología que permita realizar el etiquetado de cada uno de los rasgos de personalidad a través de textos.
- Evaluar el corpus de textos mediante el uso de técnicas disponibles de aprendizaje máquina, como son las redes neuronales.
- Realizar una evaluación y comparación del conjunto de datos a desarrollar, con otros tipos de corpus existentes.

## 1.4. Hipótesis

Es posible reconocer automáticamente la personalidad de un individuo de habla hispana a través del uso de texto de forma eficiente y rápida. **ESTO NO ES UNA HIPÓTESIS**



Se puede crear un conjunto de datos en español basado en la teoría de Big Five, que pueda ser utilizando en modelos *DL* como sustituto de los corpus de texto convencionales, y que tenga obtenga mejor precisión en el reconocimiento automático de la personalidad.

### **1.5. Justificación y contribuciones**

En esta investigación se desarrollará un corpus en español, cuyo propósito es el entrenamiento, validación y prueba de modelos de reconocimiento automático de personalidad a través del texto. El reconocimiento de personalidad será de utilidad para mejorar métodos de reconocimiento de emociones lo cual a su vez ayudará a un mejor desarrollo de los STI de enseñanza y que los educandos obtengan un mejor aprovechamiento y desempeño académico.

La principal contribución de la presente investigación será la creación de un corpus de textos en español con etiquetas de rasgos de personalidad con base en el modelo *Big-Five*, ya que actualmente no está disponible algún corpus de textos en el idioma español y podrá ser utilizado para fines diversos, como es el reconocimiento de personalidad.

### **1.6. Estructura de las tesis**

Esta tesis contiene una serie de capítulos donde se muestra el proceso de desarrollo y los resultados obtenidos de esta investigación, lo cual se describe a continuación:

El capítulo 2 muestra el marco teórico referente a la presente investigación donde se presentan los elementos teóricos necesarios para comprender a la perfección el tema a tratar, se incluyen las definiciones de conceptos que se mencionan constantemente a lo largo de todo este. También se mencionan los tipos de teorías de reconocimiento de personalidad. Además, se hace un acercamiento de lo que es y para qué sirve el procesamiento de lenguaje natural en la clasificación de textos. Por otro lado, se revisan algunas técnicas de aprendizaje máquina que están disponibles y que fueron utilizadas en el reconocimiento de personalidad, emociones y sentimientos.

En el capítulo 3 se muestran los trabajos relacionados con corpus de texto que son muy populares y se utilizan en el campo de la computación en investigaciones de reconocimiento

de personalidad. Además, se analizan otras investigaciones donde se utilizan dichos conjuntos de datos de texto para entrenar modelos de aprendizaje máquina tradicional y modelos *Deep Learning*.

En el capítulo 4 se muestra todo el proceso de creación y desarrollo del corpus de textos, se explica la metodología utilizada para la definición de la arquitectura o estructura del conjunto de datos y se describe el origen de los datos que contiene. Después, se da a conocer la metodología utilizada para reconocer la personalidad mediante texto, es decir, los pasos que se siguen para procesar los textos del corpus, extraer sus características y transformarlas en valores que puedan ser reconocidos por las redes neuronales. Por último, se da a conocer el tipo de configuración, así como los entrenamientos de estos modelos.

En el capítulo 5 se muestran las pruebas realizadas y resultados obtenidos con los modelos de aprendizaje profundo previamente configurados utilizando para su entrenamiento tanto el conjunto de datos creado, como otros corpus disponibles utilizados comúnmente en el campo de la computación para el reconocimiento de personalidad.

El capítulo 6 se muestran las conclusiones obtenidas en este trabajo de investigación. También se establece el aporte más relevante del proyecto a la comunidad científica y se describen las limitaciones que se presentaron durante el desarrollo de esta investigación. Así mismo, se mencionan los trabajos a realizar en un futuro como continuación a este proyecto. Finalmente se muestran las referencias bibliográficas consultadas para la elaboración de este trabajo.

# Capítulo 2

## 2. Marco teórico

En este capítulo se presentan las bases teóricas que sustentan este trabajo de tesis. Los temas que se abordan son: la Computación Afectiva, el Procesamiento Natural de Lenguaje y Conjunto de datos. Se dan a conocer algunas investigaciones relacionadas con estos temas y los conceptos que son fundamentales para profundizar más en el entendimiento de esta investigación.

### 2.1. Computación Afectiva

Desde la invención de las computadoras, los investigadores y científicos en el campo de las ciencias de la computación, han buscado formas de que las computadoras interactúen con las personas como lo haría un ser humano, razón por la cual se han creado muchas ramas de estudio en este campo, una de ellas es la computación afectiva.

La computación afectiva es un campo multidisciplinario que tiene dentro de sus áreas de estudio la psicología, la psicología cognitiva y la informática. De acuerdo con (Picard, 1997) la computación afectiva es el campo de estudio enfocado en lograr que las computadoras posean la capacidad de reconocer y expresar sentimientos como un ser humano, así como desarrollar la capacidad de enviar respuestas inteligentes a las emociones expresadas por los humanos.

La computación emocional tiene como uno de sus objetivos recolectar información que contenga los sentimientos de los usuarios. Una vez recolectada, la información es analizada para mejorar la experiencia del usuario, por ejemplo, en educación existen aplicaciones (Barron Estrada et al., 2019) que, en tiempo real, reconocen las emociones del estudiante con el fin de modificar su contenido de acuerdo con la emoción expresada.

### **2.1.1. Emociones**

La computación afectiva tiene como tarea encontrar significado a las emociones. En este subtema se le encuentra una definición a lo que son las emociones, así como algunas técnicas para el reconocimiento de estas con ayuda de las ciencias computacionales.

#### **2.1.1.1. ¿Qué son las Emociones?**

Para muchas personas que no están familiarizadas con el tema, es fácil confundir el concepto de emoción con términos similares como los sentimientos, el afecto o estado de ánimo. Las emociones según (Caruso & Salovey, 2005) se caracterizan por ser específicas y responder a un evento específico. En otras palabras, las emociones son una respuesta a un evento. Por otro lado, los estados de ánimo son emociones que se mantienen en el tiempo, es decir, son de larga duración y no dependen de ningún acontecimiento, sino que, debido a diferentes eventos, vivencias o traumas del pasado, contribuyen para que la persona genere una respuesta emocional.

El término emoción para Hockenbury & Hockenbury (1997), es un estado psicológico complejo que consta de tres componentes: experiencia subjetiva, respuesta fisiológica y respuesta expresiva o conductual. Las emociones son reacciones de menor duración y más intensas que un estado de ánimo, las cuales se producen al recibir algún tipo de estímulo externo y pueden ser fisiológicas, psicológicas o conductuales.

Según Gross & Barrett (2011), las emociones se refieren a un conjunto de estados psicológicos que incluyen experiencias, comportamientos expresivos (rostro, cuerpo, habla) y respuestas del cuerpo humano (frecuencia cardíaca, respiración). Las emociones son una parte importante del modelo psicológico de la mente humana. Las emociones se expresan a través de patrones únicos y consistentes de respuestas subjetivas, expresivas y fisiológicas.

Plutchik (1979) presentó las emociones en un modelo estructural conocido como la rueda de las emociones. Este modelo es un recurso gráfico que sirve para identificar y comprender diferentes tipos de emociones, agrupándolas según su similitud. El modelo estructural de las emociones se muestra en la Figura 2-1. Para este modelo también se ha desarrollado la vista cónica o tridimensional (Figura 2-2), con el que se puede medir y analizar la intensidad de las emociones básicas a partir de las derivadas de sus compuestos.

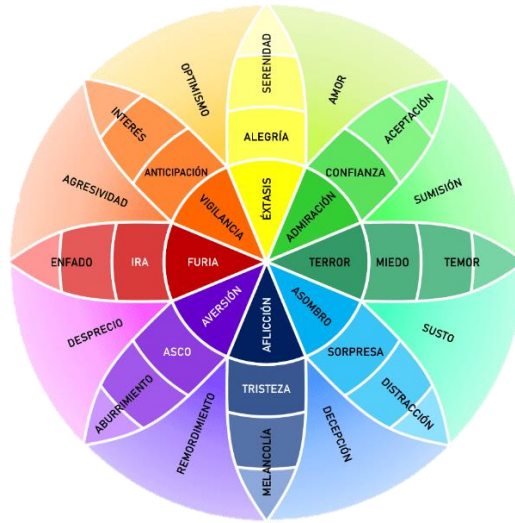


Figura 2-1. Rueda de las emociones o modelo circular de Plutchik. (Plutchik, 1979)

Existen dos aspectos teóricos de la investigación de las emociones:

1. Teorías Cognitivas: Donde la emoción es vista como la capacidad de buscar soluciones o información en el campo de la cognición y tomar decisiones basadas en la experiencia.
2. Teorías Neurológicas: Indica que una emoción es un estado mental debido al efecto de ciertas sustancias químicas en el cerebro límbico.

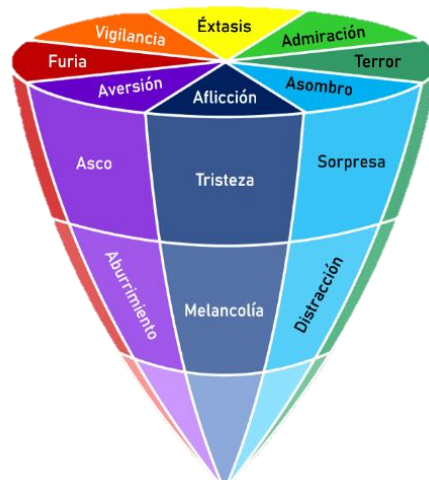


Figura 2-2. Modelo cónico o tridimensional de Plutchik. (Plutchik, 1979)

### **2.1.1.2. Técnicas de reconocimiento de Emociones**

Un sistema cognitivo es un sistema natural o artificial de procesamiento de información capaz de percibir, aprender, razonar, comunicar, actuar y poseer un comportamiento adaptativo (Morris et al., 2006). El reconocimiento automático de emociones puede llevarse a cabo por varios métodos o modalidades. La psicología de la percepción señala que los sistemas cognitivos hacen uso de diversas modalidades sensoriales, así como señas para poder transmitir un mensaje con un significado; estas modalidades pueden ser visuales, auditivas, táctiles e incluso de forma escrita. Para fines tecnológicos, por modalidad o método se entiende al tipo específico de información y al formato de representación en el que se almacena. Los métodos utilizados para reconocimiento de las emociones se pueden clasificar en cuatro categorías:

- Fisiológicas (expresiones faciales, gestos o movimientos corporales). En el rostro se ven reflejadas expresiones faciales y gestos, los cuales contienen información no verbal que complementa la comunicación con otro ser humano. Gracias al rostro se puede comprender el estado emocional de una persona a través de sus diferentes expresiones, convirtiéndolo en una herramienta fundamental para comunicar o expresar los sentimientos y emociones en un diálogo o interacción. Los trabajos de Busso et al. (2004) y Valstar et al. (2012) son un ejemplo de este tipo de enfoque.
- Comportamiento (voz y texto): Reconocer el movimiento de los gestos en la voz es una forma de recopilar información de la comunicación humanas. La comunicación verbal, engloba no solo a las palabras que se dicen sino también a los diferentes tonos auditivos que se dan a la voz al momento de hablar o expresar las emociones. Todas las emociones se pueden clasificar, ya que están presentes en todas las etapas de la comunicación verbal humana. También se ha buscado métodos para que los robots, computadoras y otros dispositivos inteligentes generen una respuesta emocional a través de sonido. En el trabajo de Alu et al. (2017) se crea una técnica para procesar y clasificar archivos de audio mediante el reconocimiento de emociones. Otro enfoque de esta técnica es utilizado para reconocimiento de las emociones a partir de un texto. Definir las emociones en los textos es un reto porque depende en gran medida del contexto; además, las expresiones de los humanos contienen más de una emoción, por lo tanto, es difícil distinguir unas emociones de otras en un texto. En su

estudio (Calefato et al., 2017), crearon una herramienta para reconocer emociones a partir de textos recopilados de una página de internet que da soporte a programadores, los textos contienen comentarios y algunas preguntas y respuestas de temas diversos de los usuarios de la página. De acuerdo con Seyeditabari et al. (2018), la mayoría de los estudios de reconocimiento de emociones en el procesamiento del lenguaje natural basado en texto utilizan tres técnicas diferentes:

- 1) Texto etiquetado: el texto etiquetado, como su nombre lo indica, utiliza una gran base de datos de texto representativo a cada una de las clases que utilizan etiquetas para cada documento completo para realizar el entrenamiento de los modelos.
  - 2) Lexicón de emociones: a diferencia del texto resaltado, el lexicón de emociones utiliza los valores de las emociones expresadas en cada palabra utilizada, por lo que los sentimientos en el documento se calculan midiendo los diferentes emociones en cada palabra del documento.
  - 3) Incrustación de palabras: esta es una nueva técnica basada en la idea de que la información que se encuentra en un conjunto de datos que es semánticamente similar; donde cada palabra del corpus se puede representar como un vector en un espacio de  $n$  dimensiones, llamado espacio vectorial. De algún modo, las distancias entre estos vectores representan la diferencia semántica entre estas palabras.
- Señales cerebrales: esta técnica se basa en el uso de un electroencefalograma (EGG), que mide la actividad eléctrica en el cerebro. Con el EGG, se puede la actividad cerebral con la finalidad de procesar y encontrar características interesantes en dichas señales para poder realizar el reconocimiento de emociones.
  - Sistema multimodales: éste es un método moderno que integra las técnicas anteriores; su finalidad principal es simular la interacción humana, aunque hay muchos retos y problemas que resolver, en general al hablar de multimodal se refiere a técnicas donde se capturan las expresiones faciales significativas, gestos corporales, movimientos de cabeza, sonidos y connotaciones entre letras de los participantes.

### **2.1.2. Sentimientos**

En este subtema se define el significado de los sentimientos, así como algunas teorías y técnicas de análisis de sentimientos en el área de la computación.

#### **2.1.2.1. ¿Qué son los Sentimientos?**

Para los profesionales e investigadores existe la confusión entre la diferencia de los términos de sentimiento y opinión, y si el campo de estudio debería llamarse análisis de sentimientos o minería de opiniones. Para Liu (2015) las opiniones expresadas oralmente o por escrito pueden causar cambios en las creencias y el estado emocional de una persona, es decir, se produce un sentimiento el cual puede ser positivo o negativo.

En el diccionario (Merriam-Webster, 2022) un sentimiento se define como una actitud, pensamiento o juicio provocado por una emoción, mientras que una opinión se define como un punto de vista, un juicio o evaluación generada en la mente de la persona sobre un tema en específico. La diferencia es muy sutil, ya que cada uno contiene un elemento del otro. Las definiciones hacen saber que mientras una opinión es el punto de vista de una persona sobre algo, un sentimiento es simplemente un sentimiento.

#### **2.1.2.2. Técnicas de reconocimiento de Sentimientos**

El análisis de sentimiento, también conocido como minería de opiniones, tiene como objetivo analizar las opiniones, sentimientos, actitudes, valoraciones y emociones de las personas hacia las entidades y sus características. El término entidades se refiere a productos, servicios, organizaciones, personas, acontecimientos, problemas o algún tema en general. (Liu, 2015). El área de trabajo es muy amplia para este campo de estudio y cada enfoque está estrechamente relacionado, pero con un diferente propósito, entre los enfoques se encuentran: el análisis de sentimientos, minería de opiniones, la extracción de opiniones, minería de sentimientos, el análisis de emociones, el análisis de afecto.

Feldman, R. (2013), define el análisis de sentimientos como la tarea de buscar la opinión del autor sobre ciertas entidades. Él describe que en el área del análisis de sentimiento existen cinco problemas específicos los cuales son:



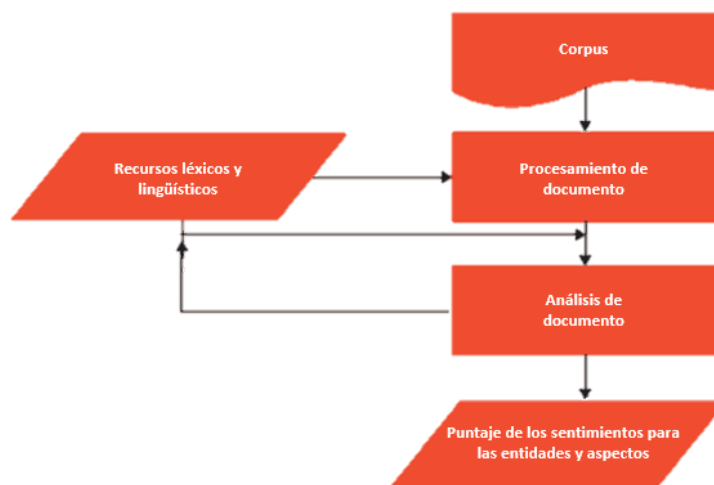
- a) **Análisis de sentimiento a nivel de documento:** En este análisis se da por hecho que el documento ya contiene la opinión expresada en texto por el autor sobre un tema en específico. Existen dos vertientes sobre el análisis de sentimientos a nivel de documento: el aprendizaje supervisado y el aprendizaje no supervisado. En el enfoque de aprendizaje supervisado existe una cantidad de clases conocidas donde el modelo clasificará el documento. Para esto, el modelo se entrena con datos suficientes para cada clase. El caso más simple es cuando solo existen dos tipos de clases: positivo o negativo. Hay otros casos donde existe el positivo, negativo y neutro o puede tener un valor. Los métodos de análisis de sentimientos a nivel de documento con aprendizaje no supervisado se basan en determinar la orientación semántica de algunas frases en los documentos. El documento se clasifica como positivo, si la orientación semántica promedio de estas oraciones está por encima del umbral definido, de otro modo se considera negativo.
- b) **Análisis de sentimiento a nivel de oración:** Este enfoque se utiliza para las situaciones en las que se busca reconocer sentimientos de un documento que contiene varios comentarios de la misma entidad en el mismo archivo, o cuando se busca un punto de vista más a detalle de la opinión de la entidad. Esto se logra separando la oración completa en varias oraciones que contengan una sola opinión. Después se determina el tipo de proceso que se realizará si la oración es subjetiva u objetiva. Si la oración es subjetiva se analizará de lo contrario se omitirá. Una vez identificadas las oraciones subjetivas, se procede a clasificarlas en positivas o negativas. Para este tipo de enfoque a nivel de oración se utiliza comúnmente el aprendizaje supervisado o aprendizaje no supervisado.
- c) **Análisis de sentimiento basado en aspectos:** Hay ocasiones que el documento contiene opiniones de una entidad que tiene muchos rasgos y cada uno de estos rasgos tiene su propia opinión. Para este tipo de casos se utiliza el análisis de sentimientos basado en aspectos (o análisis de sentimientos basado en atributos). Existen varios enfoques, el más común, es donde se identifican y se extraen del documento, todos los aspectos en un conjunto de datos de opiniones de productos obteniendo las frases nominales que presentan una frecuencia por encima del umbral definido. Otro enfoque utilizado es cuando se presentan los aspectos implícitos en las oraciones que

son los aspectos que no están escritos textualmente en la oración pero que se sobreentienden porque se mencionan implícitamente en las expresiones de sentimiento. Después estos aspectos implícitos se extraen de las oraciones y se hacen coincidir con los aspectos explícitos mediante una asociación de concurrencia de dos frases. Por último, mediante el aprendizaje supervisado o no supervisado se determina la polaridad de cada expresión de sentimiento tomando en cuenta el léxico.

- d) Análisis de sentimiento comparativo: El objetivo del sistema de análisis de sentimientos en este enfoque es señalar las oraciones que poseen las opiniones comparativas. Luego se extrae la entidad o entidades preferidas en cada opinión. Para esto se utiliza los adjetivos comparativos adverbios como: ‘más’, ‘menos’ o frases como ‘favorece’, ‘sobrepasa’, ‘supera’, ‘prefiere’, ‘mejor que’, ‘superior’.
- e) Adquisición del léxico de sentimientos: El léxico de sentimientos es el recurso principal los algoritmos de análisis de sentimientos. El léxico de sentimientos se obtiene a partir de tres maneras: el método en donde las personas codifican manualmente el léxico, a este método se le llama el método manual, otro enfoque es con la técnica basada en diccionario, donde se crea un grupo de palabras base, utilizando herramientas como *WordNet* (Fellbaum, 1998) y por último el método basado en conjunto de datos (corpus) en donde un pequeño número de palabras semilla se expanden a través del uso de conjunto de datos enormes de algún dominio.

La arquitectura de un sistema típico de análisis de sentimientos (mostrado en la Figura 2-3) para Feldman (2013) comienza desde la entrada del documento que contiene la opinión de la entidad expresada de forma escrita. Luego al texto del documento se le realiza un preprocesamiento con diversos recursos o técnicas léxicas o lingüísticas como la atomización (*tokenization* en inglés), derivación (*stemming* en inglés), etiquetado del discurso, y extracción de características o entidades. Menciona que, dentro del sistema, el componente principal es el módulo de análisis de documento, el cual también hace uso de recursos léxicos y lingüísticos para analizar los sentimientos del documento preprocesado en forma de anotaciones. Después dependiendo el tipo de opinión expresada en el documento, se adjuntará al enfoque de análisis adecuado pudiendo ser: análisis de sentimientos por documento, análisis de sentimientos por oración, análisis de sentimientos por entidad o

análisis de sentimientos comparativo. Una vez terminado el análisis, estas anotaciones son presentadas al autor de la opinión en alguna interfaz o herramienta de visualización.



*Figura 2-3. Arquitectura de un sistema típico de análisis de sentimientos.*

### **2.1.3. Personalidad**

Hoy en día, conocer los rasgos de personalidad es muy importante ya que permite comprender mejor el comportamiento y la manera de actuar de los individuos. Por ejemplo, ubicar a una persona en un puesto funcional dentro de una organización o alcanzar un mejor aprovechamiento académico de los alumnos en las escuelas o mejorar el desempeño de un individuo durante su desarrollo cognitivo o profesional.

Para Funder (2013) hablar personalidad es todo lo relacionado con los patrones de pensamiento, emociones y comportamientos característicos de un ser humano, tomando también en cuenta los mecanismos psicológicos detrás de estos patrones ya sea que estén ocultos o no. La psicología se tiene que valer de todos sus subcampos de estudio para poder explicar la personalidad de los seres humanos y tener una visión integral de las personas en su entorno cotidiano.

Personalidad proviene del latín persona, que se refiere a la máscara que los actores usaban en alguna escena teatral; con el tiempo, el término persona se relacionó con todos los aspectos externos de un individuo o el tipo de imagen o forma de ser que refleja hacia las demás personas. Por lo que, etimológicamente hablando, se llega a la conclusión de que el término personalidad hace referencia a las características o aspectos externos y visibles que la gente

percibe de otra persona, dicho en otras palabras, es la impresión que la gente tiene de otro ser humano (Schultz & Schultz, 2013).

### 2.1.3.1. Teorías de la personalidad

Entender la personalidad mediante la psicología, resulta extremadamente difícil o casi imposible de hacer ya que tratar de entender a un ser humano es imposible. Funder (2013) menciona que la única alternativa es separar lo que vas a observar. Para esto menciona que en vez de mirar todo al mismo tiempo, se deben buscar los patrones más específicos y encontrar la manera de ligarlos con las observaciones. Una vez hecho esto, es necesario poner límites a algunos tipos de observaciones, ciertos tipos de patrones y algunos pensamientos de estos patrones. Una de las ramas de estudio de la psicología que profundiza en este tema es: la psicología de la personalidad. Este subcampo de la psicología estudia la personalidad en varios enfoques básicos (Funder, 2013), los cuales se describen la Tabla 2-1.

*Tabla 2-1. Enfoques básicos de la psicología de la personalidad (Funder, 2013).*

Tipos de enfoque básicos de la psicología de la personalidad	
Nombre Enfoque	Descripción
Enfoque rasgos de personalidad	Enfoque basado en las diferencias psicológicas de las personas, así como la medición de estas diferencias.
Enfoque Biológico	Enfoque basado en entender la mente mediante la anatomía, fisiología, genética y evolución del cuerpo humano.
Enfoque Psicoanalítico	Enfoque sobre el inconsciente de la mente de las personas. De su la naturaleza y capacidad de resolución de conflictos mentales internos.
Enfoque Fenomenológico	Enfoque basado en la experiencia consciente de la vida de las personas, así como de su fenomenología.
Enfoque del aprendizaje	Enfoque basado en la teoría de aprendizaje social, en los cambios de comportamiento de los individuos como reacción a un castigo o premio u otras experiencias en la vida y en la psicología cognitiva de la personalidad

Dentro del enfoque de rasgos de la personalidad, existen modelos o teorías de personalidad. Los cuales buscan conocer la personalidad dividiéndola en rasgos con la finalidad de asignar una medición a cada uno de ellos. Los cinco modelos enfoque de rasgos de la personalidad más importantes son los siguientes (John & Srivastava, 1999):

- La teoría de los rasgos de Allport: Gordon Allport planteó la hipótesis de que las características personales se pueden codificar en el lenguaje y las más importantes de éstas se deben describir en una sola palabra conocida como hipótesis léxica. Analizando dicha hipótesis, Allport junto con Odbert, extrajeron 4,500 palabras de

un diccionario de alrededor de 18,000 palabras que se podrían utilizar para describir a los rasgos de personalidad y las agruparon en tres categorías (Allport & Odbert, 1936; Allport, 1937):

- 1) Rasgos de personalidad
  - 2) Estados, actitudes, emociones y estado de ánimo actual
  - 3) Valoraciones sociales
- 16 factores de personalidad de Cattell: El químico Raymond Cattell hizo una distinción entre rasgos superficiales y rasgos fuente. Para Cattell, los rasgos superficiales son los comportamientos superficiales y se pueden observar a simple vista en las personas, por otro lado, los rasgos fuente son estructuras psicológicas más complicadas y dan el origen a los rasgos superficiales y dan una explicación a sus correlaciones. Basado en la hipótesis léxica de Allport, Cattell desarrollo y aplicó técnicas de análisis factorial, así como un análisis sistemático de diferentes fuentes de datos para reducir la lista de Allport en un pequeño número de rasgos esenciales, esta lista estaba conformada por más de 4,500 palabras. La idea de Cattell era que los rasgos psicológicos básicos y principales (Tabla 2-2) pueden describirse completamente mediante 16 rasgos primarios o rasgos fuente (Cattell et al., 1970).

*Tabla 2-2. 16 factores primarios y los cinco rasgos de personalidad globales.*

Extroversión/ Introversión	Alta ansiedad/ Baja Ansiedad	Fortaleza mental/ Receptividad	Independencia/ Comodidad	Autocontrol/ Falta de moderación
Cálido- Reservado	Emocionalmente Estable- Reactivo	Cálido –Reservado	Dominante -Obediente	Animado- Serio
Animado- Serio	Cauteloso- Confiado	Sensible-Insensible	Atrevido- Tímido	Consciente de las reglas- Ventajoso
Audaz- Tímido	Aprensivo- Seguro de sí mismo	Pensativo-Práctico	Vigilante- Confiado	Pensativo-Práctico
Privado- franco Autosuficiente- Orientado al grupo	Aprensivo- Seguro de sí mismo	Abierto al cambio- Tradicional	Abierto a cambio- Tradicional	Perfeccionista – Tolera desorden

Análisis posteriores del mismo Cattell, resultaron en señalar cinco rasgos de personalidad globales, (Cinco Grandes Originales):

- 1) Extroversión/Introversión
- 2) Ansiedad alta/Ansiedad baja
- 3) Mentalidad dura /Receptividad
- 4) Independencia/Acomodación

### 5) Autocontrol/Falta de Restricción

- Las tres dimensiones de la personalidad de Eysenck: Hans Eysenck propuso este modelo también conocido como el Tres Gigantes. Al igual que Cattell, Eysenck mediante un análisis factorial identificó tres superfactores derivados de los rasgos de personalidad comunes y los factores de orden superior: la extraversión, el neuroticismo y el psicoticismo (Eysenck & Himmelweit, 1947; Eysenck & Eysenck, 1976). En la Tabla 2-3, se observa la diferencia entre los tres superfactores y los rasgos primarios que conforman a cada uno de estos.

*Tabla 2-3. Tres Gigantes y rasgos primarios propuestos por Eysenck.*

Superfactores	Rasgos primarios
Extraversión	Antisocial, creativo, agresivo, impersonal, impulsivo, frío, egocéntrico, sin empatía
Neuroticismo	Animado, asertivo, ávido de sensaciones, emprendedor, sociable, despreocupado, activo, dominante
Psicoticismo	Ansioso, deprimido, sentimientos de culpa, baja autoestima, tenso, irracional, tímido, malhumorado, emocional

- El indicador de tipo Myers-Briggs (MBTI): Con base a la teoría de Carl Gustav Jung que clasificaba a las personas en tres aspectos: 1) Extroversión versus Introversión; 2) Sensación versus Intuición y 3) Prejuicio versus Percepción; Katherine Cook-Briggs e Isabel Briggs Myers crearon el MBTI (Myers et al., 1998) con cuatro dimensiones, donde se clasifican los tipos de personalidad específicos y combina las cuatro dimensiones dominantes. Las cuatro dimensiones MBTI son las siguientes:

- 1) Extroversión (E)–(I) Introversión
- 2) Sensación (S)–(N) Intuición
- 3) Pensamiento (T)–(F) Sentimiento
- 4) Prejuicio (J)–(P) Percepción

Los 16 tipos de rasgos están asociados con un tipo específico de rasgo de personalidad. Los resultados del MBTI se muestran en un código de cuatro letras, cada una de estas representa las categorías dominantes de cada dimensión, por lo que, de acuerdo con este modelo, sólo puede haber 16 tipos de personalidad aparente. Este modelo ha sido duramente criticado por ser demasiado simple para reconocer las

diferencias individuales y además es muy cuestionable la confiabilidad y validez de sus resultados al no poder explicar los resultados de la vida real.

- **Modelo de los Cinco grandes (*Big Five Personality Traits*):** Este modelo de rasgos de personalidad es el más importante y aceptado por la comunidad científica de nuestros tiempos para estudios de reconocimiento de personalidad, por lo que esta investigación está basada en este modelo. Se abordará este tema más a profundidad en la siguiente sección.

#### **2.1.3.2. Modelo de los cinco grandes (*Big-Five*)**

Después de los trabajos de Allport basados en la hipótesis léxica y con ayuda de Odbert, se reducen de 18,000 palabras descriptivas de la personalidad a 4,500 (Allport & Odbert, 1936). Cattell siguió reduciendo aún más esta lista, a un total de 35 rasgos. De estos términos, Fiske (1949) estructuró de una manera más simple reduciendo los rasgos a 22. El resultado de esta reestructuración más tarde se le denominaría como los Cinco Grandes. Estos factores fueron refinados y reanalizados en diversos estudios de Tupes & Christal (1961), encontrando cinco factores recurrentes y relativamente fuertes. Más tarde Goldberg (1981) nombraría estos factores como “*Big-Five*” (Cinco grandes). Cada uno de estos factores posee un amplio tamaño y resume una cantidad enorme de características de personalidad distintas para cada dimensión.

El modelo *Big-Five* también es conocido con el acrónimo de *OCEAN* (por sus siglas en inglés) correspondiente a cada dimensión de la personalidad. A continuación, se describirán los cinco rasgos de personalidad de este modelo (Tkalčič et al., 2016):

- **Apertura a la experiencia (*Openness to Experience*):** Se refiere a las personas que son creativas y de mente abierta e inteligentes. Personas con apertura están dispuestos a realizar actividades nuevas, son personas individualistas y poco convencionales, con atracción a cosas inusuales y las personas con baja apertura se inclinan más por lo tradicional y conservador.
- **Conciencia (*Conscientiousness*):** Son las personas organizadas, son coherentes, y tienen control sobre sus impulsos. Personas con alto puntaje en esta dimensión son perfeccionistas, organizadas, responsables y por el contrario personas con bajo puntaje son impulsivas, descuidadas, desorganizadas y distraídas.

- Extroversión (*Extraversion*): Las personas extrovertidas les gustan las emociones y buscan estimulaciones constantes, así como disfrutan estar en compañía de otras personas. Individuos con puntajes altos en extroversión poseen buenas habilidades comunicativas y sociales, son activos, optimistas o valientes. Por su contraparte, personas poco extrovertidas tienden a ser tímidas, discretas, calladas, mesuradas o reservadas.
- Amabilidad (*Agreeableness*): Esta dimensión está basada en las buenas relaciones sociales con otros seres humanos, cooperación y empatía. Las personas amables son generosas, simpáticas, bondadosas, comprensivas y confiables, pero las personas poco amables son testarudas, agresivas, competitivas debido a la poca seguridad en sí mismas.
- Neuroticismo o Estabilidad Emocional (*Neuroticism*): Este factor se caracteriza por la presencia de emociones negativas y explica la manera en la que gente confronta las situaciones estresantes de la vida cotidiana. Las personas neuróticas tienden a ser nerviosas, mal humoradas, ansiosas, impacientes e incluso pueden llegar a sentirse infelices. Los individuos con puntaje bajo o con estabilidad emocional son optimistas, temples, racionales y seguras de sí mismas.

### **2.1.3.3. International Personality Item Pool (IPIP)**

El modelo de rasgos de personalidad de los Cinco Grandes tiene como enfoque la medición de cada uno de los factores de personalidad. Para la medición de estas dimensiones se utilizan cuestionarios de dominio público basados en reactivos o preguntas (también conocidos en psicología como ítems), por ejemplo, el *International Personality Item Pool* (IPIP) (Goldberg, 1999). IPIP es un laboratorio científico donde se desarrollan métodos o técnicas avanzadas para medir los factores de personalidad. También contienen un inventario de miles de ítems y cientos de escalas para la medición de los rasgos de personalidad (International Personality Item Pool, 2021). En los reactivos del cuestionario, se les plantea a los participantes un evento o situación de la vida real y donde elegir (en una escala de cinco puntos) la opción que se adapte más a su manera de actuar o comportarse.



#### **2.1.3.4. Reconocimiento automático de personalidad**

Para el reconocimiento automático de rasgos de personalidad se utilizan métodos de aprendizaje máquina donde a partir de un conjunto de datos, se puede determinar si una persona presenta o no determinado rasgo de personalidad.

Un ejemplo de este tipo de trabajo es el realizado por Kampman et al. (2018) donde se creó una arquitectura multimodal para calcular las puntuaciones de los rasgos de personalidad a partir de audios, videos y texto. En su método utilizan varias capas de Redes Neuronales Convolucionales de manera individual para cada tipo de archivo. Por otra parte, muestran una fusión multimodal de las tres redes obteniendo mejores resultados en el reconocimiento de personalidad.

Para reconocimiento de factores de personalidad con enfoque en el audio hay trabajos como el de Sarkar et al. (2014) en el cual se basa en una técnica de aprendizaje máquina conocida como regresión logística con orientación a la clasificación. Para el entrenamiento utilizaron el conjunto de datos de personalidad de YouTube, de donde extrajeron características audiovisuales, demográficas y de sentimientos para realizar la clasificación de los rasgos de personalidad.

En el ámbito de reconocimiento de los cinco factores de personalidad a través del texto existen trabajos que utilizan métodos basados en características lingüísticas y métodos de aprendizaje máquina comunes o aprendizaje profundo. En el trabajo de Liu et al. (2016) se utilizan los caracteres de los textos para elaborar vectores de palabras para el cálculo de características. El conjunto de datos de texto para el entrenamiento del modelo fue obtenido de la plataforma *Twitter* y el modelo de aprendizaje profundo fueron unas redes neuronales recurrente (RNN).

Finalmente, los archivos de video pueden ser utilizados para extraer características físicas del rostro y expresiones emocionales de personas con la finalidad de reconocer la personalidad de los individuos. En la investigación de Gürpınar et al. (2016), se presenta un método que extrae las expresiones faciales y emocionales que una persona genera en un determinado contexto. Para esto utilizaron redes neuronales convolucionales profundas (DCNN) las cuales fueron entrenadas previamente para reconocimiento facial, de emociones e información del ambiente. Después hicieron una fusión de los conjuntos de las

características de expresiones faciales y las características de la escena de la primera imagen de cada video para alimentar a un modelo de regresión el cual realizó la clasificación de personalidad. La red regresiva utilizada se llama *Kernel ELM* (Huang et al., 2011), es muy similar al *Support Vector Machine* (SVM), pero su aprendizaje es más rápido.

## **2.2. Procesamiento Natural de Lenguaje**

En el campo de las ciencias computacionales, para que una máquina pueda entender los textos y los algoritmos puedan realizar alguna clase de operación o análisis, es necesario convertir los textos en un formato que la máquina pueda entender como lo son los números o el código binario. El Procesamiento de Lenguaje Natural (NLP por sus siglas en inglés) es el área de investigación donde se desarrollan métodos que permitan a las computadoras interpretar el lenguaje de los seres humanos (Kulkarni & Shivananda, 2019).

El Procesamiento de Lenguaje Natural (*Natural Language Processing*) para Chowdhury (2003), es un campo de investigación que combina áreas de estudio como las matemáticas, ciencias computacionales, inteligencia artificial, psicología, lingüística, etc. NLP tiene como objetivo la investigación de técnicas para hacer que las computadoras comprendan el lenguaje humano a través de expresiones de texto. Para ello tiene diversos enfoques de estudio como el reconocimiento de voz o la traducción automática de textos entre otros.

## **2.3. Aprendizaje máquina**

De acuerdo con Conway & White (2012), el aprendizaje máquina (*ML* por sus siglas en inglés) es hablar de un conjunto de varios métodos, técnicas o herramientas creadas para calcular, estimar, reconocer patrones o extraer de información y características principales de algún objeto del mundo real en un determinado contexto. Dicho de otro modo, el aprendizaje máquina es el medio por el cual los humanos le enseñan a las computadoras a entender un tema del mundo real, con el objetivo de que las computadoras usen el conocimiento aprendido para hacer tareas más complejas.

En la obra de Negnevitsky (2011) se menciona que el *ML* envuelve mecanismos adaptativos que le brindan la posibilidad a las computadoras de aprender analógicamente algún tema o

tarea específica, con base en ejemplos del mismo tipo. Esta capacidad de aprender por parte de las computadoras puede mejorar el desempeño de un sistema inteligente con el paso del tiempo. Los tipos de métodos más destacables del aprendizaje máquina son las redes neuronales y los algoritmos genéticos.

Los algoritmos de aprendizaje máquina, se basan en reglas para realizar diversas tareas que tienen que ver con procesamiento de datos tomando como base ejemplos del resultado que se espera. Para Chollet (2017) en un algoritmo de aprendizaje máquina se necesitan tres componentes:

1. Puntos de datos de entrada: Se refiere a los archivos (videos, imágenes, audios) de donde el algoritmo va aprenderá a calcular un resultado. Es decir, si la tarea es reconocimiento facial, los datos deberán ser imágenes de rostro.
2. Ejemplos de resultado esperado: se refiere a que los archivos ingresados en los algoritmos deben tener relación con el resultado esperado y debe estar identificado o etiquetado para que se puede correlacionar los datos.
3. Por último, se necesita una forma de saber si el algoritmo hizo bien la tarea para el que fue creado. Para esto se desarrollan métricas que comparan los resultados obtenidos (salida actual), con los resultados esperados (salida esperada).

### **2.3.1. Aprendizaje no supervisado**

En el aprendizaje no supervisado se busca el procesamiento de datos de entrada sin ayuda de objetivos; para esto, los datos deben ser comprendidos, transformados y en su defecto eliminados, con el objetivo de entender de una mejor manera las relaciones entre los datos. El aprendizaje no supervisado es muy importante para el análisis de datos, ya que es de gran ayuda para resolver problemas donde se tiene que conocer o identificar las características de un conjunto de datos antes de intentar resolverlo por medio de aprendizaje supervisado (Chollet, 2017). Ejemplos muy conocidos dentro de aprendizaje no supervisado son la reducción y la agrupación (*Clustering*).

1. La agrupación en clústeres es una técnica que hace uso de la medición de la distancia en el movimiento iterativo de elementos semejantes que se encuentran cerca en relación con los demás. Una vez realizado el agrupamiento, los elementos agrupados de manera más compacta alrededor de  $n$  centroides, entran en la categoría de ese

grupo. Dentro de los tipos de agrupamiento existentes en el aprendizaje no supervisado el más conocido es el agrupamiento *K-means* (Patterson & Gibson, 2017).

2. El objetivo de la reducción de dimensión es disminuir un conjunto de datos grande, destinándolo a un espacio de menores dimensiones. Para ello, se debe identificar las variables más importantes cuyo tamaño de información sea lo suficientemente basto y usarlas para hacer el análisis (Swamynathan, 2017).

### **2.3.2. Aprendizaje supervisado**

Los algoritmos o modelos de aprendizaje supervisado tienen dentro de sus tareas la descripción, clasificación, distinción, predicción de clases o datos. Dicho en otras palabras, un modelo de aprendizaje supervisado puede ser capaz de aprender a partir de los datos o información proporcionada y debe poder realizar estimaciones o cálculos con nuevos datos desconocidos. Entre los algoritmos más comunes de este tipo son los algoritmos basados en reglas son: los árboles de decisión, redes Bayesianas, las máquinas de soporte vectorial (SVM) y las redes neuronales (Osisanwo et al., 2017).

#### **2.3.2.1. Árboles de decisión**

Los árboles de decisión tienen como objetivo crear valores que permitan predecir los resultados de una clase específica o variables esperadas a través de una serie de reglas de decisión de aprendizaje provenientes de datos anteriores; normalmente los árboles de decisión se utilizan en problemas de clasificación y regresión (Thomas & Galambos, 2004). En los árboles de decisión los nodos, están etiquetados por conceptos, las vertientes (ramas) están etiquetadas por los pesos de estas, mientras que las hojas, que son la parte final del árbol, conciernen a las categorías (Michalski et al., 2013). La Figura 2-4 muestra un ejemplo de árbol de decisión correspondiente a una situación de diagnóstico médico.

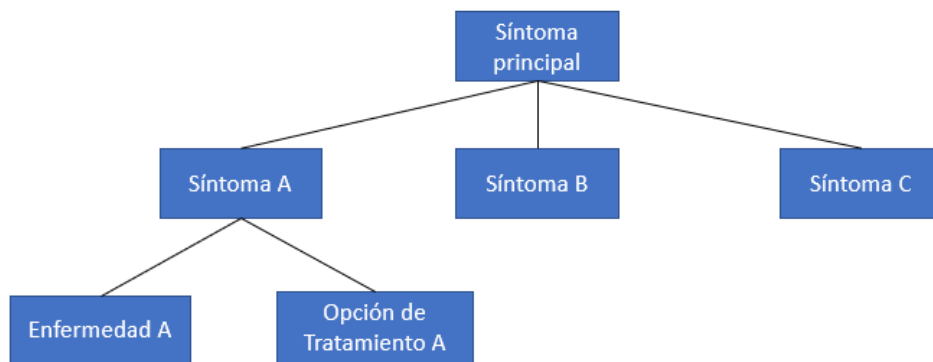


Figura 2-4. Ejemplo de un árbol de decisión.

### 2.3.2.2. Algoritmos Basados en Reglas

Los algoritmos basados en reglas en general comparten una gran semejanza a los árboles de decisión en la manera clasificar, con la diferencia que, en estos algoritmos se crean una serie de reglas que conducen a una clasificación más compacta de los documentos para cada una de las categorías, sin necesidad de tener que recorrer todo el árbol (Kirk, 2014).

### 2.3.2.3. Redes Bayesianas

Una Red Bayesiana es un algoritmo de tipo de aprendizaje estadístico que representa gráficamente las relaciones de probabilidad existente entre un determinado grupo de variables o características. Su principal característica que marca una diferencia con respecto a los árboles de decisión o las redes neuronales es que puede contar con información previa referente a las relaciones estructurales entre sus variables, acerca un problema suscitado. Para Kotsiantis et al. (2007) la estructura de las Redes Bayesianas (BN) está dada por los siguientes puntos:

- a) Definir que un nodo es un nodo raíz (no tiene padre)
- b) Definir que un nodo es un nodo hoja (no hereda hijos)
- c) Hay que declarar que un nodo es una causa directa de otro nodo
- d) Hay que señalar que un nodo no está conectado directamente con otro nodo
- e) Se debe exponer que dos nodos son independientes en ciertas condiciones
- f) Declarar la ubicación de los nodos, tener un orden parcial de los nodos
- g) Proveer un orden completo de los nodos

#### 2.3.2.4. Support Vector Machine

Las *Support Vector Machine* (SVM) son del tipo de clasificadores lineales, donde el algoritmo recibe como datos de entrada cierto tipo de información y el clasificador puede indicar la clase a la que pertenece con respecto a número de opciones posibles. De acuerdo con (Michalski et al., 2013) las SVM son estructuras que tiene como base la teoría estadística del aprendizaje y su función es transformar el espacio de entrada en otro de mayor tamaño (infinito) en el que el problema se puede resolver con el uso de un hiperplano idóneo. La Figura 2-5 se puede observar la clasificación de la SVM con el hiperplano antes mencionado.

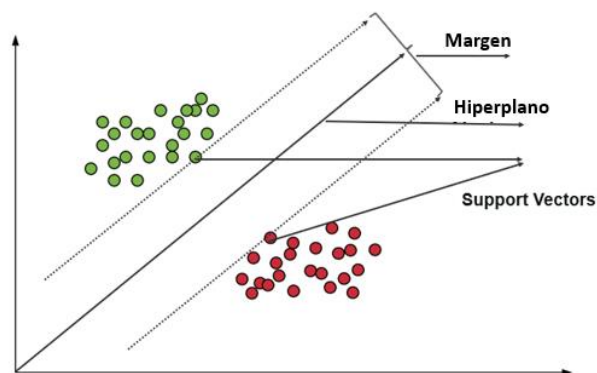


Figura 2-5. Ejemplo de clasificación con SVM.

#### 2.3.2.5. Redes Neuronales

Para Negnevitsky (2011) las Redes Neuronales Artificiales (ANN) constan de una serie de neuronas simples y altamente interconectadas, cuya función es muy similar a la de una neurona biológica del cerebro humano. Las neuronas están conectadas entre sí y transmiten señales entre ellas por medio de enlaces ponderados. Cada neurona recibe varias señales de entrada, pero solo genera una sola señal de salida. Esta señal de salida se divide en varias ramas y se transmite por medio de ellas hacia otras neuronas de la red. La señal enviada es la misma información para cada una de las neuronas (no se divide). La Figura 2-6 representa las conexiones de una ANN. Una Red Neuronal está compuesta de tres capas, la primera capa se llama capa de entrada y es ahí donde se ingresan los datos a la red, seguido por una o más capas ocultas o capa de en medio que es donde se realizan los cálculos y por último la capa de salida, donde se presenta la clasificación o valor esperado de la red neuronal.

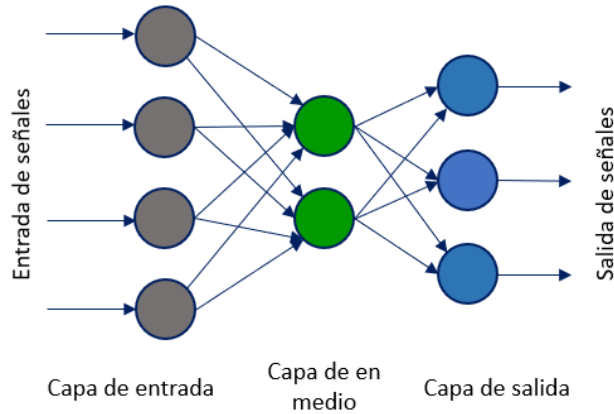


Figura 2-6. Arquitectura de una Red neuronal artificial.

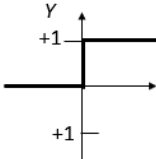
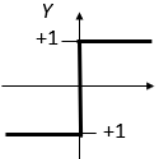
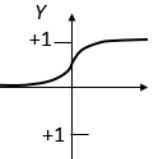
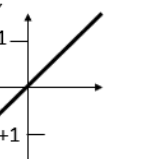
McCulloch & Pitts (1943) propusieron una idea para determinar el valor de la salida de la neurona. En esta idea la neurona calcula todos los pesos sumados de las señales de entrada y compara esta sumatoria con un umbral previamente definido,  $\theta$ . Si la entrada neta es menor que el umbral, la neurona se activa y su salida es -1, en caso contrario si la entrada neta es mayor o igual que el umbral, el valor será +1. Par este caso, en la siguiente expresión (1) se denota la función de activación de la neurona.

$$X = \sum_{i=1}^n x_i w_i \quad (1)$$

$$Y = \begin{cases} +1, & \text{IF } X \geq \theta \\ -1, & \text{IF } X < \theta \end{cases}$$

Donde  $X$  es la entrada ponderada hacia la neurona,  $x_i$  es el valor de la entrada  $i$ ,  $w_i$  corresponde al peso de la entrada  $i$ ,  $n$  se refiere al número de entradas de la neurona y  $Y$  es la salida de la neurona. Al tipo de expresiones como esta se les conoce como función de activación. Existen otros tipos de funciones de activación populares las cuales aparecen en la Tabla 2-4.

Tabla 2-4. Funciones de activación de las neuronas.

Función Step	Función Sign	Función Sigmoid	Función Linear
 $Y^{step} = \begin{cases} 1, & \text{IF } X \geq 0 \\ 0, & \text{IF } X < 0 \end{cases}$	 $Y^{sign} = \begin{cases} +1, & \text{IF } X \geq 0 \\ -1, & \text{IF } X < 0 \end{cases}$	 $Y^{sigmoid} = \frac{1}{1 + e^{-x}}$	 $Y^{linear} = X$

### 2.3.3. Aprendizaje profundo

La Inteligencia Artificial (IA), tiene sus orígenes varias décadas atrás y su misión principal es automatizar las tareas intelectuales que realizan los seres humanos cotidianamente. Para esto se vale del aprendizaje profundo (DL, *Deep Learning*) que es un subcampo del aprendizaje máquina y éste a su vez, es un subcampo de la IA (Chollet, 2017). En la Figura 2-7 se observa la relación entre la inteligencia artificial y el aprendizaje profundo.

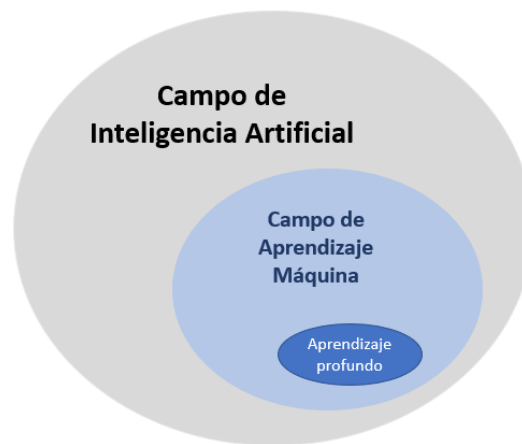


Figura 2-7. Relación entre la inteligencia artificial y el aprendizaje profundo.

La IA ofrece una solución a problemas que son difíciles para los seres humanos, pero sencillos para las computadoras, estos problemas se resuelven mediante el uso de las matemáticas formales (Goodfellow et al., 2017). Pero el verdadero reto para la IA es ofrecer una solución a las tareas que son sencillas para las personas, pero difíciles de describir, es decir, problemas que los humanos resuelven por intuición, como lo es el reconocimiento de palabras habladas, reconocimiento de emociones, o reconocimiento facial por fotografías.



Para LeCun et al. (2015), el aprendizaje profundo es una rama del *ML* que permite aprender con base a la experiencia a las computadoras y comprender el mundo como lo ven los humanos en términos de jerarquía de conceptos. El *DL* logra un gran poder y flexibilidad representando el mundo real en diferentes jerarquías anidadas de conceptos. La jerarquía de conceptos otorga la posibilidad a las computadoras de aprender conceptos difíciles creándolos a partir de otros conceptos más simples, un ejemplo de esto serían muchas capas densas.

De acuerdo con la definición de LeCun et al. (2015), el aprendizaje profundo les brinda a los modelos computacionales compuestos de varias capas, la capacidad de procesar y aprender representaciones o características de datos con múltiples niveles de abstracción. El aprendizaje profundo abre paso a nuevas estructuras muy complejas de grandes conjuntos de datos mediante el uso de algoritmos de propagación hacia atrás mostrando cómo una máquina debe manipular o hacer cambios en sus parámetros internos utilizados para calcular la representación en las capas partiendo de las representaciones de la capa anterior.

El ejemplo clásico de un modelo *DL* es la red profunda *feedforward* o perceptrón multicapa (*MLP*, *Multilayer Perceptrón*), se hablará más a detalle de esta red en la [sección 2.3.3.2]. Un *MLP* es solo una función matemática que procesa un conjunto de datos de entrada a datos de salida. La función está compuesta de muchas funciones más simples (Goodfellow et al., 2017).

Para Patterson & Gibson (2017), una red neuronal profunda puede ser definida como “una red neuronal con más de dos capas”. Señalan que una red *DL* no ocupa de una persona que determine manualmente las características importantes, sino que la extracción de características lo realiza automáticamente el mismo modelo *DL*. La extracción de características es el proceso que tiene la red neuronal de decidir qué características del *dataset* o conjunto de datos se puede o no, usar como identificadores para etiquetar los datos de forma confiable. Además, señalan ciertos puntos donde se observan las diferencias principales entre una red neuronal típica y una red *DL*, las cuales son:

- Posee más neuronas que las redes anteriores.
- Cuenta con formas más complejas de conectar las neuronas y las capas.

- Tienen mucho mayor desempeño debido poder computacional disponible para rendimiento.
- Realizan la extracción de características de manera automática.

Los modelos más populares de ANN con *DL* son el Perceptrón Multicapa (MLP, *Multilayer Perceptron*), las Redes Neuronales Recurrentes (RNN, *Recurrent Neural Network*), las Redes Neuronales Convolucionales (CNN, *Convolutional Neural Network*) y las redes de memoria de corto plazo (LSTM, *Long Short Term Memory*).

### 2.3.3.1. Perceptrón Multicapa

Un *MLP* es una red neuronal profunda también llamada *feedforward neural networks* con una capa de entrada de neuronas, por lo menos una capa intermedia de neuronas o también llamada capa oculta y una capa de salida de neuronas. En la capa de entrada es donde recibe señales de entrada del mundo exterior y las envía a todas las neuronas que se encuentran en la capa oculta. La capa oculta detecta las características de los patrones de entrada y posteriormente son representadas por los pesos de las neuronas de la capa oculta. Y finalmente la capa de salida dicta el patrón de salida de toda la red (Negnevitsky, 2011). En la Figura 2-8 se observa un ejemplo de una red MLP con dos capas ocultas.

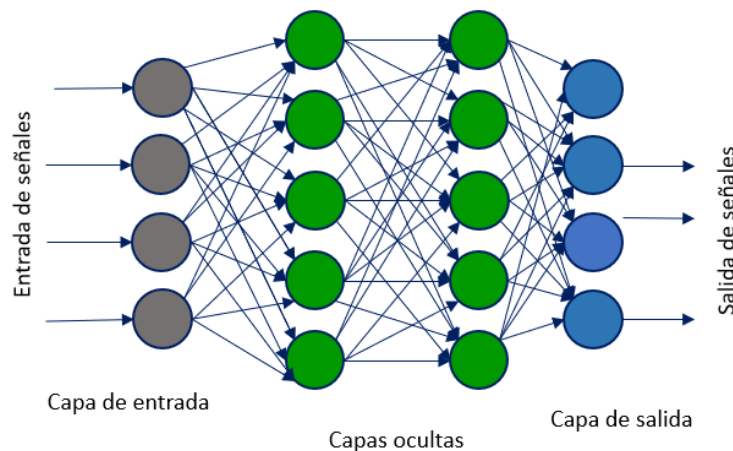


Figura 2-8. Perceptrón multicapas con dos capas ocultas.

El objetivo de las redes neuronales *feedforward* o ANN es aproximar la función  $f$  cualquiera. Por ejemplo, si se tiene un clasificador donde  $y = f(x; \theta)$  que mapea un valor de entrada  $x$  en alguna categoría  $y$ . La ANN a través de la función, aprende los mejores parámetros para  $\theta$  que resultan en la mejor función de aproximación. Se dice que las ANN son *feedforward*

porque la información va por medio de la función siendo evaluada desde  $x$ , después se realizan unos cálculos intermedios que servirán para definir  $f$  y por último la salida  $x$ . En este tipo de redes no hay retroalimentación ni conexiones en las salidas del modelo ya que estas se alimentan de nuevo a sí mismo (Goodfellow et al., 2017).

Algunos conceptos importantes que se deben de conocer de cualquier ANN son los siguientes (Patterson & Gibson, 2017):

- **Pesos de las conexiones:** son los valores que amplían o minimizan la señal de entrada a una neurona. En las imágenes representativas de las ANN son las líneas / flechas que van de un nodo a otro y su notación en una función se expresa como  $w$ .
- **Sesgo (*biases*):** los sesgos son valores escalares cuyo propósito es garantizar que al menos, ciertos nodos se activen por capa, sin tomar en cuenta la intensidad de la señal, los sesgos son los responsables de que se dé el aprendizaje en la red, pues al haber baja señal estos dan acción a la red. Su notación como  $b$  y al igual que los pesos se van modificando durante todo el proceso de aprendizaje.
- **Función de activación:** son las funciones que dictan el comportamiento de las neuronas. Las funciones de activación transforman el conjunto de pesos, sesgos y entradas. El resultado de esta transformación es enviado a la siguiente capa de nodos. Existen diferentes tipos de funciones de activación con cuya activación depende del rango establecido. Se dice que una neurona “se activó”, cuando una neurona pasa de un valor distinto de cero a otra neurona dentro de la red neuronal.

### **2.3.3.2. Redes Recurrentes**

Las redes neuronales recurrentes (*RNN*) para Rumelhart et al. (1986), son un tipo de redes neuronales que se especializan en procesar datos secuenciales. Una *RNN* tiene la capacidad de procesar una secuencia de valores que pueden ser de tamaño variable  $x^{(1)}, \dots, x^{(T)}$ ; y escalar estas secuencias a otras mucho más largas y de una manera más práctica a comparación de cualquier otra red neuronal que no se especialice en procesar datos en secuencia.

La interpretación de Graves (2012) señala que las ANN que forman conexiones cíclicas entre sus nodos y capas se les considera redes neuronales recurrentes (*RNN*). Un *MLP* solo puede mapear vectores de entrada a salida, por otro lado, un *RNN* puede mapear toda una historia o

secuencia de valores de entrada hacia cada salida. La clave principal en una *RNN* es que las conexiones recurrentes poseen una “memoria” de entradas previas para mantenerse en el estado interno de la red, la cual puede tener una influencia sobre la salida de la red. En la Figura 2-9 se puede observar una representación de una *RNN*.

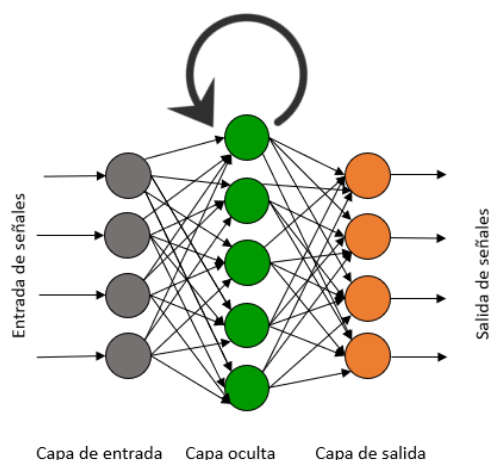


Figura 2-9. Red Neuronal Recurrente.

Las RNN se pueden utilizar para problemas de la clasificación, regresión y la generación de nuevos resultados. El problema que ha existido desde la creación de las RNN es que son difíciles de entrenar, ya que requieren más recursos computacionales que una ANN. Pero gracias a los avances tecnológicos e investigaciones recientes en el área de computación (optimización, paralelismo, arquitecturas de red y las unidades de procesamiento gráfico, GPU) han permitido que tanto los usuarios como investigadores, tengan fácil acceso al uso de estas redes y puedan realizar experimentos con datos de tipo secuencial.

Las *RNN* toman como entrada cada vector de secuencia de vectores y los modelan uno por uno. Esto le da la oportunidad a la RNN que mantenga su estado mientras que modela cada uno de los vectores de entrada en la venta de vectores de entrada. A eso se le conoce como modelado de dimensión de tiempo y es un sello característico de las redes neuronales recurrentes (Patterson & Gibson, 2017). Las redes neuronales recurrentes son ideales para modelar funciones en donde los vectores de entrada y salida tienen características de dependencia temporal entre los valores. Las *RNN* modelan el aspecto temporal de los datos ya que crean ciclos dentro de la red (por tal motivo estas redes tienen el nombre de “recurrente”).

### 2.3.3.3. Redes de memoria de corto plazo

Las redes de gran memoria de corto plazo (LSTM, *Long Short Term Memory*) es una variante de las RNN y fueron introducido en 1997, cuya idea se basa en la idea de que cada neurona realice un bucle, con la finalidad de generar caminos por los cuales el gradiente de aprendizaje fluya por largos periodos de tiempo Hochreiter & Schmidhuber (1997). Una representación de una red LSTM se observa en la Figura 2-10.

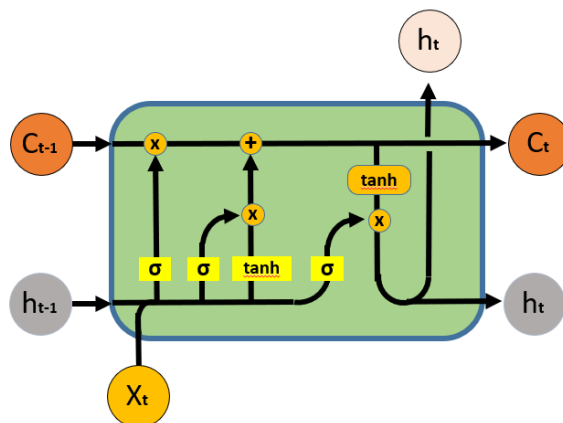


Figura 2-10. Red LSTM.

Los componentes principales de una LSTM son las puertas de olvido y puerta de entrada y la celda de memoria, cuyo contenido de este último es controlado por ambas puertas respectivamente. Gracias a la estructura de puerta es posible que la información se mantenga a lo largo de muchos pasos de tiempo. Una enorme ventaja de la LSTM es que evita el desvanecimiento o desaparición del gradiente, lo cual es muy común en otras redes neuronales recurrentes (Patterson & Gibson, 2017). Las LSTM son conocidas por tener mejor adaptación de ecuaciones y mejor propagación hacia atrás. Algunos casos de uso donde es recomendable usar redes LSTM son:

- Generación de oraciones
- Clasificación de series de tiempo
- Reconocimiento de voz
- Reconocimiento de escritura a mano
- Modelado de música polifónica

#### 2.3.3.4. Redes Convolucionales

El objetivo principal de las redes neuronales convolucionales (CNN, *Convolutional Neural Networks*), es el aprendizaje de características significativas de un conjunto de datos de forma de orden superior mediante convoluciones. Son altamente recomendables cuando se trata de problemas de clasificación de imágenes. Se puede realizar reconocimiento de rostros, de individuos, letreros de calles entre otros datos de tipo visual. También se pueden utilizar para análisis de textos por medio de reconocimiento óptico de caracteres y análisis de sonido (Patterson & Gibson, 2017).

Existen muchas variantes en la arquitectura de las redes CNN, pero en general poseen un patrón de capas principales las cuales son (Patterson & Gibson, 2017):

- 1) Capa de entrada (*Input Layer*): a diferencia de las redes ANN, la capa de entrada de las CNN acepta datos en la forma espacial de tres dimensiones (un solo vector unidimensional), ancho x alto y como adicional tiene una profundidad que representa los canales de color (*RGB*).
- 2) Capas de extracción de características (*Feature-extraction layers*): estas capas tienen un patrón repetitivo con la capa de secuencia: Capa convolucional (*Convolutional layer*) y la capa de agrupamiento (*Pooling layer*). Estas capas tienen tarea de encontrar cierto número de características en las imágenes para posteriormente construir características de orden superior. Esto representa a uno de los objetivos del DL que es aprendizaje automático de características y no de forma manual.
- 3) Capas de clasificación (*Classification layers*): las capas de clasificación pueden ser de una o más capas conectadas completamente (*fully connected*) a todas las neuronas de la capa anterior y su finalidad es tomar las características de orden superior y producir probabilidades de clase o puntaje. La salida de estas capas genera una salida bidimensional de dimensiones  $[b \times N]$  donde  $b$  representa el número de ejemplos del mini lote y  $N$  es el número de clases que se busca puntuar.

En la Figura 2-11 se muestran la arquitectura general de una red CNN.

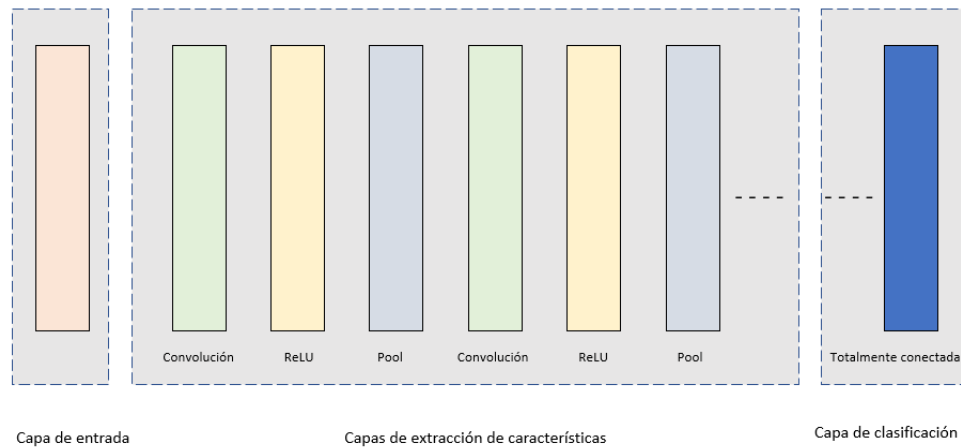


Figura 2-11. Arquitectura de una red CNN general de alto nivel (Patterson & Gibson, 2017).

A continuación, se describen algunos conceptos de las redes CNN para su mejor entendimiento.

## Convolución

Una convolución matemáticamente hablando es una operación que describe una regla para combinar dos conjuntos de información en uno solo por medio del uso de transformadas de Fourier. Recibe una entrada de datos se le aplica un núcleo (*kernel*) de convolución y arroja un mapa de características como salida (Patterson & Gibson, 2017). La operación de una convolución se puede observar en la Figura 2-12. Cada operación convolucional la determina el paso (*stride*), tamaño de filtro (*filter size*) y relleno de ceros (*zero padding*) (Albawi, S. et al., 2018).

- *Stride* (es un número positivo) determina el paso deslizante, es decir, si se tiene un *stride* de 1, significa que el núcleo o filtro se deslizará un lugar a la derecha cada convolución y luego se calculará la salida.
- Los filtros (*kernel*) es el núcleo que tiene un ancho y una altura más pequeños que el ancho y la altura del volumen de entrada (Patterson & Gibson, 2017).
- El relleno de ceros (*zero padding*) agregas filas y columnas a la matriz del volumen de entrada original con la finalidad de controlar el tamaño del mapa de características resultante. Es decir, el objetivo del *zero padding* es incluir datos en los bordes de la matriz de entrada ya que, de no existir este componente, la salida de la convolución es más pequeña que la entrada. El relleno de ceros evita que las redes se reduzcan

de tamaño y las proporciona ilimitadas capas profundas en la red neuronal (Albawi, S. et al., 2018).

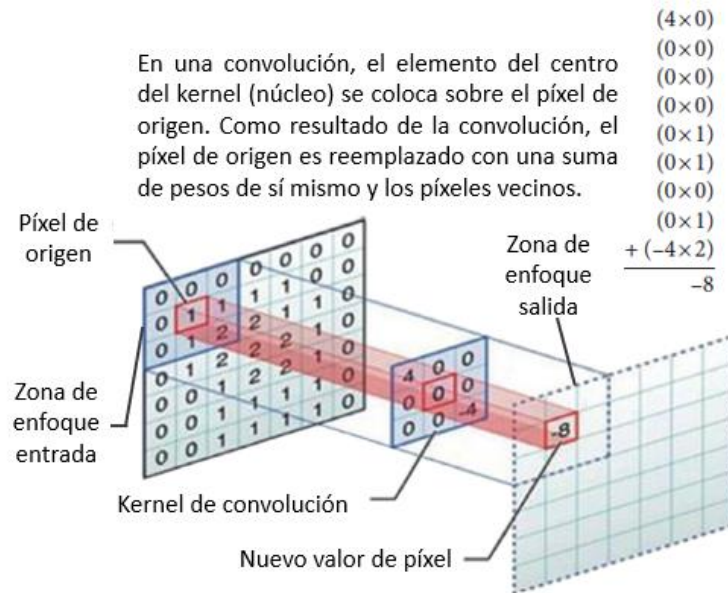


Figura 2-12. Desplazamiento el filtro sobre una entrada determinada para realizar la convolución (Albawi, S. et al., 2018).

La salida resultante de esta capa es la sumatoria de la multiplicación de elemento por elemento entre el *kernel* y los datos de la zona de enfoque obteniendo como resultado una sola entrada en el mapa de características de la capa de salida (Albawi & Ucan, 2018).



## Capas convolucionales

Las capas convolucionales son el componente más importante en la arquitectura de una red CNN. Esto es porque en las capas convolucionales, es donde se lleva a cabo la operación de convolución. Estas capas transforman los datos de entrada por medio de un parche de neuronas conectadas localmente a la previa capa. La capa convolucional realizará el cálculo de un producto escalar (Patterson & Gibson, 2017). Es decir, la matriz del *kernel* se multiplicará por la zona de enfoque de la capa de entrada. El resultado de esta multiplicación se colocará en el lugar correspondiente del centro de enfoque de la capa de salida. Después se puede deslizar la zona de enfoque y llenar los otros elementos del resultado de la convolución (Albawi & Ucan, 2018).

## Capas de agrupación (*pooling*)

Estas capas se encuentran normalmente entre las capas convolucionales. Esto es debido a que con las capas de agrupación (*pooling*) se reduce progresivamente el tamaño espacial (ancho y alto) de la representación de los datos y ayudan a controlar el sobreajuste (*overfitting*). Las capas de agrupación pueden utilizar varios tipos de filtros para realizar el proceso de reducción de resolución en el volumen. El filtro más utilizado para esta reducción de tamaño es la operación  $\max()$ . A esta operación se le conoce como agrupación máxima (*max pooling*). Esta operación cuenta con un filtro de  $2 \times 2$ , de donde se toma el mayor de cuatro números en el área del filtro. En la Figura 2-13 se observa un ejemplo de agrupación máxima.

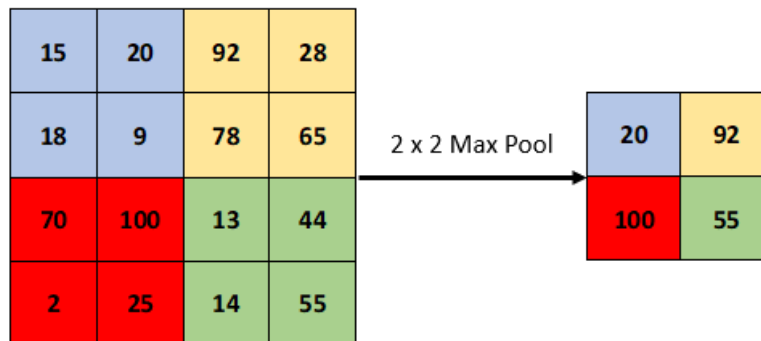


Figura 2-13. Ejemplo de una capa de pooling (Max pooling).

Una de las configuraciones más comúnmente usada en la capa de agrupación es tener el filtro *max pooling* con filtro de 2 x 2 y un paso 2 (Patterson & Gibson, 2017). Las capas de agrupación tienen otras características como:

- No tener parámetros para la capa.
- Utilizan hiperparámetros adicionales en lugar de parámetros.
- Esta capa calcula una función fija del volumen de entrada.
- No utilizan el relleno de ceros (*zero padding*) para agrupar capas.

#### **2.3.3.5. Aprendizaje profundo con datos en texto**

El texto, es una secuencia de caracteres o secuencia de palabras y es una de las formas más extendidas de datos de secuencia. El texto es utilizado por modelos de procesamiento de secuencias de aprendizaje profundo para tareas de clasificación de documentos, análisis de sentimientos o reconocimiento de emociones, identificación del autor o para respuestas a ciertas preguntas de un tema en específico.

El aprendizaje profundo hace uso del Procesamiento Natural de Lenguaje (NLP) para reconocimiento de patrones aplicado a las palabras, oraciones y párrafos. Las redes neuronales de *DL* no son capaces de realizar las tareas anteriores con el texto sin procesar, solo funcionan con tensores numéricos. La manera de poder trabajar el texto con redes neuronales es transformando el texto en tensores numéricos, a este proceso se le denomina vectorización de texto (Chollet, 2017). Para esto existen varias maneras:

- Separación del texto en palabras y convertir cada palabra en un vector.
- Segmentación el texto en caracteres y transformar cada carácter en un vector.
- Extraer N-gramas (grupos superpuestos de varias palabras o caracteres consecutivos) de las palabras o caracteres y transformar estos N-gramas en un vector.

En el mundo de la computación a las palabras, caracteres o N-gramas se les conoce como *tokens*. Y al proceso de dividir en el texto en tokens se denomina *tokenización*. Para cualquier proceso de vectorización de texto se aplica algún tipo de *tokenización*. Después se asocian esos vectores numéricos con los *tokens* que se generaron. Estos vectores se guardan como secuencia de tensores para poder introducirlos como datos de entrada para las redes

neuronales de *DL* (Chollet, 2017). En la Figura 2-14 se observa un ejemplo de vectorización de texto usando la *tokenización*.

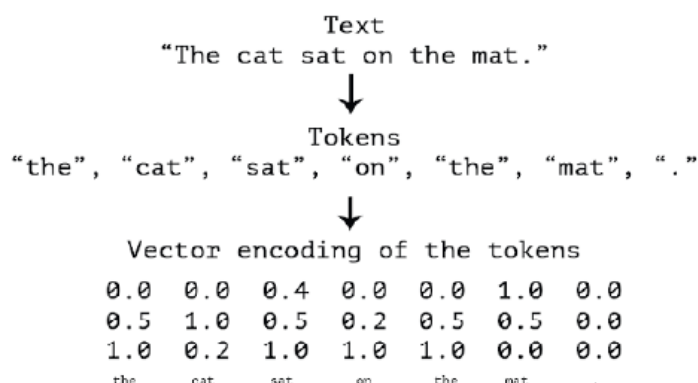


Figura 2-14. Vectorización de texto. Imagen de Chollet (2017).

Entre las formas más comunes utilizadas para asociar un vector a un token está la llamada codificación en caliente de *tokens* (*one-hot encoding of tokens*), las incrustaciones de palabras (*word embeddings*) y la bolsa de palabras (*bag of words*).

### Codificación *one-hot* de *tokens* (*one-hot encoding of tokens*)

El método *one-hot ending* es la forma más utilizada para convertir un token en un vector y consiste en asociar a cada palabra un índice entero único. Después este índice entero  $i$  único, se convierte en un vector binario de tamaño  $N$ , el tamaño del vocabulario, que sería todo ceros excepto por la  $i$ -ésima entrada, que sería 1. La técnica *one-hot ending* también se puede realizar a nivel de carácter (Chollet, 2017). En general, los vectores que se obtienen de *one-hot ending* tienen las siguientes características:

- Son vectores binarios.
- Son dispersos, es decir, en su mayoría están hechos de ceros.
- Son de muy alta dimensión (tienen la misma dimensionalidad que el número de palabras en el vocabulario).
- Los vectores producidos contienen 20,000 dimensiones o más (capturando un vocabulario de 20,000 tokens en este caso).

## **Incrustación de palabras (Word Embeddings)**

Otra técnica muy conocida para asociar un vector con una palabra es el uso de “vectores de palabras” o “incrustación de palabras” (*word embedding*). Los vectores obtenidos a través de la técnica *word embedding* son de punto flotante de baja dimensión, es decir, son vectores "densos" o compactos, caso contrario de los vectores obtenidos de *one-hot encoding*. En la técnica *words embeddings*, las incrustaciones de palabras se aprenden directamente de los mismos datos por lo que es común ver *word embeddings* de 256 dimensiones, 512 dimensiones o 1,024 dimensiones cuando se trata de vocabularios muy grandes. Debido a esta característica los vectores del modelo *words embedding* pueden almacenar más información en muchas menos dimensiones (Chollet, 2017).

## **Bolsa de palabras (BoW, Bag of Words)**

La técnica Bolsa de palabras (*BoW, Bag of Words*) tiene múltiples propósitos ya que se puede utilizarse como algoritmo para selección de características y clasificación de documentos e imágenes. En clasificación de documentos, un *BoW* es un vector de número de ocurrencias de palabras, que también se puede definir como la "representación de histograma basada en características independientes" (Qader et al., 2019).

En este método, los textos como oraciones o documentos se representan como la bolsa de sus propias palabras, y solo son consideradas las palabras duplicadas y se ignora tanto la gramática como el orden de las palabras. La técnica *BoW* se utiliza para entrenar a los clasificadores de textos donde las características se producen a partir de la frecuencia o aparición de cada palabra (Kim et al., 2017).

## **2.4. Conjunto de datos**

Lingüísticamente hablando, la palabra corpus es utilizada para referirse a cualquier tipo de recopilación de textos. Existen tres tipos de recolección de textos dependiendo del grado de los criterios de selección, los cuales son (Torruella & Llisterri, 1999):

- Archivo o colección: Es un conjunto de texto en soporte informático sin algún tipo de relación entre ellos.

- Biblioteca de Textos Electrónicos: Es una colección de textos de tipo informático que es guardado en un formato estándar con cierto tipo de normas o criterios poco rigurosos para la selección del contenido.
- Corpus Informatizado: Es un conjunto de textos seleccionados de acuerdo con criterios lingüísticos establecidos, codificados de modo estándar y homogéneo. Tiene como objetivo poder ser utilizados en procesos informáticos cuya finalidad es mostrar el comportamiento de uno o más idiomas.

Los dos primeros tipos de recopilaciones no implican una selección o una ordenación hecha siguiendo criterios lingüísticos, mientras que los corpus sí. Los criterios lingüísticos para la selección o una orden de los datos de los corpus pueden ser según Sinclair (1996):

- a) Externos: Si indica datos sobre el autor, el medio utilizado, el nivel social de los participantes, la función comunicativa del texto, etc.
- b) Internos: Cuando se refiere a los esquemas lingüísticos contenidos en los textos.

# Capítulo 3

## 3. Estado del arte

Esta sección se describe y analizan varios trabajos relacionados con los corpus de texto o conjuntos de datos. Los conjuntos de datos son utilizados para entrenar los clasificadores o modelos de *ML* tradicional y modelos *DL* para tareas de reconocimiento de personalidad automática. Estos conjuntos de datos de texto son en su mayoría de idioma diferente al español y algunos corpus son extracciones de textos de redes sociales como *Facebook* o *Twitter*. Dichos corpus, fueron considerados para el desarrollo e integración de este proyecto debido a que comparten el mismo objetivo de reconocimiento de personalidad mediante texto.

### 3.1. Corpus de texto utilizados en el reconocimiento de personalidad

En los últimos años se han realizado diversas investigaciones con el objetivo de reconocer la personalidad automáticamente a través del texto. El objetivo de este y otros trabajos similares, es que el clasificador o modelo reciba alguna frase o un párrafo de texto y automáticamente, arroje tu puntuación correspondiente a cada uno de los cinco rasgos de personalidad, es decir, los resultados muestran en forma de una escala determinada la proporción en que la persona posee el rasgo de personalidad. Por ejemplo, el clasificador tiene en una escala de 0 a 5 para el rasgo de extroversión (*extraversion*), donde 0 puede significar que no posee el rasgo y 5 que se tiene total presencia del atributo.

En las siguientes secciones se muestran algunos de los corpus de texto basados en modelos de reconocimiento de personalidad como el modelo *Big Five*. Por otro lado, se presentan algunas investigaciones relacionadas con modelos de aprendizaje automático y aprendizaje profundo donde hacen uso de corpus de texto para el entrenamiento y evaluación de sus modelos.

### 3.1.1. Essays

Essays es un conjunto de datos creado por Pennebaker & King (2000) y contiene aproximadamente 2,400 ensayos de flujo de conciencia etiquetados con los rasgos de personalidad de los autores. El corpus está organizado de la siguiente manera:

- Identificador de los autores.
- Texto de los ensayos de flujo de conciencia de los autores.
- Rasgos de personalidad etiquetados como (y/n) para cada uno de los cinco rasgos de personalidad: EXT (*Extraversion*), NEU (*Neuroticism*), AGR (*Agreeableness*), CON (*Conscientiousness*) y OPN (*Openness to Experience*).

### 3.1.2. MyPersonality

El corpus myPersonality desarrollado por Kosinski, Kosinski et al. (2015), es uno de los corpus de texto más populares y usados en pruebas de reconocimiento de personalidad con base a texto. Cuenta con alrededor de 10,000 estados en idioma inglés de 250 usuarios extraídos de la aplicación *Facebook* y cuyos datos se encuentran etiquetados de manera similar al conjunto de datos *essays*, con valores binarios (y/n) para indicar si tiene o no los rasgos de personalidad, así como también con sus respectivos valores de medición de personalidad obtenidos por pruebas IPIP correspondiente a cada autor.

### 3.1.3. B5

El corpus b5-post (Ramos et al., 2018) es un conjunto de datos que contiene 2.2 millones de palabras en idioma portugués, con 194,000 actualizaciones de estado de 1,019 usuarios de la plataforma de *Facebook*. Además, cuenta con autoevaluaciones llenadas por los mismos usuarios del inventario BFI-44. Este corpus cuenta con datos como:

- Identificador.
- Lugar de procedencia de texto (*Facebook* o texto controlado).
- Género.
- Edad.
- Religión.
- Los cinco rasgos de personalidad con sus respectivos valores de medición de personalidad.

#### **3.1.4. Pan2015**

Los corpus de PAN 2015 (Rangel et al., 2015) están compuestos por 4 conjuntos de datos en diferentes idiomas (español, inglés, italiano y holandés). Cada corpus tiene etiquetas de género: masculino y femenino, edad: 18-24, 25-34, 35-49, 50+ y también cuenta con la autoevaluación de los cinco valores de rasgos de personalidad *OCEAN*, obtenidos de la prueba en línea del inventario BFI-10 (Rammstedt & John, 2007), con valores entre -0,5 y 0,5.

#### **3.1.5. Otros conjuntos de datos de Redes sociales y plataformas de internet**

Existen trabajos de reconocimiento de personalidad donde se usaron conjuntos de datos de diferentes idiomas obtenidos de diferentes plataformas de internet o redes sociales como *Twitter*. Un ejemplo de este tipo de corpus es el corpus “EAR *dataset*” que fue creado por (Mehl et al., 2001) y contiene 118,259 palabras y 15,269 expresiones de conversación. Además, este conjunto de datos incluye autoevaluaciones, las cuales se obtuvieron pidiéndole a los voluntarios que llenaran un cuestionario de autoevaluación estándar. También el conjunto de datos almacena las calificaciones del modelo *Big-Five* de 18 observadores independientes. También se han hecho trabajos con el *Youtube personality dataset* presentado por (Biel et al., 2013), que contiene aproximadamente 400 videos de cámaras web de usuarios dedicados a subir videos a la plataforma de *YouTube*.

### **3.2. Modelos de reconocimiento de personalidad con base en texto**

Un ejemplo de reconocimiento de personalidad a través del uso de conjuntos de datos es el trabajo de Tinwala & Rauniyar (2021) donde usan el modelo *Big-Five* y el *dataset* llamado *essays* para entrenar una red Neuronal convolucional profunda (*DCNN*, *Deep Convolutional Neural Network*) y para ello realizan una extracción de características a nivel de documento con *word2vec* (Mikolov, et al., 2013a; Mikolov, et al. 2013b) y las características de Mairesse (Mairesse et al., 2007). El objetivo del modelo es clasificar la presencia o ausencia del rasgo de personalidad y se utilizó para la evaluación la métrica *F1 score*.

Como se mencionó anteriormente, la investigación de Xue et al. (2018) tiene como enfoque el reconocimiento de la personalidad de usuarios de redes sociales en línea por medio de los mensajes de texto. Para esto crearon una nueva estructura de red neuronal *DL* llamada



AttRCNN. Se trata de una red neuronal jerárquica variante de las redes *CNN*, capaz de aprender representaciones semánticas profundas de las publicaciones de texto de cada usuario en línea de las redes sociales. Después utilizando la herramienta desarrollada por (Pennebaker & King, 2000) llamada *LIWC (Linguistic Inquiry and Word Count)*, que es un software de análisis de texto principalmente utilizado en el área de la psicología, extraen las características lingüísticas basadas en diccionarios de psicología y combinan características semánticas profundas con características lingüísticas estadísticas tomadas directamente de las publicaciones de texto. Estas características son introducidas en algunos algoritmos de regresión tradicional para calcular el valor real de los cinco rasgos de personalidad del modelo *Big-Five*. Para el entrenamiento del modelo utilizan el conjunto de datos *myPersonality* y su métrica para la evaluación de su trabajo fue MAE (*Mean Absolute Error*) obteniendo un MAE no mayor de 0.45888.

En Giménez et al. (2017), para el reconocimiento de personalidad se propone una red *CNN* entrenada previamente con una representación de vectores de palabras (*word embeddings*) llamado modelo *GloVe* desarrollada por (Pennington et al., 2014). *GloVe* combina métodos de factorización de matrices globales con métodos de ventana de contexto local, lo que le permite aprender mejores características en trabajos donde no exista dependencia externa. Para realizar la evaluación del modelo se utilizó el corpus PAN-AP-2015 que se empleó en la tercera tarea compartida de perfilado de autor (*3rd Author Profiling shared task*). Este corpus consta de una recolección de textos extraídos de la red social *Twitter*. Los rasgos de personalidad se evaluaron con pruebas estandarizadas BFI-10 (Rammstedt & John, 2007) con una escala de  $-0,5$  y  $0,5$ . Finalmente para la evaluación de cada uno de los rasgos de personalidad de los modelos utilizaron la métrica de error cuadrático medio (RMSE). El mejor resultado en la tarea compartida fue de 0.1442 RMSE.

En el trabajo de Wang et al. (2020) se construyó un grafo de redes convolucionales de personalidad (*GCN, Graph Convolutional Networks*) que es una red neuronal multicapa que puede operar desde el grafo y puede calcular el vector de incrustaciones del nodo basado en las propiedades del nodo vecino (Li & Goldwasser, 2019). Este modelo consiste en tres capas: (1) la capa de *embedding layer*: donde la red recibe como entrada las representaciones *one-hot encoding* tanto para los usuarios, para las palabras y para los documentos. En esta

capa realiza el aprendizaje de estos datos. (2) después sigue la capa *GCN* donde se extraen características de interacción del grafo construido. Y finalmente (3) la capa de clasificación donde por medio de cinco clasificadores (uno para cada rasgo de personalidad) realiza el cálculo. Para entrenar y evaluar su investigación utilizaron dos diferentes conjuntos de datos: *myPersonality* que es un *dataset* creado por Kosinski et al. (2015) y *essays*. El mejor resultado obtenido en esta investigación fue 80% de exactitud.

Otra investigación que intenta reconocer la personalidad de un individuo es el de Tandra et al. (2017) que está basado en el modelo de personalidad *Big Five Model*. Los autores utilizaron el corpus *myPersonality* para realizar pruebas con métodos de *ML* tradicional y *DL*. En los modelos de aprendizaje máquina tradicional utilizaron algoritmos como *Support Vector Machine (SVM)*, *Logistic Regression*, *Naive Bayes*, *Gradient Boosting* y *Linear Discriminant Analysis (LDA)* y los modelos *Deep learning* usados fueron el *MLP (Multilayer Perceptron)*, *GRU (Gated Recurrent Unit)*, *LSTM (Long Short-Term Memory)*, *CNN 1D* que es una red neuronal convolucional unidimensional y por último una combinación de éstas dos últimas (*LSTM + CNN 1D*). Adicionalmente, para la extracción de características, es decir, para encontrar la relación entre las palabras con personalidad, se aplicaron funciones lingüísticas como *LIWC* y *SPLICE* (Moffitt et al., 2012) en modelos tradicionales de aprendizaje máquina y para modelos *DL* con incrustación de palabras se utilizó *GloVe*. Los resultados de este trabajo alcanzaron un 79.49% de exactitud.

En el trabajo de Siddique & Fung (2017) se propone un clasificador de palabras multilingüe. Hicieron un entrenamiento de incrustaciones de palabras bilingües no supervisadas a partir de un corpus en paralelo de idioma inglés y chino. Estos conjuntos de datos son: el corpus *myPersonality*, las transcripciones *Youtube personality dataset* y el *UM corpus* (Tian et al., 2014). Este último consta de textos en idioma chino e inglés. Estas representaciones de palabras entrenadas son las entradas para la red *CNN*, la cual realiza el mapeo de los rasgos de personalidad de los Cinco Grandes (*Big-Five*). Los resultados obtenidos fueron una puntuación *F-score* promedio de 66.1% en tareas multilingüe.

En la investigación de Santos & Paraboni (2019) se realizó una comparación de una serie de modelos de reconocimiento de facetas de personalidad por medio de texto. Estas facetas pueden inferirse por medio del inventario BFI-44 (John et al., 2008). que está basado en el

modelo *Big-Five*. Estas facetas son Asertividad y Actividad representando la categoría de Extraversión (*Extraversion*), Altruismo y Complacencia para Amabilidad (*Agreeableness*), Orden y Autodisciplina bajo Conciencia (*Conscientiousness*), Ansiedad y Depresión sobre Neuroticismo (*Neuroticism*), y por último la Estética e Ideas haciendo referencia a Apertura a la experiencia (*Openness to experience*). Para el entrenamiento y evaluación de sus modelos, utilizaron el corpus B5-post que contiene 2.2 millones de palabras en idioma brasileño recopilado de *Facebook*. Idearon unos modelos creados con base en la clasificación *Naïve Baye* y *word2vec* respectivamente, para comparar tres clasificadores binarios para cada una de las anteriores facetas a partir del texto. Estos son: (1) la técnica *BoW* de bolsa de palabras con las 3 mil palabras más repetidas del corpus; (2) *skip* o Vectores de palabras promedio obtenidos de un modelo *skip-gram-100* y por último (3) *cbow* o vectores de palabra promedio obtenidos de un modelo *cbow-1000*.

En el trabajo desarrollado por Majumder et al. (2017) se muestra un método donde, a partir del corpus *essays* (usado como datos de entrenamiento), se utiliza una red neuronal convolucional (*CNN*) de clasificación binaria para cada rasgo de personalidad con la misma estructura, cuya finalidad es clasificar si el rasgo de personalidad está presente o no. El proceso comienza con un preprocesamiento y filtrado de los datos de entrada, donde se transforman todas las oraciones de cada ensayo en vectores de características de n-gramas, seguido de una extracción de características utilizando *word2vec* donde concatenan estos vectores con las características semánticas por palabra y con las características de Mairesse (Mairesse et al., 2007), que es un conjunto de características a nivel de documento para la detección de personalidad. El resultado obtenido es una representación de longitud variable, el cual se introduce en la *CNN* donde se realiza la clasificación obteniendo como mejor resultado 59.38% de exactitud.

En el trabajo realizado por Pabón & Orozco, (2022) se utilizó el conjunto de datos llamado *Youtube personality dataset* que son transliteraciones de grabaciones de vídeo recopiladas de *YouTube* el cual contiene puntuaciones generadas automáticamente para los cinco grandes rasgos de personalidad, las cuales están dentro de la misma base de datos. Se utilizaron tres diferentes tipos de incrustación de palabras como *Glove*, *Word2Vec* y *BERT* (Devlin et al., 2018). Para utilizarlos en tres diferentes modelos de clasificación: (1) Modelo de Regresión:

para predecir la puntuación de cada rasgo de personalidad; (2) Modelo clasificación binaria: para reconocer entre presencia fuerte y débil de cada rasgo de personalidad y finalmente (3) un Modelo de clasificación triple: que detecta la clase de acuerdo con los tres tipos de niveles diferentes de manifestaciones de cada rasgo de personalidad ya sea bajo, medio o alto.

En el trabajo de Ríssola et al., (2019), crearon un modelo para la evaluación automática de personalidad de las conversaciones. Este conjunto de datos contiene extracciones de conversaciones cotidianas de 96 voluntarios que usan un dispositivo llamado *Electronically Activated Recorder (EAR corpus)*. El modelo presentado en la investigación consta de dos componentes y opera en dos niveles diferentes. El primer componente se trata de red neuronal cápsula compuesta de redes *LSTM* bidireccionales con el enfoque de extraer diferentes aspectos semánticos del enunciado y codificarlos en representaciones vectoriales de bajo nivel. El segundo componente trabaja a nivel de usuario y es el encargado de seleccionar entre diferentes características semánticas producidas en el primer componente para construir una representación de nivel superior de cada personalidad por medio de un clasificador no supervisado.

En la investigación de Ong et al. (2017) realizan pruebas de reconocimiento de personalidad basado en el modelo *OCEAN*, y para esto se utilizó *Twitter* como medio para recopilar información y crear la base de datos para entrenamiento de sus modelos.

En el trabajo de Xue et al. (2017) desarrollaron un nuevo algoritmo del *ML* llamado aprendizaje de distribución de etiquetas (*LDL, Label Distribution Learning*). Los textos son en lenguaje chino y se recopilaron de Sina Weibo que es un sitio de *microblogging* de actualizaciones, estados de usuarios y se aplicó la prueba inventario BFI-44 para obtener sus puntajes de personalidad. La extracción de características se realizó en 3 categorías: (1) características estáticas: que tenían pequeños cambios con el tiempo, como lo es el género, nombre, etc.; (2) características dinámicas: que son las características que cambiaban con el tiempo, como seguidores, etc. y por último (3) características de contenido: a este lo representan los blogs, características lingüísticas y psicológicas, etc. A cada instancia se le dio una etiqueta. Se usaron 8 algoritmos *LDL* como *KNN*, *Bayes* y *SVM*. Y finalmente se compararon con algunos algoritmos de *ML* como *Random Forest*, *M'5 Rules*, *Random Tree*,

*Support Vector Regression*, *Linear Regression* y *MLP*. El algoritmo de *LDL* con *Support Vector Machine* proporcionó la mayor precisión con base a la métrica de evaluación MAE.

Un trabajo reciente donde hacen uso de redes neuronales de tipo *DL* con el mismo enfoque que esta investigación es el de Mohades et al. (2021). Los autores hicieron un experimento para entrenar una red *CNN* con modificaciones en los filtros y para ello utilizaron el algoritmo *AdaBoost*, que considera la posibilidad de usar diferentes tamaños de filtros y con ello mejorar la clasificación para la tarea de reconocimiento de personalidad mediante texto. Para la evaluación utilizaron el conjunto de datos *essays* obteniendo resultados muy aceptables y en algunos casos superiores comparados con otros trabajos donde utilizaron técnicas de *ML* y *DL*.

También hay trabajos donde se utilizan dos corpus de texto tal es el caso del trabajo Sun et al., (2018) donde crean un modelo llamado *2CLSTM*, que incluye incrustación de palabras usando *GloVe* y *LSTM* bidireccionales concatenados con capas *CNN* con *kernels* (filtros) de 1,2,3 gramos y clasificación, para detectar la personalidad de los usuarios usando textos. El modelo consta de dos partes. Una *2LSTM* cuyo enfoque es juntar de manera correcta el contexto y extraer las características semánticas de las palabras y una *CNNLSG* que aprende las características de la estructura de las oraciones. Para las evaluaciones utilizaron dos conjuntos diferentes de datos que contienen textos largos y textos cortos: el *dataset essays* y el conjunto de datos de personalidad de *YouTube*.

En la investigación de Yu & Markov (2017) se utilizaron técnicas de *DL* para aprender la representación de datos de manera adecuada y automáticamente en tareas de reconocimiento de personalidad. Para ello usaron el corpus *myPersonality* que cuenta con publicaciones de usuarios de *Facebook*. Las pruebas consistieron en varias arquitecturas de redes neuronales como las redes completamente conectadas (*FC*, *fully-connected networks*), *CNN* y *RNN*. Los resultados que obtuvieron muestran un mejor desempeño por parte de la red *CNN* con agrupación promedio (*average pooling*) alcanzando un  $60,0 \pm 6,5\%$  con la métrica de evaluación *F1 score*.

# Capítulo 4

## 4. Desarrollo del proyecto

En este capítulo se describe el desarrollo del proceso de creación del corpus de texto, las técnicas utilizadas de aprendizaje máquina y profundo para su evaluación, y las tecnologías usadas para la implementación.

### 4.1. Una metodología para la creación del corpus *PersonText*

Muchos de los trabajos de investigación que hacen uso de técnicas de *ML* o *DL*, realizan sus entrenamientos con base a un corpus etiquetado con dos o más clases de un determinado tema o dominio. En esta investigación se definió a un corpus de texto como un conjunto de frases, opiniones u oraciones compuestas de palabras o caracteres. En estas oraciones los participantes manifiestan su opinión con respecto a algún tema en particular o también pueden expresar sentimientos o emociones.

Uno de los mayores problemas es que la mayoría de estos conjuntos de datos se encuentran en el idioma inglés. Por lo que surge la necesidad de crear un corpus de textos en español basado en el modelo *Big-Five* para ser utilizado en tareas de reconocimiento de personalidad automática.

En esta sección se muestra la metodología para la creación del corpus en español al que se le llamó *PersonText*, el cual es un conjunto de datos que contiene textos de opiniones de voluntarios, y datos de información personal de los mismos, como su edad, sexo y las evaluaciones de los cinco rasgos de personalidad del modelo *Big-Five*. Los datos del corpus *PersonText* se obtuvieron de una plataforma llamada *PersonApp* (Bátiz, 2021; Bátiz et al., 2022).

Cabe mencionar que tanto los textos, así como el contenido del corpus es completamente anónimo, por lo que se respeta la privacidad de los participantes. Se revisó cada texto y en las partes donde se menciona algún nombre, se omitió o en su defecto, fue remplazado por el siguiente acrónimo “\$NAME\$.”

#### 4.1.1. Plataforma PersonApp

*PersonApp* es un sistema multiplataforma desarrollado internamente por un grupo de investigación del Tecnológico Nacional de México campus Culiacán (Bátiz, V. 2021; Bátiz, V. et al., 2022). Contiene dos temas importantes en la evaluación de modelos de reconocimiento automático de personalidad aparente. En la Figura 4-1 se observa la interfaz de la ventana principal de esta plataforma.

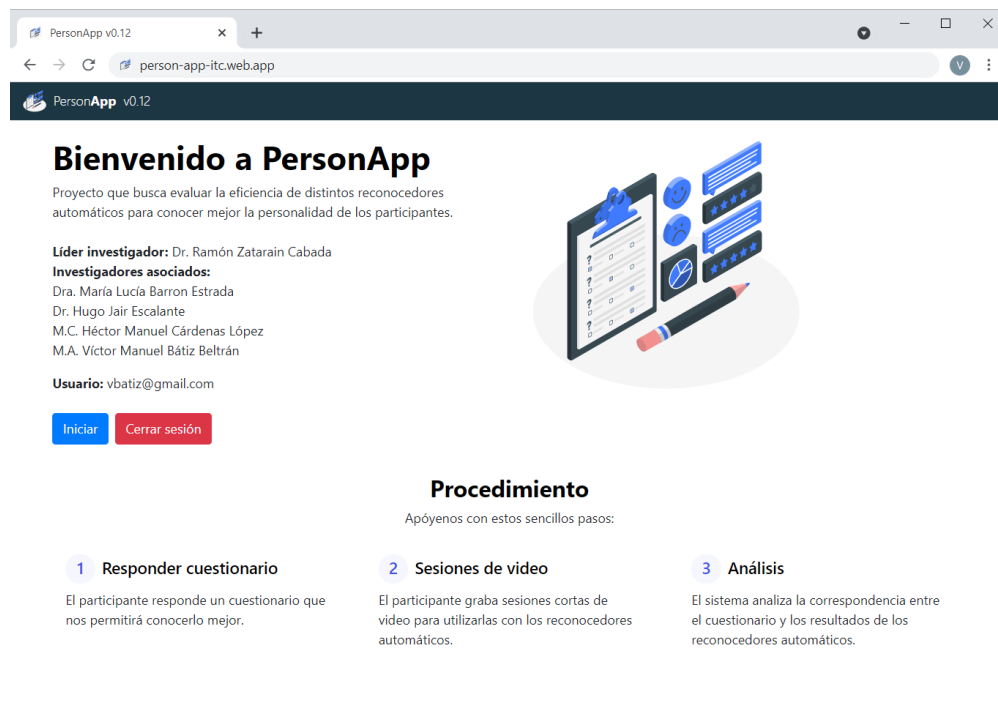


Figura 4-1. Interfaz de bienvenida de la plataforma PersonApp (Bátiz et al., 2022)

El principal objetivo de *PersonApp* es recolectar información y asignar valores a los rasgos de personalidad de cada participante mediante la realización de una prueba estandarizada llamada *IPIP* la cual consiste en una evaluación de 50 ítems de los marcadores mencionados por Goldberg en el modelo *OCEAN*. En la Figura 4-2, se muestra una parte de la sección de las preguntas de la prueba estandarizada *IPIP*, donde los usuarios responden una serie de 50 preguntas de opción múltiple, eligiendo la que mejor se adapte a su personalidad. Después los resultados obtenidos de la prueba se toman como un valor real para cada uno de los cinco rasgos de personalidad. Finalmente se compara la exactitud de los valores generados automáticamente por los modelos de clasificación de personalidad.

El segundo objetivo de la plataforma es poder recolectar videos (con audio incluido) de los usuarios que realizan los experimentos incluidos en *PersonApp* y utilizar estos datos como archivos de entrada para los clasificadores automáticos que buscan reconocer la personalidad.

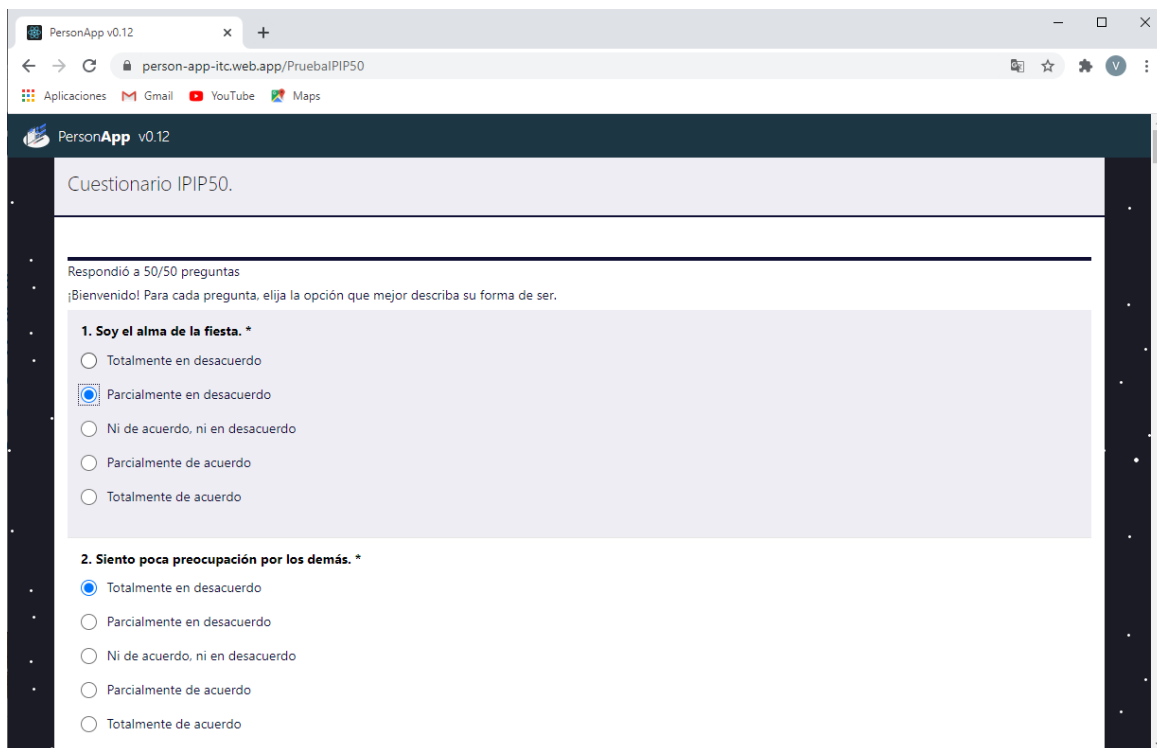
The image is a screenshot of a web browser displaying the 'PersonApp v0.12' interface. The browser's address bar shows 'person-app-ipc.web.app/PruebaIPIP50'. The page has a dark blue header with the 'PersonApp v0.12' logo. Below the header, the title 'Cuestionario IPIP50.' is displayed. A progress bar indicates 'Respondió a 50/50 preguntas'. A welcome message reads: '¡Bienvenido! Para cada pregunta, elija la opción que mejor describa su forma de ser.' The questionnaire consists of two items. Item 1 is '1. Soy el alma de la fiesta. \*' with five radio button options: 'Totalmente en desacuerdo', 'Parcialmente en desacuerdo' (which is selected), 'Ni de acuerdo, ni en desacuerdo', 'Parcialmente de acuerdo', and 'Totalmente de acuerdo'. Item 2 is '2. Siento poca preocupación por los demás. \*' with five radio button options: 'Totalmente en desacuerdo' (which is selected), 'Parcialmente en desacuerdo', 'Ni de acuerdo, ni en desacuerdo', 'Parcialmente de acuerdo', and 'Totalmente de acuerdo'.

Figura 4-2. Cuestionario IPIP de 50 ítems, imagen obtenida de (Bátiz et al., 2022)

Una vez que el participante ha respondido las 50 preguntas, la plataforma *PersonApp*, redirecciona al usuario a la sección de grabación de videos. En esta sección consta de una serie de preguntas o “experimentos”. En cada experimento los participantes realizan una grabación de video de ellos mismos respondiendo una serie de preguntas de tres temas diversos, expresando su punto de vista o ideas utilizando sin alguna restricción de vocabulario, es decir, los usuarios son libres de responder las preguntas utilizando su léxico habitual, todo esto con la finalidad de que las respuestas sean reales y sinceras para poder tener datos reales del tipo de personalidad del individuo. Las preguntas que se realizan en los 3 experimentos de esta sección se describen en la Tabla 4-1.



Tabla 4-1. Preguntas o instrucciones para las grabaciones de los participantes.

Experimento	Pregunta o instrucción realizada a participante
Experimento 1-1	Una vez que presiones iniciar, habla sobre un tema que te guste. Puedes hablar sobre lo que sea. La recomendación es imaginar que se habla con amigos en una reunión de <i>Zoom</i> o <i>Teams</i> acerca del tema.
Experimento 1-2	En esta actividad pretenderás estar en una entrevista virtual en línea, trata de responder las siguientes preguntas ¿Cuáles son tus mayores fortalezas y debilidades?, ¿Por qué consideras que deberíamos contratarte?
Experimento 1-3	Por último, pretenderás ser una persona creando un video para exponer un tema en una plataforma en línea ( <i>Youtube</i> , <i>Facebook</i> , <i>Twitch</i> , etc.), tú eliges el tema y la manera de presentarlo.

Todos los datos y videos recopilados de la plataforma son almacenados de manera segura en la plataforma *Firebase*, que es una herramienta de *Google* que ofrece diversos servicios gratuitos, así como también de paga, y son de gran ayuda para el desarrollo de aplicaciones. Los servicios que utiliza *PersonApp* es la autenticación, hospedaje, base de datos no relacional de *Firestore* y el almacenamiento de archivos (Firebase, 2022).

Toda la información y grabaciones de los videos de los usuarios son completamente anónimos y seguros, además solo son usados para fines de investigación, por lo que los correos electrónicos y nombres fueron omitidos del corpus. En la Figura 4-3 se muestran algunos de los datos que se encuentran almacenados de los usuarios en la nube de *Firebase*.

	fecha	sociabilidad	apertura	anabilidad	responsabilidad	tipotest	uid	tiempoSegundos	neuroticismo	email
0	2021-04-29 00:31:07.671000+00:00	0.7	0.6	0.8	0.4	IPIP50	8QJw6ATMQIWupPEHhYQwRd3C2	293	0.6	edu.mx
1	2021-09-20 18:34:57.599000+00:00	0.7	0.6	0.4	1.0	IPIP50	sBMzEL6gnTMHxL9U29XCacsKog1	356	0.6	ig@gmail.com
2	2021-10-29 00:08:47.412000+00:00	0.5	0.7	0.8	0.7	IPIP50	T8yWJweTSS4DnU3oJSSaAVc72	206	0.4	edu.mx
3	2021-09-20 14:15:16.534000+00:00	0.6	0.6	0.6	0.6	IPIP50	InMj2vmeRzS01eYsZf63wey8z1	123	0.6	edu.mx
4	2021-04-23 01:37:08.434000+00:00	0.6	0.7	0.9	0.8	IPIP50	n84XTQwKheUqUcjo0lusWEAtg1	850	0.7	ig@hotmail.com

Figura 4-3. Información de usuarios recolectada por plataforma *PersonApp* y almacenada en nube de *Firebase*.

Debido al propósito que tiene esta investigación, el cual es crear un corpus en español confiable para utilizarlo en tareas de reconocimiento automático de personalidad, se optó por tomar la información contenida en la plataforma *PersonApp* para la creación de un nuevo conjunto de datos, utilizando sólo la información que está basada en el modelo *Big-Five* y que puede ser utilidad para tareas de reconocimiento de personalidad mediante texto. Los datos principales que contiene *PersonApp* y que se extrajeron para la creación del corpus *PersonText* fueron:

- El identificador de usuario.
- Los videos de las opiniones de los usuarios.
- Los resultados de las evaluaciones de los cinco rasgos de personalidad de las pruebas estandarizadas IPIP.

#### 4.1.2. Transformación de video a texto

*PersonApp* contiene los archivos de videos grabados por los participantes, así como los valores asociados a los rasgos de personalidad de la prueba IPIP basados en el modelo de los cinco grandes. En estos videos los participantes expresan libremente su opinión sobre diversos temas o preguntas elaboradas con el propósito de hacer denotar su personalidad. La plataforma almacena estos archivos de video en la nube de Firebase. Estos videos se almacenan por carpetas correspondientes a cada experimento. Cada carpeta de los experimentos separa y almacena a su vez los videos de cada usuario por carpetas que llevan por nombre el identificador único asignado de cada participante. En la Figura 4-4 se muestra un ejemplo del almacenamiento de los videos en la nube.

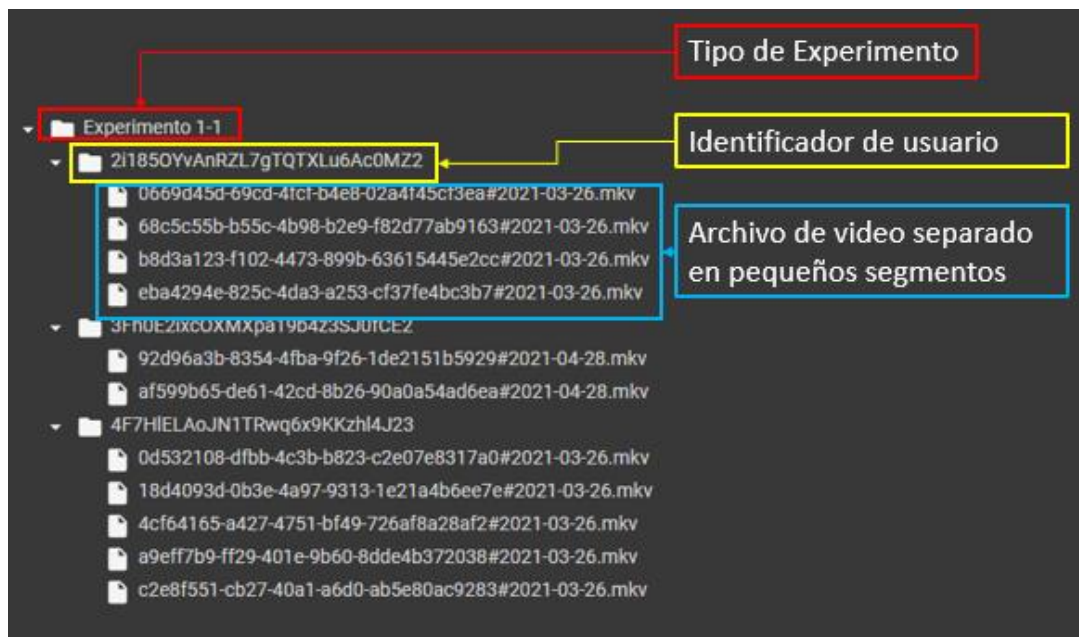


Figura 4-4. Almacenamiento de videos en la nube de Firebase.

Cada carpeta contiene la grabación de video de cada experimento y el video se encuentra dividido en versiones cortas de 15 segundos por lo que fue necesario procesar alrededor de 700 videos en total.

Para la creación del corpus se decidió extraer los textos de las video grabaciones de los usuarios. Para realizar la extracción de textos se siguieron los siguientes pasos:

1. Descarga de videos de nube de Firebase.
2. Conversión de videos a audio.
3. Conversión de audio a texto.

Como la cantidad de videos es demasiada se optó por hacer la descarga de manera automática bajo el lenguaje de programación Python. En la Figura 4-5 se observa los pasos seguidos para la descarga de videos y extracción de texto de ellos. Para la extracción de textos y creación de la primera versión del corpus *PersonText*, se utilizó la biblioteca Speech Recognition (Zhang, A., 2015) bajo el lenguaje de programación de Python. Esta primera versión solo contiene los textos extraídos de las grabaciones de los participantes y su identificador de usuario. En la Figura 4-6 se puede observar un primer prototipo del corpus *PersonText* utilizando la biblioteca de *Python Speech Recognition*.



Figura 4-5. Metodología para creación de corpus de texto.

Debido a que varios usuarios cometieron errores en la grabación de sus videos o que el audio no se escuchaba correctamente, se tuvieron que descartar varios videos de participantes, así como su información personal.

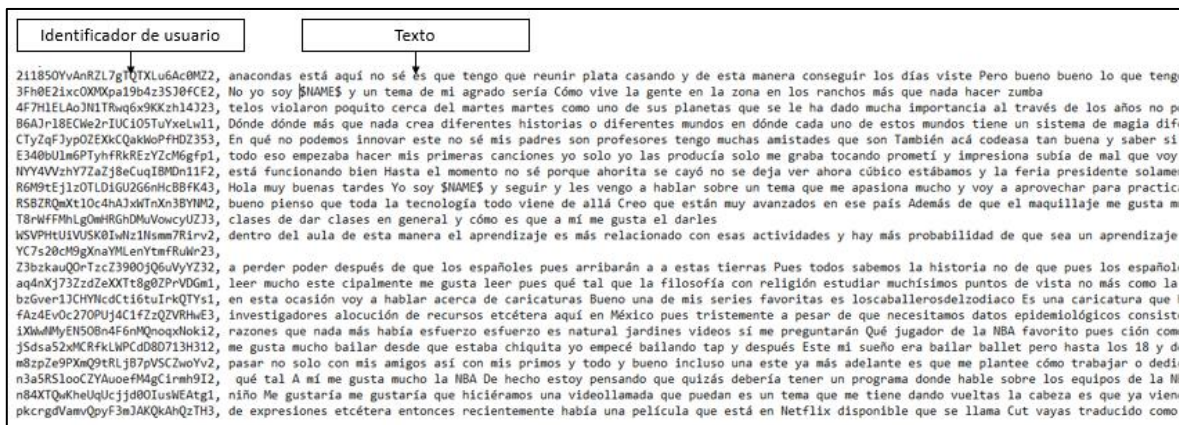


Figura 4-6. Primer prototipo del corpus PersonText.

Una vez obtenido los textos automáticamente, se compararon con discursos de los videos reales y se realizaron modificaciones manualmente a los textos del primer prototipo del corpus. Entre estas modificaciones se puede mencionar la omisión de cualquier nombre que se encontró en los textos y se sustituyó por el acrónimo \$NAME\$, además se corrigieron los errores ocurridos en la extracción automática de texto.

#### 4.1.3. Extracción de resultados de pruebas IPIP

Una vez obtenido los textos, se procedió a extraer los valores de los cinco rasgos de personalidad correspondiente a cada usuario del modelo *OCEAN*.

Estos valores fueron obtenidos mediante la realización de la prueba estandarizada IPIP, la cual consiste en una evaluación de 50 preguntas con respuestas de opción múltiple, donde el voluntario elige la opción que más se asemeje a su comportamiento o personalidad real. Estos resultados se encuentran en una escala de valores de 0 a 1 donde 1, es completa presencia del atributo de personalidad y “0” nula presencia.

Para obtener los resultados de las evaluaciones de los cinco rasgos de personalidad de cada usuario que se encuentran almacenados en el repositorio de la nube de *Firebase*, se utilizó un cuaderno (*Notebook*) de Google Colaboratory (Google Colaboratory, 2022), bajo el lenguaje de programación Python. Se tomó de nueva cuenta el identificador de usuario (*user ID*) como medio para poder identificar los valores de cada participante y exportarlos al corpus de texto. En la Figura 4-7 se observan los cinco rasgos de personalidad con las respectivas evaluaciones de varios usuarios.

	amabilidad	sociabilidad	responsabilidad	apertura	neuroticismo	uid
0	0.8	0.7	0.4	0.6	0.6	tkQJw6ATMQIWupPEHhYCwRoI3C2
1	0.4	0.7	1.0	0.6	0.6	sBMzEL6gnTMHXL9U29XCacsxKoq1
2	0.8	0.5	0.7	0.7	0.4	T8yiWJwcT5S4DiriJ3oJSSsAVc72
3	0.6	0.6	0.6	0.6	0.6	lnMj2vmeRxSI01eYsjZfe3wey8z1
4	0.9	0.6	0.8	0.7	0.7	n84XTQwKheUqUcjd0OlusWEAtg1
...	...	...	...	...	...	...
128	1.0	0.6	0.9	0.6	0.2	DMYQUxOB5cSGYvcgMCIhZne4G4u2
129	0.9	0.6	0.4	1.0	0.7	CTyZqFJypOZEXkCQakWoPfhDZ353
130	0.8	0.7	0.6	0.7	0.6	4UTaotXI9GhBoGTIsY10ELmyMsU2
131	0.6	0.7	0.9	0.8	0.3	sFYdnMnKG9Sy7cL0sju55MQSN543

Figura 4-7. Evaluaciones de los cinco rasgos de personalidad de varios usuarios.

#### 4.1.4. Relación de textos y etiquetas de valores de rasgos de personalidad.

Una vez obtenido tanto los textos como los resultados de las evaluaciones de la prueba IPIP, el siguiente paso fue agrupar estos datos correspondientes a cada usuario y exportarlos a un solo archivo, obteniendo así un segundo prototipo del corpus *PersonText*. En la Figura 4-8 se muestra el segundo prototipo del corpus *PersonText*, en el cual ya se encuentran las etiquetas del usuario, los textos y los cinco rasgos de personalidad: apertura, responsabilidad, extraversión, amabilidad, neuroticismo con su respectivo valor. Para realizar este proceso se utilizó de igual manera un Notebook en Google *Colaboratoy*.

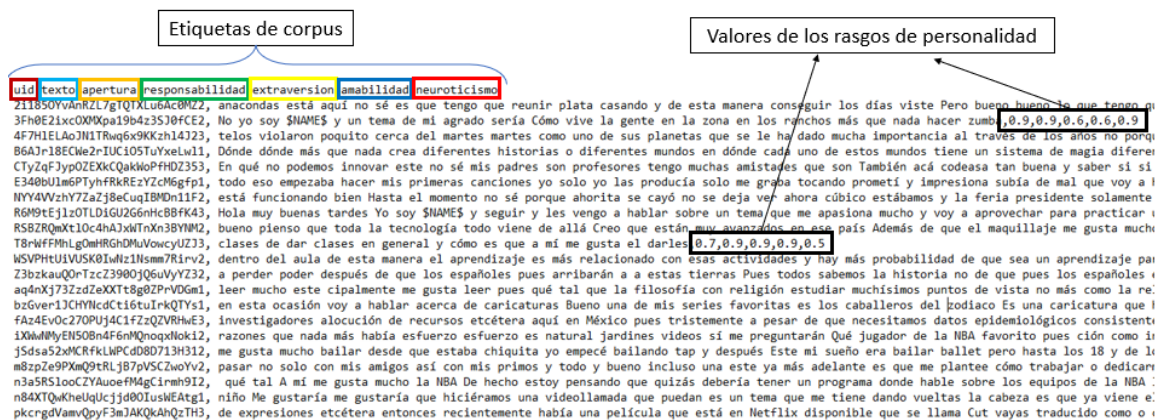


Figura 4-8. Imagen del segundo prototipo del corpus *PersonText*.

#### 4.1.5. Creación de etiquetas de presencia / ausencia de rasgo de personalidad

Al finalizar la construcción del primer prototipo de *PersonText*, el cual ya incluía los textos y sus respectivos valores de personalidad de cada usuario surgió la siguiente pregunta: ¿Cómo saber si nuestro corpus está correctamente estructurado?

Antes de definir la estructura del corpus y empezar a trabajar con las evaluaciones del corpus mediante los modelos *DL*, primeramente, se tuvo que definir el tipo de problema que se quería resolver. Para esto se tenían dos enfoques:

- 1) Problema de Regresión: Calcular los valores numéricos con los rasgos de personalidad de cada usuario a partir de texto.
- 2) Problema de Clasificación: Clasificar si hay presencia o no, de cada uno de los atributos de personalidad en los individuos mediante texto.

Se decidió por tomar la segunda vertiente, la cual resumió la investigación a un problema de clasificación, es decir, el modelo reconoce si la persona tiene o no el atributo de personalidad a partir de introducir un texto mediante una clasificación binaria. En el modelo un “Sí”, indica la presencia del rasgo de personalidad y un “No”, hace referencia a que el rasgo de personalidad es nulo en el individuo. Una de las razones para no elegir el enfoque de regresión, es que calcular el valor numérico del atributo de personalidad no indica o no tiene una interpretación clara para los fines de reconocimiento de personalidad.

Una vez definido que se trabajaría con un problema de clasificación, se procedió a incluir al corpus *PersonText* las etiquetas de “presencia de atributo de personalidad”, es decir para cada uno de los cinco rasgos de personalidad se le agregó una etiqueta. Por ejemplo, para el atributo de apertura la nueva etiqueta quedó como “presencia de apertura”. Para definir dicha condición, se tomó como base los puntajes de los resultados de las pruebas IPIP, donde si el resultado de la evaluación era superior a 0,5 se le agregó manualmente un “Sí”, de lo contrario se colocó un “No”.

Después, se hizo una codificación de las etiquetas de presencia / ausencia del rasgo de personalidad. Una vez determinado si la persona posee o no el rasgo de la personalidad, se convirtieron las etiquetas de personalidad de “No” y “Sí” en números para que el modelo las pueda entender.



Para representar un “Sí” se asignó el número “1” y para el caso de “No” se asignó un “0”. Como el conjunto de datos es categórico, se necesita convertir las etiquetas de personalidad de No y Sí, en un tipo flotante (*float*) para que el modelo pueda realizar los cálculos. Para lograr esta tarea, se implementó el método *to\_categorical()* de *Keras*. El Fragmento de código para asignar valores flotantes a nuestras etiquetas de No y Si se muestra en la Figura 4-9.

```
etiquetas = np.array(datos_limpios['presenta_apertura'])
y = []
for i in range(len(etiquetas)):
    if etiquetas[i] == 'No':
        y.append(0)
    if etiquetas[i] == 'Sí':
        y.append(1)
y = np.array(y)
etiquetas = tf.keras.utils.to_categorical(y, 2, dtype="float32")
del y
```

Figura 4-9. Asignación de valores numéricos a etiquetas.

#### 4.1.6. Estructura final del corpus *PersonText*

Como paso final, en la definición de la estructura se decidió agregar algunos datos de los participantes como son: el sexo, su fecha de nacimiento, la edad y el tipo de experimento.

El corpus *Persontext* cuenta con un total de 213 textos en español. Los datos del corpus se encuentran apilados y divididos en secciones que van desde el usuario, el texto de las respuestas de cada experimento, los valores del resultado de las pruebas IPIP, así como la etiqueta de presencia (Sí/No) de cada rasgo de personalidad y la información adicional. En la Figura 4-10 se observa una parte del corpus *PersonText* en una versión final.

uid	texto	fecha_prueba	apertura	presenta_ap	responsabili	presenta_re	sociabilidad	presenta_so	amabilidad	presenta_an	neuroticismo	presenta_ne	fechaNacimi	sexo	edad_calcul	Experim	
n84XTQwKh	Un tema qui	23/04/2021	0.7	Si		0.8	Si	0.6	No	0.9	Si	0.7	Si	03/09/1991	Femenino	29	Exp 1-1
iXWwNMMyE	Si a mi me p	04/03/2021	0.8	Si		0.8	Si	0.3	No	0.8	Si	0.9	Si	23/07/1994	Masculino	26	Exp 1-1
n3a5RSlooC2	Qué tal a mi	04/03/2021	0.5	No		0.5	No	0.6	No	0.6	No	0.7	Si	24/10/1995	Masculino	25	Exp 1-1
CTyZqFjypO	Me gusta mi	23/04/2021	1	Si		0.4	No	0.6	No	0.9	Si	0.7	Si	25/10/1980	Femenino	40	Exp 1-1
B6AjrI8ECW	Les hablaré	04/03/2021	0.7	Si		0.8	Si	0.7	Si	0.5	No	0.9	Si	02/12/1995	Masculino	25	Exp 1-1
pkcrgdVamv	Hola pues le	23/04/2021	0.8	Si		0.6	No	0.5	No	0.8	Si	0.8	Si	15/11/1986	Femenino	34	Exp 1-1
Z3bzkaUQOr	Hola Pues a	26/03/2021	0.8	Si		0.6	No	0.8	Si	0.6	No	1	Si	22/09/1995	Masculino	25	Exp 1-1
WSVPHtUivL	Hola Mi non	23/04/2021	0.7	Si		0.6	No	0.5	No	0.9	Si	0.5	No	05/05/1987	Femenino	33	Exp 1-1
aq4nXj73Zzd	Hola Mi non	29/04/2021	0.8	Si		0.6	No	0.4	No	0.8	Si	0.3	No	07/08/1995	Masculino	25	Exp 1-1
bzGverLJCHY	En esta ocas	10/02/2021	0.8	Si		0.8	Si	0.6	No	0.8	Si	0.6	No	15/03/1984	Masculino	36	Exp 1-1
E340bUIm6P	El tema del	26/03/2021	0.6	No		0.7	Si	0.6	No	0.7	Si	0.3	No	01/03/1996	Masculino	25	Exp 1-1
T8rWfFMhLg	El otro día e	26/03/2021	0.7	Si		0.9	Si	0.9	Si	0.9	Si	0.5	No	12/04/1994	Masculino	26	Exp 1-1
3Fh0E2ixcOX	Bueno yo so	28/04/2021	0.9	Si		0.9	Si	0.6	No	0.6	No	0.9	Si	02/09/1996	Masculino	24	Exp 1-1
2i185OYvAnf	Bueno y con	26/03/2021	0.6	No		0.4	No	0.6	No	0.9	Si	0.8	Si	30/01/1997	Masculino	24	Exp 1-1
RSBZRQmXtl	Bueno uno c	26/03/2021	0.6	No		0.7	Si	0.4	No	0.7	Si	0.3	No	09/06/1996	Femenino	24	Exp 1-1
fA4zEvOc27C	Bueno un ur	26/04/2021	0.7	Si		0.6	No	0.8	Si	0.9	Si	0.2	No	24/12/1986	Femenino	34	Exp 1-1
m8zpZe9PXn	Bueno pues	26/03/2021	0.9	Si		0.7	Si	0.2	No	0.7	Si	0.6	No	10/07/1997	Masculino	23	Exp 1-1
4F7HIELAoJN	A mi me gus	26/03/2021	0.8	Si		0.9	Si	0.6	No	0.7	Si	0.8	Si	06/11/1996	Masculino	24	Exp 1-1
j5ds52xMcF	A mi me gus	26/04/2021	0.8	Si		0.8	Si	0.8	Si	1	Si	0.3	No	06/09/1995	Femenino	25	Exp 1-1
n84XTQwKh	Bueno mis f	23/04/2021	0.7	Si		0.8	Si	0.6	No	0.9	Si	0.7	Si	03/09/1991	Femenino	29	Exp 1-2
iXWwNMMyE	Pues porqu	04/03/2021	0.8	Si		0.8	Si	0.3	No	0.8	Si	0.9	Si	23/07/1994	Masculino	26	Exp 1-2
n3a5RSlooC2	Una de mis f	04/03/2021	0.5	No		0.5	No	0.6	No	0.6	No	0.7	Si	24/10/1995	Masculino	25	Exp 1-2
CTyZqFjypO	Yo creo que	23/04/2021	1	Si		0.4	No	0.6	No	0.9	Si	0.7	Si	25/10/1980	Femenino	40	Exp 1-2
B6AjrI8ECW	Bueno pues	04/03/2021	0.7	Si		0.8	Si	0.7	Si	0.5	No	0.9	Si	02/12/1995	Masculino	25	Exp 1-2
pkcrgdVamv	Bueno Mi pr	23/04/2021	0.8	Si		0.6	No	0.5	No	0.8	Si	0.8	Si	15/11/1986	Femenino	34	Exp 1-2
Z3bzkaUQOr	Hola buenas	26/03/2021	0.8	Si		0.6	No	0.8	Si	0.6	No	1	Si	22/09/1995	Masculino	25	Exp 1-2

Figura 4-10. Vista parcial del corpus *PersonText* versión Final.

#### 4.2. Metodología para crear y entrenar Redes Neuronales DL para reconocimiento de personalidad mediante texto

Una vez creado el corpus *PersonText* en su última versión, se procedió a resolver las siguientes preguntas de investigación: ¿Se podrá usar *PersonText* para entrenamiento de modelos *DL*? ¿Qué modelos *DL* y configuraciones serán las más apropiadas para la tarea de reconocimiento de personalidad?, ¿Cómo evaluar nuestro corpus?.

En esta sección de la investigación se describe la metodología desarrollada para crear y entrenar modelos *DL* como lo son las redes neuronales que tienen como objetivo llevar a cabo la evaluación del corpus *PersonText* en la tarea del reconocimiento automático de personalidad mediante texto. Uno de los impedimentos de los modelos *ML* y *DL* es la incapacidad de procesar el texto como tal, por lo que se debe convertir el texto en un lenguaje que tanto la computadora como el mismo modelo puedan reconocer, por lo que se utilizaron técnicas de NLP para poder así realizar las pruebas necesarias con texto.

También en este capítulo se describe el preprocesamiento realizado al corpus *PersonText* para su limpieza y la extracción de características más importantes, los cuales serán los datos de entrada para los modelos de *Deep Learning* utilizados para realizar la clasificación de forma binaria (Sí/No).

En la Figura 4-11 se observa la metodología desarrollada para crear redes neuronales utilizando el corpus de texto.



Como primer paso, al corpus de texto, se le realiza una serie de preprocesamiento o limpieza para eliminar el ruido o datos incorrectos; además se divide el corpus en una proporción de 75% de datos que serán usados para el entrenamiento y el restante para hacer respectivas validaciones. Una vez realizado el preprocesamiento, se extraen las características principales de los textos y se transforman a valores numéricos (vectores). Dichos vectores serán los datos de entrada a nuestros modelos *DL* para hacer los respectivos entrenamientos. Durante la etapa de entrenamiento se modifican las configuraciones de las redes neuronales hasta obtener el mejor resultado y la mejor red neuronal (ajuste de hiperparámetros). Una vez encontrado el mejor modelo, se procede a hacer el reconocimiento de los rasgos de personalidad con los datos de validación. El modelo tendrá como datos de salida una clasificación binaria, indicando si el individuo posee o no el rasgo de personalidad a partir del aprendizaje previamente realizado con los datos de entrenamiento.

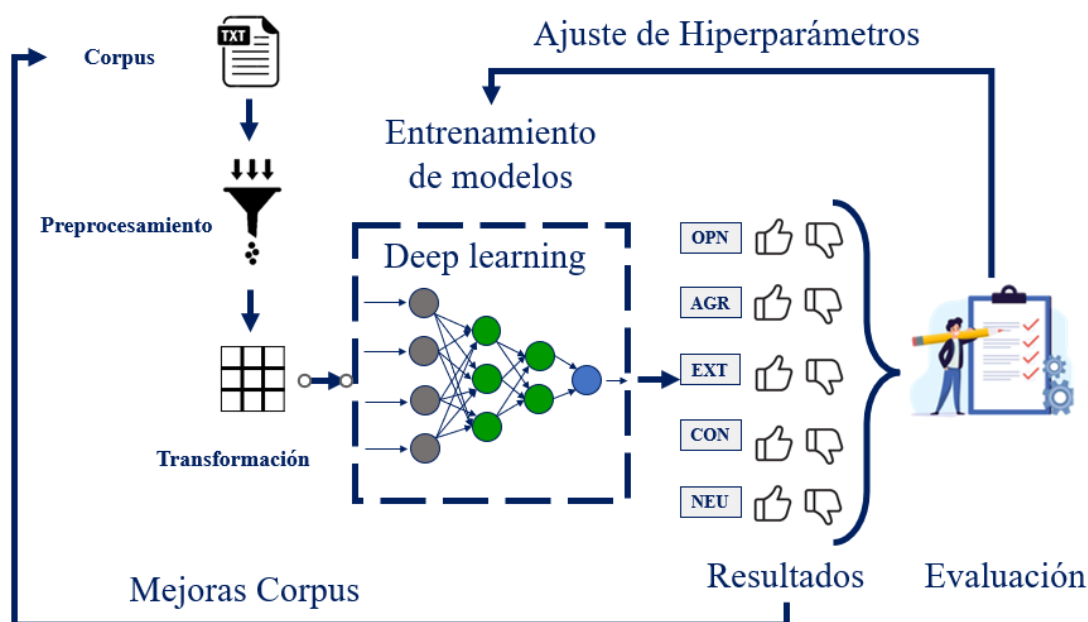


Figura 4-11. Metodología para crear redes neuronales utilizando el corpus de texto.

Las secciones siguientes describen el proceso para preprocesar el corpus de texto para su limpieza, la extracción de características importantes, los datos de entrenamiento, y por último las configuraciones y pruebas de los modelos *DL*.

#### 4.2.1. Datos de entrenamiento

Los datos de entrenamiento son los datos que utilizan las redes neuronales o algoritmos *DL* para reconocer, extraer y aprender las características principales de este conjunto. Esto le brinda la capacidad, al modelo, de calcular o clasificar, si un nuevo conjunto de datos posee o no las características aprendidas.

Por esto es de vital importancia tener cuidado con los datos que se le proporcionarán a los algoritmos por lo que los datos de entrenamiento deben poseer las siguientes características:

- Libres de ruido: Se considera que hay ruido en los datos cuando hay existencia de datos con errores en su medición o en su registro, es decir, los datos que fueron mal etiquetados (datos que no pertenecen a la clase asignada), mal escritos o capturados o en su defecto números mal calculados, etc.
- Patrones atípicos: se refiere a los datos que se etiquetan correctamente pero que son muy diferentes con respecto a los otros datos de la misma clase.
- Muchos datos: los datos de entrenamiento deben ser muchos, ya que entre más datos contenga un conjunto de datos de entrenamiento, más características aprenderá de los mismos.

Actualmente el corpus *PersonText* tiene 213 textos con sus respectivas etiquetas e inventario de resultados de pruebas IPIP. El contenido (textos, etiquetas e información) y estructura del corpus fue verificado por el grupo de investigadores del Tecnológico Nacional de México campus Culiacán y se tomó la decisión de utilizarlo para el entrenamiento del nuevo modelo.

Una vez que se definió el conjunto de datos de entrenamiento, el siguiente paso que se realizó fue hacer un preprocesamiento a los textos de los corpus. El preprocesamiento de los datos se entiende como el proceso de limpieza de estos para poder manipularlos de manera fácil en los siguientes pasos de la investigación.

#### 4.2.2. Preprocesamiento del corpus

Antes de comenzar con la creación de redes neuronales, su configuración y el entrenamiento de los modelos, se realizó una limpieza o preprocesamiento a cada uno de los textos del corpus *PersonText*. Esto se realizó con la finalidad de evitar tener ruidos en conjunto de datos de entrenamiento y extraer las características más importantes de él, para aumentar la

fiabilidad en el entrenamiento y evaluación de los modelos de clasificación de texto. Cuando se habla de tener un conjunto de datos de entrenamiento sin ruido o limpio, es decir se trata de eliminar del corpus de texto los signos de puntuación, las cifras numéricas, correos o enlaces URL, etc. de los textos, ya que se consideran irrelevantes para el objetivo de esta investigación.

Dentro del preprocesamiento o limpieza hecho al corpus se pueden mencionar las siguientes acciones:

- a) Convertir palabras a minúscula. En esta parte se convirtieron todas las palabras a minúsculas para que todas las palabras tengan un mismo formato.
- b) Remover URL, los correos y saltos de línea. Puede haber situaciones donde se pueden incluir en los textos algún enlace o *URL* de una o más páginas de internet, así como direcciones de correo electrónico y saltos de línea, por lo que se hizo una eliminación de todos estos.
- c) Eliminar nuevas líneas y pestañas y saltos de línea. En caso de la extracción automática de texto, puede que los textos contengan los caracteres para indicar nuevas líneas o pestañas como (\n, \t y \s), los cuales son removidos.
- d) Remover comillas simples. Se remueven comillas (') de los textos.
- e) Eliminar comas. Se eliminan comas (,) que sirven para separar textos o continuar con la misma idea y se sustituyen por espacios en blanco.
- f) Remover números. Puede ser que algún usuario escriba algún número en forma de números reales por lo que se puede sustituir por la letra o eliminarlos como se hizo en esta investigación.
- g) Remover caracteres no alfanuméricos. Cualquier signo que no sea número o letra es removida de los textos como lo son los siguientes caracteres: #, \$, %, ¡, “, ?, etc.
- h) Eliminar guiones entre palabras. Se eliminan los guiones (-) que se pueden encontrar entre palabras como “ciencia-ficción”, resultando como “ciencia ficción”.
- i) Eliminar los guiones dobles y triples. Puede ser el caso donde se encuentren dos o más guiones junto con alguna palabra por lo que se sustituyen con un espacio en blanco.

- j) Eliminar espacios en blanco al principio o final de un texto, así como los espacios dobles. Un espacio en blanco también puede ser considerado como un token. Por lo que se eliminan.
- k) Quitar *Stopwords* del texto. También conocidas como palabras vacías, ya que son consideradas palabras que no ayudan en absoluto en tareas de clasificación de texto, es decir no aportan nada y si son removidas no afecta en lo más mínimo al sentido de la oración. En este caso la biblioteca *nlk* proporciona ya una lista predeterminada de *stop words* para el idioma español, las cuales se pueden descargar y comenzar a usar. La biblioteca de *nlk* contiene muy pocas palabras vacías en español por los que se creó una nueva lista de *stopwords*.
- l) Realizar lematización. La lematización es una forma más avanzada de derivación (*stemming*) y consiste en convertir todas las palabras a su forma infinitiva. La lematización convierte diferentes formas de palabras o inflexiones a una forma estándar. Por ejemplo, convierte la palabra *jugando* en *jugar* o la palabra *escribió* en *escribir* y *dormía* en *dormir*, etc. Un lematizador trabaja con más reglas de lenguaje e información contextual a comparación de la derivación (*stemming*). También hace uso de un diccionario para buscar palabras equivalentes, razón por la cual requiere más potencia de procesamiento y tiempo para generar resultados.
- m) Remover signos de puntuación. Se remueven todos los signos de puntuación que contengan las palabras como los acentos (á, é, í, ó, ú), diéresis (¨), virgulilla o tilde (~), etc.

#### **4.2.3. Extracción de características / Tokenización**

Una vez realizada la limpieza de nuestro corpus *PersonText*, se procedió a realizar la extracción de características más importantes de nuestros textos. Para esto se utilizó la técnica *Bag of Words (BoW)* mencionada en la [Sección 2.3.3.5], también conocida como selección de atributos o variables. En esta técnica, los textos y oraciones son considerados como una bolsa de sus propias palabras y solo se considera las palabras duplicadas, ignorando la gramática y el orden de las palabras. Normalmente *BoW* se utiliza en métodos de clasificación de textos, para clasificar o categorizar textos y calcular pesos para cada palabra, los cuales son el número de repeticiones de cada una de las palabras.

Para realizar la extracción de características con la técnica *BoW*, primero es necesario transformar o convertir los textos a datos numéricos para que nuestro Modelo *DL* pueda realizar sus cálculos. Para esto se le realiza una tokenización al corpus normalizado. Cuando se habla de tokenizar, se refiere a la acción de separar todas las palabras del texto en entidades llamadas tokens.

En la Figura 4-12 se muestran dos ejemplos de tokenización de oraciones.

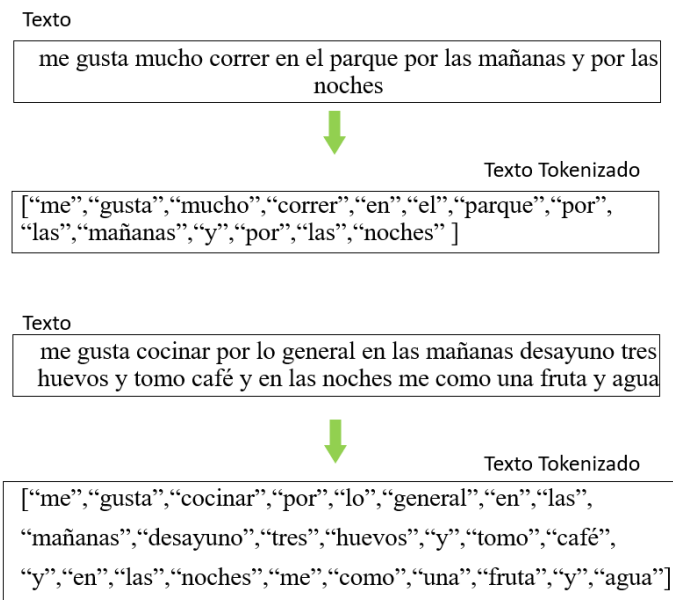


Figura 4-12. Proceso de Tokenización de texto.

En la técnica *BoW* se basa en calcular la cantidad de veces que aparece cada token en todo el corpus de texto para encontrar los tokens más recurrentes o similares y generar una bolsa de palabras o diccionario. Para realizar esto se utiliza la clase *Tokenizer* de *Keras*. Esta clase permite transformar el texto en función de repetición de las palabras, donde la palabra más común tendrá un valor tokenizado de 1, la siguiente palabra más común el valor 2, y así sucesivamente con el resto de los *tokens*. Como resultado se obtiene un diccionario que contiene una clave y un valor. La clave corresponde a la palabra única y el valor concierne al token asignado calculado con respecto a la frecuencia de la palabra.

Al realizar esta acción es posible detectar *tokens* (palabras) que no son tan frecuentes o no agregan algún valor o sentido y pueden ser omitidos con la finalidad de mejorar los datos de entrenamiento y optimizar la precisión de nuestro modelo de aprendizaje máquina. En la

Figura 4-13 se observa una representación del diccionario obtenido con la técnica *BoW*, a partir de los tokens correspondiente al ejemplo de la ilustración pasada, el cual se encuentra ordenado de forma descendente con respecto a la frecuencia de aparición de los tokens en las oraciones.

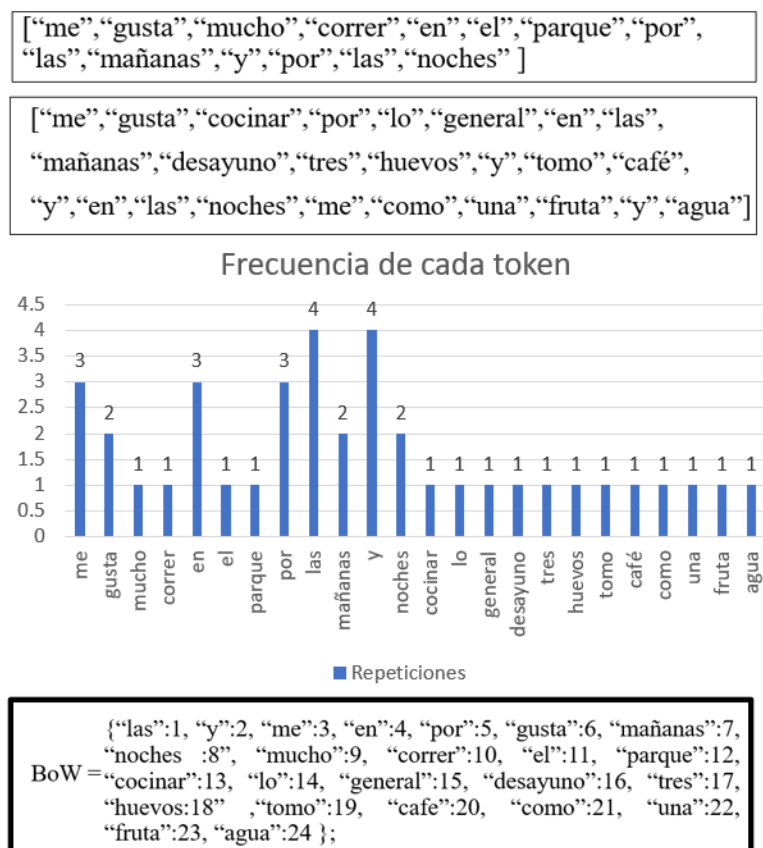


Figura 4-13. Orden de tokens por la frecuencia de palabras.

En la Figura 4-14 se observa el fragmento del código para tokenizar el corpus *PersonText*. Con esta misma clase se obtiene un diccionario de tokens ordenados descendientemente por el número de frecuencias o apariciones de cada token en el corpus.

```

# Vectorizar el corpus
tokenizer = Tokenizer()
"""
Recibe el corpus en forma de una lista de textos (strings) y los convierte en un
diccionario de tokens asignando un identificador único a cada token
"""
tokenizer.fit_on_texts(new_data)
word_index = tokenizer.word_index
total_unique_words = len(tokenizer.word_index) + 1

sequences = tokenizer.texts_to_sequences(new_data)
max_seq_length = max([len(x) for x in sequences])
textos = pad_sequences(sequences, maxlen=max_seq_length)

```

Figura 4-14. Fragmento del código para tokenizar el corpus *PersonText*.

En la primera línea el método *fit\_on\_texts()*, se utiliza para crear un identificador único a cada token del diccionario. El método recibe como parámetro el corpus normalizado (limpio), el cual contiene un conjunto de listas de *strings*, correspondiente a cada uno de los textos de cada participante.

Una vez obtenido el diccionario, se utiliza el método *texts\_to\_sequences()*, que tiene la función de representar cada token del diccionario en un vector con un identificador único, es decir, sustituye los identificadores únicos por los tokens en cada vector. Cada vector representa a un texto de un usuario.

Debido a que los textos del corpus poseen diferentes cantidades de palabras, ocasiona que la cantidad de tokens incluidos en los vectores creados, sean de diferente tamaño. Atendiendo a esta situación se utiliza el método *pad\_sequences()* con el que se realiza normalización de cada vector a un tamaño único. Para esto se identifica el vector de mayor tamaño, y el resto de los vectores es rellenado con ceros tantas veces sea necesaria hasta llegar a la longitud del vector más largo (texto con mayor número de palabras o tokens encontrados en el corpus). Una vez codificado el corpus *PersonText* utilizando la técnica *Bag of Words*, el siguiente paso es crear y configurar los modelos *DL*.

### 4.3. Creación de las redes neuronales de aprendizaje profundo

El modelo *DL* está compuesto de una serie de capas y cada una de estas tiene como función procesar datos de entrada y generar una salida de datos. Las capas se encuentran conectadas

en forma secuencial. Es decir, tiene capa de entrada, seguido de las capas ocultas donde se realiza los cálculos y finalmente una capa de salida, donde se obtiene el resultado. Dentro de las capas ocultas se encuentra una capa de incrustación de palabras (*Embedding Layer*). Esta capa recibe el corpus de texto codificado con la técnica *BoW* visto en la sección 4.2.3 y expande los tokens a un vector de mayor tamaño directamente proporcional al tamaño del vocabulario y a la longitud máxima de la secuencia del texto (texto más largo del corpus).

Después de la capa de incrustación se añaden otras capas como las capas densas o (*dense layer*), capas de regularización *Dropout* que es cuando se omiten aleatoriamente las neuronas de la red neuronal, capa de aplanamiento *Flatten* que es cuando se transforma la matriz multidimensional en un vector unidimensional, capas convolucionales, de *pooling*, recurrentes, etc. Dentro de las capas densas cada neurona recibe una entrada de todas las neuronas de las capas anteriores, a esto se le denomina capa totalmente conectada (*Fully Connected layer*).

En las capas densas es donde están presentes las funciones de activación. Como se señaló en la sección 2.3.3.1 las funciones de activación son las que dictan el comportamiento de las neuronas, es decir, las funciones de activación deciden si una neurona pasa o no a la siguiente capa de neuronas (se activa o no). Existen diferentes tipos de funciones de activación cuya activación depende del rango definido. Además, cada tipo de función de activación tiene su propia particularidad; es decir, algunas funciones de activación trabajan mejor en las capas densas y otras tienen mejor aplicación en las capas finales donde se realiza una clasificación final. Algunos ejemplos de funciones de activación son *sigmoid*, *ReLU*, *tanh*, entre otras.

#### **4.3.1. Configuración de modelos de Aprendizaje Profundo**

Ya que se tiene establecida la arquitectura del modelo, se debe definir cuál será el proceso de aprendizaje. Para esto es necesario especificar el tipo de tasa de aprendizaje (*learning rate*). La tasa de aprendizaje modifica la cantidad en la cual se ajustarán los parámetros durante la optimización logrando minimizar el error en los cálculos de la red neuronal. Existen diferentes métodos de optimización tal como como *Stochastic gradient descent (SGD)*, *Adagrad*, *Adam*, *Adadelta*, etc. Para este proyecto se decidió utilizar el método *RMSProp*.

Una de las configuraciones importantes de cualquier red neuronal *DL* son las funciones de activación (*Activation functions*). Entre las funciones de activación más comunes se pueden



mencionar las funciones *sigmoid*, *tanh*, *Relu*, *Linear*, entre otras más. En esta investigación se utilizó la función de activación *softmax* ya que es la recomendada para tareas de clasificación donde se tienen múltiples clases.

Así mismo, se define la función de pérdida (*loss function*) la cual calcula que tan cerca esta una red neuronal de lo ideal, comparando las predicciones de la red con las predicciones durante el aprendizaje. Cuanto menor es el resultado de esta función, más eficiente es la red neuronal. Existen diferentes tipos de funciones de pérdida como Error cuadrático Medio (*Mean Square Error, MSE*), Error cuadrático absoluto (*Mean absolute error los, MAE*), Error logarítmico cuadrático Medio (*Mean squared log error los, MSLE*), Pérdida de bisagra (*Hinge loss*) o Pérdida logística (*Logistic loss*). Para el actual trabajo se trabajó con función de pérdida entropía cruzada categórica (*Categorical cross entropy*) ya que es de las más idóneas en tareas de clasificación.

Como último paso se debe definir el tipo de métrica para evaluar el modelo durante la fase de entrenamiento y prueba. En la siguiente sección se muestran diferentes tipos de configuraciones de modelos *DL* que se definieron en este proyecto con el objetivo de explorar y encontrar el modelo que ofrece los mejores resultados con el propósito de poder reconocer la personalidad mediante texto.

#### 4.3.2. Perceptrón Multicapa

Esta configuración se llama Perceptrón multicapa y es adecuada para problemas de clasificación. En la Figura 4-15, se observa una muestra de la configuración del *Multilayer Perceptron*.

```
model = Sequential()
model.add(layers.Embedding(max_words, 128, input_length=max_len))
model.add(layers.Flatten())
model.add(layers.Dense(512, activation='relu'))
model.add(layers.Dense(2, activation='softmax')) #model.add(layers.Dense(3, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model.summary()
```

Figura 4-15. Modelo Perceptrón multicapa.

En la figura se observa al modelo Perceptrón multicapa y su configuración donde recibe como parámetros el vector que se obtenido de la técnica *Bag of Words*, donde las frases son codificadas a un formato que puedan ser interpretados y procesados por la red neuronal y así realizar una mejor clasificación o reconocimiento de rasgos de personalidad.

La primera capa tomará como entrada un vector llamado *max\_words*, que indica el total de tokens únicos que existen en nuestro directorio, es decir, el tamaño del vocabulario del corpus normalizado que en total son 2048 tokens. Este vocabulario será utilizado para medir el desempeño de la red neuronal. Por otro lado, se define un vector de salida el cual representa a la dimensión de la incrustación densa, cuyo valor es 128. Este valor se asignó por *default* por ser una potencia de 2. Por último, recibe otra variable de entrada llamada *max\_len*, el cual representa a la longitud de las secuencias de tokens del corpus. Para su elección se considera el vector que contiene mayor cantidad de secuencias de tokens que simboliza el texto más largo del corpus, el cual será de 99 caracteres en esta investigación.

Después de la capa *Embedding*, se agregó una capa aplanada (*Flatten*) la cual tiene como función convertir la entrada de cualquier dimensionalidad a una de 1 x n. Es decir, un vector unidimensional. Posteriormente se utiliza una capa densa totalmente conectada con 512 neuronas y se asigna la función de activación llamada unidad lineal rectificadora (*ReLU*). La función *ReLU* permite la estimación de relaciones no lineales entre los datos.

Por último, se utiliza una capa densa con salida de 2 neuronas y se establece la función de activación *softmax* para realizar la clasificación de presencia o ausencia.

#### 4.3.3. Long Short-Term Memory Simple (LSTM)

El siguiente modelo utilizado en esta investigación se trata de una red neuronal recurrente *LSTM* simple, que también cuenta con una capa *Embedding* la cual se describió anteriormente. Después se agregó la capa *LSTM*, con 15 neuronas de entrada y con *dropout* de 0.5. El propósito es mejorar la precisión del modelo de reconocimiento de personalidad. La configuración del modelo *LSTM* se observa en la Figura 4-16.

```
model1 = Sequential()
model1.add(layers.Embedding(max_words, 20))
model1.add(layers.LSTM(15, dropout=0.5))
model1.add(layers.Dense(2, activation='softmax'))

model1.compile(optimizer='rmsprop', loss='categorical_crossentropy', metrics=['accuracy'])
```

Figura 4-16. Modelo LSTM simple.

El *Dropout* es un método que desactiva aleatoriamente una cierta cantidad de neuronas de una Red neuronal. Con este método se busca reducir el sobreajuste (*overfitting*), es decir, se

quiere evitar que una neurona aprenda características o relaciones de otras neuronas vecinas de la red neuronal ya que estas relaciones pueden formar un patrón muy específico y realizar la clasificación demasiado bien, lo cual es un problema. Un modelo *DL* puede tener *overfitting*, cuando se entrena con muchos datos, y donde se corre el riesgo de que nuestro modelo comience a aprender del ruido y de los datos erróneos del conjunto de datos, provocando que el modelo no realice una correcta clasificación.

#### 4.3.4. LSTM Bidireccional

Este modelo también posee una capa *Embedding*. Después se agrega la capa bidireccional de *LSTM*. Esta capa de *LSTM* tiene 20 neuronas y *dropout* de 0.6. La configuración del modelo *LSTM* Bidireccional (*BiLSTM*) se observa en la Figura 4-17.

```
model2 = Sequential()
model2.add(layers.Embedding(max_words, 40, input_length=max_len))
model2.add(layers.Bidirectional(layers.LSTM(20, dropout=0.6)))
model2.add(layers.Dense(2, activation='softmax'))
model2.compile(optimizer='rmsprop', loss='categorical_crossentropy',
               metrics=['accuracy'])
```

Figura 4-17. Modelo LSTM Bidireccional.

Después se agrega una capa densa que tiene como salida 2 neuronas y con función de activación *softmax*. En todos los modelos *DL* se realizan ajustes y modificaciones de hiperparámetros con la finalidad de mejorar la precisión del reconocedor de personalidad.

Para esto en los modelos *DL* se realiza una serie de combinaciones con los hiperparámetros para lograr mejorar la precisión de los modelos. Dentro de los hiperpámetros que se modificaron está el optimizador *rmsprop*, donde se definió la función de pérdida *categorical\_crossentropy* y se definió que la variable a checar sería la precisión (“*accuracy*”)

#### 4.3.5. Red Neuronal Convolutacional

Las redes neuronales convolucionales (*CNN*) tienen como función dividir y modelar la información que reciben en partes pequeñas y combinan esta información en las siguientes capas del modelo. Este tipo de redes son muy usadas también en tareas de clasificación de texto. La Figura 4-18 muestra la configuración del modelo *CNN*.

```

from keras import regularizers
model3 = Sequential()
model3.add(layers.Embedding(max_words, 40, input_length=max_len))
model3.add(layers.Conv1D(20, 6, activation='relu',
                        kernel_regularizer=regularizers.l1_l2(l1=2e-3, l2=2e-3),
                        bias_regularizer=regularizers.l2(2e-3)))
model3.add(layers.MaxPooling1D(5))
model3.add(layers.Conv1D(20, 6, activation='relu',
                        kernel_regularizer=regularizers.l1_l2(l1=2e-3, l2=2e-3),
                        bias_regularizer=regularizers.l2(2e-3)))
model3.add(layers.GlobalMaxPooling1D())
model3.add(layers.Dense(2, activation='softmax'))
model3.compile(optimizer='rmsprop', loss='categorical_crossentropy', metrics=['acc'])
metric = 'val_acc'

```

Figura 4-18. Modelo de CNN.

La red CNN (Conv1D) recibe el vector de la capa *Embedding* y sobre este se aplica un *kernel* que retorna un mapa de las características del texto original. En la capa Conv1D, el mapa también llamado filtro de salida en la convolución tiene un tamaño de 40, y el total de filtros de este tipo son 6. Por otro lado, se definió como activación la función *ReLU*. Para el tipo de desplazamiento (*stride*) se utilizó el valor que viene definido por default.

La siguiente capa es una capa de reducción la cual es llamada *MaxPooling1D*. Esta capa tiene como objetivo reducir las dimensiones de los datos de entrada para la siguiente capa. Esta capa divide los datos en recuadros y selecciona solo el valor máximo. El tamaño de este recuadro es de 5.

# Capítulo 5

## 5. Pruebas y Resultados

En este capítulo se muestran los resultados obtenidos de las pruebas realizadas al corpus *PersonText*, además se observa la fiabilidad del corpus como conjunto de datos en los modelos *DL* con enfoque en el de reconocimiento de personalidad mediante texto propuestos en el capítulo anterior.

También se muestran pruebas realizadas con nuestra misma metodología propuesta pero implementada a otros corpus importantes de dominio público utilizados en temas de reconocimiento de personalidad bajo el modelo *Big-Five*.

### 5.1. Métrica de evaluación

La métrica de evaluación se trata de la interpretación los resultados y para ello se hace uso de medidas para calcular el error obtenido. Esto tiene como finalidad validar el grado de fiabilidad del modelo creado. Por lo tanto, se puede definir una métrica de evaluación como una función que se usa para medir el rendimiento del modelo *DL*.

Existen diferentes tipos de métricas de evaluación como la pérdida logarítmica, área bajo la curva, matriz de confusión, exactitud de clasificación, etc. La métrica con la que se evaluó esta investigación es la exactitud de clasificación (*accuracy*). La exactitud es la relación entre el número de predicciones correctas y el número total de predicciones realizadas. Esta fórmula se muestra en la función (2).

$$Exactitud = \frac{\text{Número de predicciones correctas}}{\text{Total de predicciones realizadas}} \quad (2)$$

### 5.2. Separación de datos de Entrenamiento y Validación

Actualmente el corpus *PersonText* tiene 213 textos con sus respectivas etiquetas e inventario de resultados de pruebas IPIP. Para validar el corpus se realizaron las pruebas con los modelos creados. Para realizar el entrenamiento y la evaluación de los modelos *DL*, se dividió

el corpus *PersonText* en un 75% de datos utilizados para el entrenamiento de los modelos y 25% restante para la validación de estos. El entrenamiento dura 50 épocas y la métrica a vigilar es la validación de exactitud, guardando el mejor modelo cuando se encuentre uno mejor que el anterior.

### 5.3. Entrenamiento y Evaluación de modelos DL

Para poder evaluar el corpus *PersonText* se realizaron pruebas en varios modelos *DL*, con el objetivo de poder afirmar si el corpus es confiable como conjunto de datos en tareas de reconocimiento de la personalidad a partir de texto. Para calcular la fiabilidad de los modelos se tomó como medida la exactitud.

En un modelo *ML* o *DL* mientras menor sea la pérdida, mejor es el modelo, es decir, entre más cerca del “0” este la pérdida calculada, el modelo es más fiable. Esta pérdida se trata de una suma de errores existentes en cada época durante el entrenamiento y validación. Por otra parte, la exactitud de un modelo *DL*, se calcula después de que el modelo aprendió e hizo ajustes de los parámetros. En otras palabras, la exactitud calcula el porcentaje de error que obtuvo el modelo. En la Figura 5-1 se muestra el código que realizó la tarea de validación de los modelos de reconocimiento de personalidad.

```
#Cargamos uno de los modelos

best_model = keras.models.load_model("best_model1.hdf5")

test_loss, test_acc = best_model.evaluate(X_test, y_test, verbose=2)
print('Model accuracy: ',test_acc)

2/2 - 1s - loss: 0.6814 - accuracy: 0.6296 - 1s/epoch - 636ms/step
Model accuracy: 0.6296296119689941

predictions = best_model.predict(X_test)
```

Figura 5-1. Validación de los modelos.

Durante el entrenamiento de modelos *DL*, se debe buscar evitar el sobre ajuste del modelo. Para ello, se utilizan técnicas como detección anticipada (*early-stopping*). Esta técnica evalúa el error de la red época tras época y detendrá automáticamente el entrenamiento del modelo cuando detecte una reducción de la pérdida de validación, conservando la mejor pérdida

obtenida en alguna época pasada. La Figura 5-2 muestra el código de cómo se implementa la fase de entrenamiento y la técnica de detección anticipada.

```
earlyStopping = keras.callbacks.EarlyStopping(monitor='val_accuracy', patience=5, verbose=1, mode='min')
mcp = ModelCheckpoint("best_model0.hdf5", monitor="val_accuracy",
                    save_best_only=True, save_weights_only=False, verbose=1)
model_history = model.fit(X_train, y_train, verbose=1,
                        validation_data = (X_test, y_test), epochs=10,
                        callbacks=[earlyStopping,mcp], batch_size= 64, shuffle=True)
```

Figura 5-2. Técnica early-stopping.

Otra prueba que se realizó para la validación del corpus *PersonText*, fue comparar los resultados obtenidos del entrenamiento y la validación, contra los resultados del entrenamiento y validación de otros corpus importantes utilizados en tareas de reconocimiento de personalidad basados en el modelo de los cinco grandes. Estos corpus son de dominio público y para probarlos se les aplicó la misma metodología y modelos *DL* propuestos en esta investigación. Los otros corpus de texto utilizados para la comparación fueron: *myPersonality* y *essays dataset*, los cuales se encuentran en idioma inglés.

Para hacer dicha comparación entre los tres corpus evitando confundir a la red neuronal y obtener mejores resultados se realizaron entrenamientos individuales para cada atributo de los rasgos de personalidad. En decir, para calcular o reconocer el rasgo de amabilidad se ocupó una sola red neuronal, para el atributo extrovertido otra red y de la misma forma para el resto de los rasgos. Cada red neuronal tiene la misma configuración en cuanto a capas, numero de neuronas, parámetros, etc.

Después de los entrenamientos de las redes neuronales utilizando los tres corpus de texto diferentes, se obtuvo como resultado un total de 15 redes neuronales correspondiente a cada uno de los cinco rasgos de personalidad del modelo *OCEAN*.

### 5.3.1. Entrenamiento y Evaluación del Corpus *PersonText*

La primera prueba realizada fue validar el corpus *PersonText*. Para ello se dividió el conjunto de datos en un 75% de datos para el entrenamiento de la red neuronal y el restante para validación del modelo.

Antes de realizar la división del corpus, previamente se le hizo un preprocesamiento y limpieza del corpus. Este preprocesamiento se vio a detalle en la sección 4.2.2, la cual consta

de eliminación de números, espacios en blanco, convertir letras a minúsculas, remover comillas, eliminar caracteres no alfanuméricos, remover palabras innecesarias que no aportan sentido al texto (*Stop Words*), realizar Lematización, que es el proceso de llevar las palabras a su nivel base y por último remover acentos. Posteriormente se extrajeron las características de los textos con el método de bolsa de palabras. El mejor resultado obtenido en cuanto a *accuracy* fue de 0.74074, es decir, se alcanzó un 74% de exactitud en la validación del modelo.

### **5.3.2. Entrenamiento y Evaluación del Corpus MyPersonality**

Para probar la metodología propuesta, así como la configuración de nuestros modelos de reconocimiento, se decidió utilizar otro corpus de texto llamado *myPersonality* que también está relacionado con el modelo *Big-Five*. Este corpus de igual manera sirvió de conjunto de datos para el entrenamiento de los modelos. Así como al corpus *PersonText*, también a *myPersonality* se le realizó un preprocesamiento para eliminar el ruido antes de introducir los datos a los modelos *DL*. Para este corpus se tuvo que adaptar algunas operaciones del preprocesamiento ya que el corpus se encuentra en idioma inglés. La limpieza o preprocesamiento realizados fueron las siguientes:

- a) Filtrar textos cortos: debido a que el propósito de la investigación es reconocer la personalidad por medio de textos, se decidió eliminar del corpus los textos cortos, es decir, eliminar textos que contengan 20 palabras o menos. Esto porque en el corpus vienen muchos textos cortos los cuales no nos ayudan o tienen pocos datos que ayuden a identificar características que denoten o resalten la personalidad de un individuo mediante texto.
- b) Convertir palabras a minúsculas.
- c) Expandir contracciones: En la escritura del idioma inglés, existen una serie de palabras que se encuentran contraídas, es decir, son palabras (por lo regular dos palabras, un sujeto y un verbo) que se encuentran contraídas y se utilizan para expresar lo mismo de una manera abreviada o resumida. Un ejemplo de este tipo de palabras contraídas es la palabra “wasnt”, la cual es una contracción de “was not”. Debido a esto se creó un diccionario de palabras de palabras contraídas, agregando las palabras de las cuales se deriva. Después se elaboró un pequeño código para



buscar estas palabras contraídas en las palabras de los textos del corpus y en caso de encontrarlas, sustituirlas por las palabras de las cuales es contracción. Otros ejemplos de estas contracciones son: “*we’d*”: *we would*”, “*it’s*”: *it is*”, “*don’t*”: *do not*”, “*i’d*”: *i would*”, “*i’m*”: *i am*”.

- d) Eliminar caracteres repetidos: Se limitó la repetición de caracteres a solo dos caracteres iguales, seguidos uno del otro. En inglés es común tener conjugaciones de un verbo en infinitivo a presente continuo o futuro, por ejemplo, “*swim*” conjugado a presente continuo sería “*swimming*”. Donde la letra “m” se está repitiendo dos veces. Esto también ayudará a reducir palabras que estén incorrectamente escritas como: “*Realllllllly*”, resultando en “*Really*”.
- a) Eliminar nuevas líneas y pestañas: se eliminan los posibles comandos de nueva línea o saltos de línea que pueda contener el corpus, así como las pestañas nuevas. (“\n”, “\s”, “\t”, “\\”,).
- b) Remover comillas simples (“”) y las comas simples (,).
- c) Omitir los números.
- d) Remover Caracteres no alfanuméricos (\$, %, #,!).
- e) Eliminar guiones entre palabras, eliminar los guiones dobles y triples: pudiera haber palabras que tengan un guion entre palabras, como “*science-fiction*” o “*self-centered*”. Resultando en “*science fiction*” y “*self centered*” respectivamente.
- f) Eliminar espacios en blanco en los textos: puede suceder que tanto al principio como final de cada texto existan espacios blancos o incluso espacios en blanco dobles entre palabras, por lo que se procede a eliminar cualquiera de estas situaciones.
- g) Remover *Stopwords*: para este caso de las palabras vacías, se descargaron las *stopwords* que vienen por *default* de la biblioteca *nlk* y se eliminaron de cada texto en caso de existir coincidencia.
- h) Realizar lematización
- i) Remover signos de puntuación.

Los mejores resultados obtenidos del entrenamiento y validación de los modelos de clasificación de atributos de personalidad mediante texto utilizando el corpus *myPersonality* fueron los siguientes: utilizando una red *LSTM Bidireccional* se alcanzó una precisión de 72.66% y con una red LSTM se obtuvo una precisión de 72.30%.

### 5.3.3. Entrenamiento y Evaluación del Corpus Essays

Para el entrenamiento y validación de los modelos *DL*, tomando el corpus *essays* como conjunto de datos, se desarrolló el mismo procedimiento realizado al corpus *myPersonality*, ya que este corpus también se encuentra en idioma inglés y poseen características similares. Antes de hacer el entrenamiento se hizo el mismo preprocesamiento de los textos al corpus. El mejor resultado obtenido con el corpus *essays* fue el siguiente: 57.53% de exactitud utilizando una red neuronal *LSTM* Bidireccional bajo la misma metodología y preprocesamiento para el conjunto de datos en idioma inglés.

## 5.4. Resultados Finales

Para poder calcular cada rasgo de personalidad se utilizaron 5 redes de cada modelo independientes correspondientes a cada rasgo de personalidad resultando un total de 15 redes. Cada red tiene la misma estructura, la misma configuración en cuanto a capas, numero de neuronas, así como parámetros.

Los resultados obtenidos de los entrenamientos y validación de cada red utilizando tanto el corpus creado *PersonText* como los corpus más populares para entrenamiento de modelos de reconocimiento de personalidad de otras investigaciones se observan en la Tabla 5-1.

Tabla 5-1. Exactitud (accuracy) obtenida de pruebas con diferentes corpus.

Corpus	Clasificador	Rasgos de Personalidad – Exactitud (Accuracy)				
		Apertura	Responsabilidad	Sociabilidad	Amabilidad	Neuroticismo
PersonText	MLP	0.6852	0.6481	0.6481	0.6852	0.6481
PersonText	LSTM	<b>0.74074</b>	0.68519	<b>0.64815</b>	<b>0.72222</b>	<b>0.74074</b>
PersonText	LSTM Bidireccional	0.6852	<b>0.7037</b>	0.6296	<b>0.7222</b>	0.7037
PersonText	CNN 1D	0.6667	0.6667	0.6296	0.7037	0.7037
myPersonality	MLP	0.72302	0.57734	0.57914	0.53777	0.5971
myPersonality	LSTM	0.72302	<b>0.65468</b>	0.61691	0.60072	<b>0.6043</b>
myPersonality	LSTM Bidireccional	<b>0.72662</b>	0.65647	<b>0.63849</b>	<b>0.60971</b>	0.60252
myPersonality	CNN 1D	0.72302	0.55935	0.57914	0.49101	0.59712
essays	MLP	0.52188	0.51540	0.50243	0.50567	0.52512
essays	LSTM	0.54457	<b>0.52998</b>	0.52512	0.53323	<b>0.54781</b>

essays	LSTM Bidireccional	<b>0.57536</b>	0.52998	<b>0.53323</b>	<b>0.56240</b>	0.54295
essays	CNN 1D	0.50405	0.48622	0.51540	0.50729	0.47164

Como se observa en la tabla se hicieron pruebas para reconocer los cinco rasgos de personalidad automáticamente (Apertura, Responsabilidad, sociabilidad, Amabilidad y Neuroticismo). Se configuraron 4 redes neuronales diferentes para hacer las pruebas y se utilizaron tres corpus de texto diferentes para el entrenamiento de los modelos. Los resultados que se encuentran resaltados en negritas de la tabla corresponden a los mejores resultados obtenidos para cada una de las combinaciones.

El mejor resultado obtenido para el corpus *PersonTex* fue con una red neuronal LSTM cuya exactitud fue un 74% y el resultado más bajo obtenido corresponde a un 47%.

# Capítulo 6

## 6. Conclusiones y trabajo futuro

En esta sección se describen las conclusiones y trabajo futuro de esta investigación. Así mismo se establecen las limitaciones y principal aporte de esta tesis. Finalmente, se describe el trabajo futuro que podría fortalecer este proyecto de investigación.

### 6.1. Conclusiones

Este trabajo se basó en la hipótesis de esta de investigación, la cual consiste en validar si es posible crear un corpus de textos en español que pueda ser utilizado en modelos *DL* para reconocer la personalidad de un individuo con base en el modelo *Big Five*. Nuestra investigación comenzó por la definición de la estructura del corpus *PersonText*. Este conjunto de datos se generó con información extraída de la plataforma *PersonApp*, la cual alberga información como videos de los participantes hablando de temas diversos; también contienen los resultados de pruebas IPIP relacionados con la teoría de los cinco grandes. Una vez creado el corpus se le aplicó un trabajo de preprocesamiento para posteriormente realizar pruebas con él y validar la hipótesis formulada.

El método para validarlo es utilizando modelos *DL* tomando el corpus para entrenamiento. Dentro de los modelos se probó con redes neuronales como *MLP*, *LSTM*, *LSTM Bidireccional* y *CNN 1D*, a las cuales se le definió su configuración y arquitectura, así como diversas técnicas para mejorar la precisión o exactitud del modelo. Para evitar confundir o generar ruido en la red neuronal reconociendo los cinco rasgos de personalidad simultáneamente, se optó por entrenar 5 redes independientes correspondientes a cada rasgo de personalidad (amabilidad, apertura, responsabilidad, sociabilidad y neuroticismo). Resultando en un total de 15 redes neuronales debido a que se probó la metodología y los mismos modelos con tres conjuntos de datos diferentes. El mejor resultado fue un 74% de exactitud (*accuracy*) en la validación.

Durante la realización de esta investigación hubo varias lecciones aprendidas, las cuales sirvieron para mejorar la exactitud de los modelos de reconocimiento. Dentro de estas lecciones se mencionan las siguientes:

- a) Al principio de la investigación se intentó calcular el valor numérico de cada rasgo de personalidad obteniendo como resultado un 33% de exactitud, por lo que se tomó la decisión de hacer un cambio de enfoque. Este nuevo cambio radicó en pasar de un problema de regresión (cálculos numéricos) a un problema de clasificación binaria con etiquetas (Si / No); indicando si existe o no la presencia del rasgo de personalidad en el texto. Esto mejoró notablemente los resultados pasando de 33% de precisión inicial a un 60% (solo con este cambio de enfoque). Posteriormente se buscó incrementar este resultado.
- b) Otro punto que se puede concluir con respecto a las tareas de clasificación de personalidad mediante el texto es que se debe definir muy bien el lugar de procedencia donde se extraen los datos para crear el corpus, ya que hay textos que contienen pocas palabras lo que hace un poco desbalanceado al corpus generando así un decremento en la precisión para el reconocimiento de personalidad. Para esto se debe formular buenas preguntas a los autores con la finalidad de obtener textos que tengan relación al objetivo de la investigación.
- c) Otra lección aprendida de vital importancia para mejorar la exactitud de los modelos *DL* para el reconocimiento de personalidad mediante texto, es hacer un correcto preprocesamiento de los textos. Para ello se deben elegir correctamente las acciones de limpieza al corpus, como la eliminación de números o signos de puntuación, conversión de palabras a minúsculas, eliminación de espacios en blanco, eliminación de palabras vacías (*stopwords*), disminución de la cantidad de verbos para llevarlos a la raíz (*stemming*) o forma infinitiva (lematización).
- d) Al seleccionar bien los hiperparámetros en la configuración de los modelos *Deep Learning* hizo incrementar la precisión del modelo, por lo que se debe optar por hacer pruebas con diferentes optimizadores y funciones de pérdida, tasa de aprendizaje, así como el número de neuronas por capa y funciones de activación para lograr mejorar los modelos.

- e) Se recomienda crear minuciosamente una lista propia de palabras vacías (*stopwords*), ya que son las palabras que se retirarán de los textos por no aportar nada importante a los textos. Esto es importante ya que hay muchas palabras que están mal escritas por los usuarios, o se encuentran en otros idiomas, por lo que, si no son eliminadas, crean ruido en los textos que serán los datos de entrada.
- f) Otro punto importante y que sirvió para aumentar un 2% más la exactitud del modelo, aun cuando se pensaba que no se podía mejorar, fue elegir minuciosamente el orden de las tareas del preprocesamiento.

Una de las conclusiones principales que observan en este trabajo, es que se obtuvo una mejor exactitud de los modelos donde se usó del corpus *PersonText* en comparación con los otros modelos donde se utilizaron diferentes corpus de texto. Se piensa que esto se debe a que el corpus *PersonText* fue creado a partir de los datos obtenidos de la plataforma *PersonApp*, la cual es una plataforma que tiene como función principal recolectar información y asignar valores a los rasgos de personalidad de cada participante mediante la realización de una prueba estandarizada llamada IPIP basándose en el modelo *Big-Five*.

Una de las limitaciones que se presentaron durante la realización de esta investigación fue que no existen muchos trabajos relacionados con construcción de corpus para reconocimiento de personalidad mediante texto y los conjuntos de datos existentes son pocos. Además, los datos que contienen estos conjuntos de datos fueron extraídos de redes sociales como *Facebook* o *Twitter*, sitios, donde la gente puede estar hablando de forma sarcástica o en doble sentido, o de manera muy informal o coloquial.

Por otra parte, estos corpus de texto se encuentran en un idioma inglés, por lo que se hizo más difícil realizar su preprocesamiento. Otra limitación es que no existen corpus de texto utilizados para el reconocimiento de personalidad en idioma español por lo que nuestro trabajo de investigación es muy importante ya que se generó un corpus de texto en español a partir de una metodología totalmente sustentada bajo el modelo *OCEAN* convirtiéndose así en un valioso aporte para la comunidad científica.

Después de haber comparados nuestros resultados obtenidos en las pruebas, realizadas con las distintas configuraciones, así como los diferentes tipos corpus de texto, podemos señalar que la teoría propuesta para esta investigación es verdadera, ya que se obtienen resultados

similares y en algunos casos se obtuvo mejor exactitud utilizando el corpus *PersonText* como base de datos para su entrenamiento.

## **6.2. Trabajo a futuro**

Para trabajo a futuro se proponen las siguientes tareas:

1. Incrementar el tamaño del corpus *PersonText* que actualmente cuenta con 213 textos.
2. Explorar la incorporación textos de otros conjuntos de datos para incrementar también el tamaño del corpus.
3. Desarrollar otras técnicas para obtener datos de texto directamente desde la plataforma PersonApp.
4. Implementar modelos de clasificación más robustos y probar con diferentes parámetros durante los procesos de entrenamiento y validación.

# Referencias

- Albawi, S., & Ucan, O. (2018). Social Touch Gesture Recognition Using Convolutional Neural Network. *Computational Intelligence and Neuroscience*, 2018. <https://doi.org/10.1155/2018/6973103>
- Allport, G. W., & Odbert, H. S. (1936). Trait-names: A psycho-lexical study. *Psychological Monographs*, 47(1), i.
- Allport, G. W. (1937). *Personality: A psychological interpretation*. New York: Holt.
- Alu, D., Zoltan, E., & Stoica, I. C. (2017). Voice based emotion recognition with convolutional neural networks for companion robots. *Science and Technology*, 20(3), 222–240.
- Barron Estrada, M., Zatarain Cabada, R., & Oramas, R. (2019). Emotion Recognition for Education using Sentiment Analysis. *Research in Computing Science*, 148, 71–80. <https://doi.org/10.13053/rcs-148-5-8>
- Bátiz Beltrán, V. M. (2021). Trabajo de tesis: Sistema de evaluación de la personalidad y de las emociones en el proceso cognitivo. Tecnológico Nacional de México campus Culiacán.
- Bátiz Beltrán, V. M., Zatarain Cabada, R., Barrón Estrada, M. L., Cárdenas López, H. M., & Escalante, H. M. (2022). A multiplatform application for automatic recognition of personality traits for Learning Environments. *International Conference on Advanced Learning Technologies (ICALT)*, 2022, pp. 49-50, doi: 10.1109/ICALT55010.2022.00022.
- Biel, J.-I., Tsiminaki, V., Dines, J., & Gatica-Perez, D. (2013). Hi YouTube! Personality Impressions and Verbal Content in Social Video. In *ICMI 2013 - Proceedings of the 2013 ACM International Conference on Multimodal Interaction*. <https://doi.org/10.1145/2522848.2522877>
- Busso, C., Yildirim, S., Bulut, M., Lee, C., Kazemzadeh, A., Lee, S., Neumann, U., & Narayanan, S. (2004). Analysis of emotion recognition using facial expressions, speech and multimodal information. *ICMI'04 - Sixth International Conference on Multimodal Interfaces*, 205–211. <https://doi.org/10.1145/1027933.1027968>
- Calefato, F., Lanubile, F., & Novielli, N. (2017). *EmoTxt: A Toolkit for Emotion Recognition from Text*.
- Caruso, D. R., & Salovey, P. (2005). El directivo emocionalmente inteligente: cómo desarrollar y utilizar las cuatro técnicas emocionales claves del liderazgo.



- Cattell, R, Eber, H. W., & Tatsuoka, M. M. (1970). Handbook for the 16PF. Champaign, IL: Institute for Personality and Ability Testing.
- Chollet, F. (2017). Deep learning with python, vol. 1. Greenwich, CT: Manning Publications CO.
- Chowdhury, G. G. (2003). Natural language processing. Annual review of information science and technology, 37(1), 51-89.
- Conway, D., & White, J. (2012). Machine learning for hackers. “ O’Reilly Media, Inc.”
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.
- Eysenck, H. J., & Himmelweit, H. T. (1947). Dimensions of personality; a record of research carried out in collaboration with H.T. Himmelweit and others. London: Routledge & Kegan Paul.
- Eysenck, H. J., & Eysenck, S. B. G. (1976). Psychoticism as a dimension of personality. London: Hodder & Stoughton.
- Feldman, R. (2013). Techniques and applications for sentiment analysis. Communications of the ACM, 56(4), 82–89.
- Fellbaum, C. (1998). WordNet: An electronic lexical database. MIT press.
- Firebase. (2022). Recuperado el 25 de Octubre de 2022, de <https://firebase.google.com/?hl=es>
- Fiske, D. W. (1949). Consistency of the factorial structures of personality ratings from different sources. The Journal of Abnormal and Social Psychology, 44(3), 329.
- Funder D. C. (2013). The personality puzzle. W.W. Norton & Company.
- Giménez, M., Paredes, R., & Rosso, P. (2017). Personality recognition using convolutional neural networks. International Conference on Computational Linguistics and Intelligent Text Processing, 313–323.
- Goldberg, L. (1981). Language and individual differences: The search for universals in personality lexicons. In L. Wheeler (Ed.). Review of personality and social psychology (Vol. 2, pp. 141-165). Beverly Hills: Sage.
- Goldberg, L. (1992). The Development of Markers For the Big Five Factor Structure. Psychological Assessment, 4, 26–42. <https://doi.org/10.1037/1040-3590.4.1.26>
- Goldberg, L. (1999). A broad-bandwidth, public-domain, personality inventory measuring the lower-level facets of several five-factor models. Personality Psychology in Europe, 7, 7-28. Personality Psychology in Europe, 7, 7–28.
- Goodfellow, I., Bengio, Y., & Courville, A. (2017). Deep Learning. Adaptive computation and machine learning series. The MIT Press.

- Google colaboratory. (2022). Recuperado el 25 de Octubre de 2022, de <https://colab.research.google.com/>
- Graves, A. (2012). Supervised Sequence Labelling. In A. Graves (Ed.), *Supervised Sequence Labelling with Recurrent Neural Networks* (pp. 5–13). Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-642-24797-2\\_2](https://doi.org/10.1007/978-3-642-24797-2_2)
- Gross, J., & Barrett, L. (2011). Emotion Generation and Emotion Regulation: One or Two Depends on Your Point of View. *Emotion Review*, 3, 8–16. <https://doi.org/10.1177/1754073910380974>
- Gürpınar, F., Kaya, H., & Salah, A. (2016). Combining Deep Facial and Ambient Features for First Impression Estimation. [https://doi.org/10.1007/978-3-319-49409-8\\_30](https://doi.org/10.1007/978-3-319-49409-8_30)
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-term Memory. *Neural Computation*, 9, 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Hockenbury, D. H., & Hockenbury, S. E. (1997). *Psychology*. New York, Worth Publishers.
- Huang, G.-B., Zhou, H., Ding, X., & Zhang, R. (2011). Extreme learning machine for regression and multiclass classification. *IEEE Trans. Syst. Man Cybern.* 42(2), 513-529. *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics : A Publication of the IEEE Systems, Man, and Cybernetics Society*, 42, 513–529. <https://doi.org/10.1109/TSMCB.2011.2168604>
- International Personality Item Pool: A Scientific Collaboratory for the Development of Advanced Measures of Personality Traits and Other Individual Differences. (2021). Recuperado el 1 de abril de 2021 de <http://ipip.ori.org/>.
- John, O., Naumann, L., & Soto, C. (2008). Paradigm shift to the integrative big five trait taxonomy: History, measurement, and conceptual issues. In *Handbook of Personality: Theory and Research*, 3 Edn. (pp. 114–158).
- John, O., & Srivastava, S. (1999). The Big-Five Trait Taxonomy: History, Measurement, and Theoretical Perspectives. In *Handbook of Personality: Theory and Research* (Vol. 2).
- Kampman, O., Barezi, E. J., Bertero, D., & Fung, P. (2018). Investigating audio, visual, and text fusion methods for end-to-end automatic personality prediction. *ArXiv Preprint ArXiv:1805.00705*.
- Kim, H. K., Kim, H., & Cho, S. (2017). Bag-of-Concepts: Comprehending Document Representation through Clustering Words in Distributed Representation. *Neurocomputing*, 266. <https://doi.org/10.1016/j.neucom.2017.05.046>
- Kirk, M. (2014). *Thoughtful machine learning: A test-driven approach*. “O’Reilly Media, Inc.”
- Kosinski, M., Matz, S., Gosling, S., Popov, V., & Stillwell, D. (2015). Facebook as a Research Tool for the Social Sciences: Opportunities, Challenges, Ethical Considerations, and Practical Guidelines. <https://doi.org/10.17863/CAM.8>

- Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging Artificial Intelligence Applications in Computer Engineering*, 160(1), 3–24.
- Kulkarni, A., & Shivananda, A. (2019). Natural Language Processing Recipes: Unlocking Text Data with Machine Learning and Deep Learning using Python. <https://doi.org/10.1007/978-1-4842-4267-4>
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.
- Li, C., & Goldwasser, D. (2019). Encoding social information with graph convolutional networks for political perspective detection in news media. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2594–2604.
- Liu, B. (2015). Sentiment Analysis: Mining Opinions, Sentiments, and Emotions. <https://doi.org/10.1017/CBO9781139084789>
- Liu, F., Perez, J., & Nowson, S. (2016). A recurrent and compositional model for personality trait recognition from short texts. *Proceedings of the Workshop on Computational Modeling of People’s Opinions, Personality, and Emotions in Social Media (PEOPLES)*, 20–29.
- Mairesse, F., Walker, M. A., Mehl, M. R., & Moore, R. K. (2007). Using linguistic cues for the automatic recognition of personality in conversation and text. *Journal of Artificial Intelligence Research*, 30, 457–500.
- Majumder, N., Poria, S., Gelbukh, A., & Cambria, E. (2017). Deep Learning-Based Document Modeling for Personality Detection from Text. *IEEE Intelligent Systems*, 32, 74–79. <https://doi.org/10.1109/MIS.2017.23>
- McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4), 115–133.
- Mehl, M., Pennebaker, J., Crow, D., Dabbs, J., & Price, J. (2001). The Electronically Activated Recorder (EAR): A device for sampling naturalistic daily activities and conversations. *Behavior Research Methods, Instruments, & Computers : A Journal of the Psychonomic Society, Inc*, 33, 517–523.
- Merriam-Webster. (2022). Sentiment. In Merriam-Webster.com dictionary. Recuperado el 10 de octubre de 2022, de <https://www.merriam-webster.com/dictionary/sentiment>
- Michalski, R. S., Carbonell, J. G., & Mitchell, T. M. (2013). Machine learning: An artificial intelligence approach. Springer Science & Business Media.
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013a). Efficient estimation of word representations in vector space. *ArXiv Preprint ArXiv:1301.3781*.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26.

- Moffitt, K. C., Giboney, J. S., Ehrhardt, E., Burgoon, J. K., & Nunamaker, J. F. (2012). Structured programming for linguistic cue extraction (SPLICE). In M. L. Jensen, T. O. Meservy, J. K. Burgoon, & J. F. Nunamaker (Eds.), *Proceedings of the HICSS45 Rapid Screening Technologies, Deception Detection and Credibility Assessment Symposium* (pp. 103-108). Retrieved from [http://www.hicss.hawaii.edu/Reports/HICSS45\\_RapidScreeningTechnologiesDeceptionDetectionandCredibilityAssessmentSymposium.pdf](http://www.hicss.hawaii.edu/Reports/HICSS45_RapidScreeningTechnologiesDeceptionDetectionandCredibilityAssessmentSymposium.pdf)
- Mohades, D. F., Sadr, H., & Nazari, M. (2021). Using machine learning-based models for personality recognition. *Big Data and Computing Visions*, 1(3), 128–139.
- Morris, R., Tarassenko, L., & Kenward, M. (2006). Cognitive systems - information processing meets brain science. <https://doi.org/10.1016/B978-0-12-088566-4.X5000-4>
- Myers, I., McCaulley, M., Quenk, N., & Hammer, A. (1998). *MBTI Manual, A Guide to the Development and Use of the Myers-Briggs Type Indicator*.
- Negnevitsky M. (2011). *Artificial intelligence: a guide to intelligent systems* (3rd ed.). Addison Wesley/Pearson.
- Nwana, H. (1990). Intelligent Tutoring Systems: an overview. *Artificial Intelligence Review*, 4, 251–277. <https://doi.org/10.1007/BF00168958>
- Ong, V., Rahmanto, A. D. S., Suhartono, D., Nugroho, A. E., Andangsari, E. W., & Suprayogi, M. N. (2017). Personality prediction based on Twitter information in Bahasa Indonesia. *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 367–372.
- Osisanwo, F. Y., Akinsola, J. E. T., Awodele, O., Hinmikaiye, J. O., Olakanmi, O., & Akinjobi, J. (2017). Supervised machine learning algorithms: classification and comparison. *International Journal of Computer Trends and Technology (IJCTT)*, 48(3), 128–138.
- Pabón, F., & Orozco, J. R. (2021). Automatic Personality Evaluation from Transliterations of YouTube Vlogs Using Classical and State-of-the-Art Word Embeddings. *Ingeniería e Investigación*, 42, e93803. <https://doi.org/10.15446/ing.investig.93803>
- Patterson, J., & Gibson, A. (2017). *Deep learning: A practitioner's approach*. “O'Reilly Media, Inc.”
- Pennebaker, J., & King, L. (2000). Linguistic styles: Language use as an individual difference. *Journal of Personality and Social Psychology*, 77, 1296–1312. <https://doi.org/10.1037//0022-3514.77.6.1296>
- Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.
- Picard, R. W. (1997). *Affective computing*. In *Affective computing*. The MIT Press.

- Plutchik, R. (1979). Emotions: A general psychoevolutionary theory. *Emotion: Theory, Research, and Experience*, 1. <https://doi.org/10.1016/B978-0-12-558701-3.50007-7>
- Qader, W., M. Ameen, M., & Ahmed, B. (2019). An Overview of Bag of Words; Importance, Implementation, Applications, and Challenges. <https://doi.org/10.1109/IEC47844.2019.8950616>
- Rammstedt, B., & John, O. (2007). Measuring Personality in One Minute or Less: A 10-Item Short Version of the Big Five Inventory in English and German. *Journal of Research in Personality*, 41, 203–212. <https://doi.org/10.1016/j.jrp.2006.02.001>
- Ramos dos Santos, W., & Paraboni, I. (2019). Personality Facets Recognition from Text (pp. 185–190). [https://doi.org/10.1007/978-3-030-28577-7\\_15](https://doi.org/10.1007/978-3-030-28577-7_15)
- Ramos, R., Neto, G., Silva, B., Monteiro, D., Paraboni, I., & Dias, R. (2018). Building a corpus for personality-dependent natural language understanding and generation. *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
- Rangel, F., Rosso, P., Potthast, M., Stein, B., & Daelemans, W. (2015). Overview of the 3rd Author Profiling Task at PAN 2015. *CLEF*, 2015.
- Ríssola, E., Bahrainian, S. A., & Crestani, F. (2019). Personality Recognition in Conversations using Capsule Neural Networks. In *WI '19: IEEE/WIC/ACM International Conference on Web Intelligence*. <https://doi.org/10.1145/3350546.3352516>
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533–536.
- Sarkar, C., Bhatia, S., Agarwal, A., & Li, J. (2014). Feature analysis for computational personality recognition using youtube personality data set. *Proceedings of the 2014 ACM Multi Media on Workshop on Computational Personality Recognition*, 11–14.
- Seyeditabari, A., Tabari, N., & Zadrozny, W. (2018). Emotion Detection in Text: a Review.
- Siddique, F., & Fung, P. (2017). Bilingual Word Embeddings for Cross-Lingual Personality Recognition Using Convolutional Neural Nets. <https://doi.org/10.21437/Interspeech.2017-1379>
- Sinclair, J. (1996). Preliminary Recommendations on Corpus Typology. EAGLES Document EAG TCWG-CTYP/P. Available at: <http://www.ilc.cnr.it/EAGLES96/corpusyp/corpusyp.html>. [Accessed 13/10/2011]
- Schultz, D., & Schultz, S. (2013). *Theories of Personality*. (10th ed.). California: Wadsworth Cengage Learning
- Sun, X., Liu, B., Cao, J., Luo, J., & Shen, X. (2018). Who am I? Personality detection based on deep learning for texts. *2018 IEEE International Conference on Communications (ICC)*, 1–6.

- Swamynathan, M. (2017). Mastering Machine Learning with Python in Six Steps. <https://doi.org/10.1007/978-1-4842-2866-1>
- Tandera, T., Suhartono, D., Wongso, R., & Prasetyo, Y. L. (2017). Personality prediction system from facebook users. *Procedia Computer Science*, 116, 604–611.
- Thomas, E., & Galambos, N. (2004). What Satisfies Students? Mining Student-Opinion Data with Regression and Decision Tree Analysis. *Research in Higher Education*, 45, 251–269. <https://doi.org/10.1023/B:RIHE.0000019589.79439.6e>
- Tian, L., Wong, D., Chao, L., & Oliveira, F. (2014). A Relationship: Word Alignment, Phrase Table, and Translation Quality. *TheScientificWorldJournal*, 2014, 438106. <https://doi.org/10.1155/2014/438106>
- Tinwala, W., & Rauniyar, S. (2021). Big Five Personality Detection Using Deep Convolutional Neural Networks. <https://doi.org/10.20944/preprints202109.0199.v1>
- Tkalčič, M., Carolis, B., de Gemmis, M., Odic, A., & Kosir, A. (2016). Emotions and Personality in Personalized Services: Models, Evaluation and Applications. <https://doi.org/10.1007/978-3-319-31413-6>
- Torruella, J., & Llisterri, J. (1999). Diseño de corpus textuales y orales. *Filología e Informática. Nuevas Tecnologías En Los Estudios Filológicos*, 45–77.
- Tupes, E. C., & Christal, R. E. (1992). Recurrent personality factors based on trait ratings. *Journal of Personality*, 60(2), 225–251.
- Valstar, M., Mehu, M., Jiang, B., Pantic, M., & Scherer, K. (2012). Meta-Analysis of the First Facial Expression Recognition Challenge. *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics : A Publication of the IEEE Systems, Man, and Cybernetics Society*, 42. <https://doi.org/10.1109/TSMCB.2012.2200675>
- Wang, Z., Wu, C.-H., Li, Q.-B., Yan, B., & Zheng, K.-F. (2020). Encoding text information with graph convolutional networks for personality recognition. *Applied Sciences*, 10(12), 4081.
- Xue, D., Hong, Z., Guo, S., Gao, L., Wu, L., Zheng, J., & Zhao, N. (2017). Personality recognition on social media with label distribution learning. *IEEE Access*, 5, 13478–13488.
- Xue, D., Wu, L., Hong, Z., Guo, S., Gao, L., Wu, Z., Zhong, X., & Sun, J. (2018). Deep learning-based personality recognition from text posts of online social networks. *Applied Intelligence*, 48. <https://doi.org/10.1007/s10489-018-1212-4>
- Yu, J., & Markov, K. (2017). Deep learning based personality recognition from facebook status updates. 2017 IEEE 8th International Conference on Awareness Science and Technology (ICAST), 383–387.
- Zhang, A. (2015). Speech Recognition (Version 2.1) [Software]. Available from [https://github.com/Uberi/speech\\_recognition#readme](https://github.com/Uberi/speech_recognition#readme).

