UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

USO DE REDES NEURONALES CONVOLUCIONALES APLICADO A SENTIMENT ANALYSIS

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELÉCTRICO

RAFAEL ANDRÉS PÉREZ LÓPEZ

PROFESOR GUÍA:
JINSONG WU

MIEMBROS DE LA COMISIÓN:
PABLO MEDINA COFRE
JUAN LOZA MERCADO

SANTIAGO DE CHILE
2019

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO
POR: RAFAEL ANDRÉS PÉREZ LÓPEZ
FECHA: 2019
PROF. GUÍA: JINSONG WU

USO DE REDES NEURONALES CONVOLUCIONALES APLICADO A SENTIMENT ANALYSIS

El presente trabajo de memoria tiene por objetivo comprobar la aplicación de *Deep Learning* en el análisis de textos. La inteligencia artificial se abre paso hoy en día como la tecnología que esta revolucionando la forma en que comprendemos el mundo.

Poner esfuerzos en desarrollar maquinas independientes y autodidactas abre la posibilidad a todo un campo nuevo en el desarrollo humano.

El presente trabajo tiene como objetivo poner a prueba algoritmos de análisis de texto para calificar sentencias de los usuarios de Twitter. El *Sentiment Analysis* consiste en el primer paso para que las maquinas logren comprender, sin necesidad de un intermediario, el lenguaje humano.

Finalmente, se busca aportar con una base en cuanto análisis de texto se refiere y esto pueda derivar en aplicaciones en otros campos de las ciencias de la computación.

Dedicada a mis dos familias, especialmente a la mujer mas guerrera que conozco, Verónica.

Agradecimientos

Agradezco en primer lugar el apoyo incondicional de mi mamá, ella siempre creyó en mi potencial para lograr cualquier cosa que me propusiera. Me entrego los valores de los cuales me enorgullezco a día de hoy. Escucho todos y cada uno de mis lamentos para entregarme esa calidez que solo una madre sabe dar.

Agradezco haber nacido en una familia numerosa, haber desconocido durante toda mi infancia lo que era a el aburrimiento o la soledad. Mis compañeros de vida, mis hermanos, aquellos que siempre sabían reconfortarme con su mera presencia. En esta categoría también entra mi querido amigo Jaime, sin el cual yo no hubiera podido estar en la instancia en la que me encuentro.

Finalmente, agradecer a mi tata y mi mama, mis abuelos maravillosos. Se convirtieron en mis segundos padres y me enseñaron la belleza de la sencillez.

Tabla de Contenido

Índice de Tablas

Índice de Ilustraciones

Capítulo 1

Introducción

En este capitulo se presenta el problema abordado en este trabajo de memoria de ingeniería. Primero se explicita la **motivación** para después definir y pasar a indicar los **objetivos** generales y específicos. A continuación, se expone la **metodología de trabajo** y se mencionan los **aportes del trabajo realizado**. Finalmente, se entrega una breve descripción de la estructura de este informe.

1.1. Motivación

En esta sección se presenta el contexto en el que surgió la idea de detectar el descontento de los usuarios del sistema publico de salud.

1.1.1. Contexto General

Según un estudio realizado por la Universidad Católica Silva Henriquez a personas con un ingreso per capita inferior a 252 mil pesos, reveló que un 75 % de los encuestados considera que la atención de salud publica es "mala.º "muy mala". Además, 3 de cada 4 entrevistados admiten haber acudido a un médico particular.

El 78,7 % de los encuestados afirma que "la atención de salud es un derecho básico", y 3 de cada 4 personas atribuyen esta responsabilidad al Ministerio de Salud. Por esto, cada vez mas personas se ven forzadas a pagar consultas particulares, siendo estas las palabras del académico a cargo del estudio, Marcelo Yañez.

En paralelo, el Servicio Nacional del Consumidor (SERNAC) publicó un estudio sobre la base de reclamos y denuncias recibidas en 2012 contra las clínicas, hospitales privados, Instituciones de Salud Previsional y centros médicos.

El año 2012 se recibieron 2.787 reclamos en contra las instituciones que prestan servicios

de salud, siendo las clínicas y hospitales privados los que concentran la mayor parte de las quejas, con 43,6 %; seguido por las ISAPRE con 37,8 %; y tercer lugar los centros médicos con 18,6 %.

Los reclamos contra las Isapres sumaron 1.053 casos, siendo Consalud la que concentra el mayor porcentaje (28,3), seguida de Cruz Blanca (25,1) y Banmédica (15,8).

1.1.2. Problema a abordar

Los mecanismos actuales de fiscalización de calidad de atención resultan ineficientes y demasiado lentos en su respuesta al reclamante.

Figura 1.1: Esquema del módulo

La presente memoria busca ser un primer de intento en la solución del problema de gestión y procesamiento de reclamos, utilizando como entrada los reclamos de los usuarios del sistema de salud, tanto publico como privado, para entregar como salida un indicador del nivel de disconformidad del usuario. Este indicador se construirá a partir del estado del arte en la materia.

En otro lado, se realizará un análisis de los patrones de reclamos estudiados y comprobar la efectividad del indicador seleccionado.

1.2. Objetivos

1.2.1. Objetivos Generales

El objetivo principal de esta memoria es implementar un sistema de análisis de reclamos del sistema de salud publico y privado que entregue un indicador acorde al nivel de insatisfacción del usuario. Lo anterior se logrará por medio de *Machine Learning*, específicamente, *Sentiment Analysis*.

1.2.2. Objetivos Específicos

1. Elegir un indicador para el nivel de insatisfacción del usuario.
2. Establecer una relación entre los reclamos y las instituciones que los reciben.
3. Entregar una posible herramienta de fiscalización a los organismos pertinentes.

1.3. Hipótesis

El análisis de sentimientos de los reclamos permite identificar que centros médicos, hospitales o consultorios que están realizando un servicio deficiente y requieren ser fiscalizadas. Esto conllevará a una mejora general del servicio de salud chileno y se traducirá en una mejora en la calidad de vida de las personas.

1.4. Metodología de Trabajo

La metodología que se propone para desarrollar el sistema de análisis tiene un total de etapas, que se explican a continuación:

1. Se debe seleccionar una base de datos que contenga frases clasificadas según su polaridad; es decir, positiva o negativa.
2. Después, se debe entrenar una red neuronal con algoritmos de *Machine Learning* para determinar la polaridad de futuros reclamos.
3. Analizar el *Performance* de la red neuronal y realizar ajustes en sus parámetros a fin de obtener una clasificación mas acertada.
4. Análisis de los reclamos por organización para extraer conclusiones sobre la calidad de su atención al usuario.

1.5. Aportes del Trabajo de Memoria

El presente trabajo de memoria tiene aplicaciones en cualquier actividad que involucre un servicio determinado a un usuario, lo que permite mejorar el servicio ofrecido por la empresa u organización.

Gracias al trabajo descrito en este documento sería posible realizar una mejora sustancial en la calidad de vida de las personas mas vulnerables, aquellas que no pueden permitirse costear la salud privada y cuya única opción es el sistema publico; cuyas falencias y malas practicas son de conocimiento general.

Finalmente, la industria podría aplicar lo desarrollado para aumentar su calidad de servicio, y a su vez, su prestigio como empresa.

1.6. Estructura de la Memoria

Esta memoria presenta los siguientes capítulos:

- El Capítulo 2 contiene el Marco Teórico, con toda la base teórica para entender el trabajo realizado, y además del Estado del Arte, con los trabajos realizados en el área hasta la actualidad.
- El Capítulo 3 contiene la Metodología de Trabajo, detallando los materiales usados y los procedimientos.
- El Capítulo 4 muestra los Análisis y Resultados pertinentes.
- El Capítulo 5 enseña las Conclusiones del trabajo y propone el Trabajo Futuro.
- Por ultimo, las Referencias consultadas.

Capítulo 2

Marco Conceptual, Teórico y Estado del Arte

En este capítulo se muestran los conceptos necesarios para comprender el contexto del problema a resolver en el trabajo de memoria, junto con el estado del arte en la materia. También se muestran los fundamentos matemáticos tras la solución propuesta.

2.1. Marco Conceptual

Esta sección se exponen los conceptos básicos para entender el [**?**] trabajo de memoria. Comienza con una descripción de la **Web 2.0** y su efecto en **Internet** para terminar definiendo ***Twitter***.

2.1.1. Web 2.0

Web 2.0 es un concepto que se acuñó en 2003 y que se refiere al fenómeno social surgido a partir del desarrollo de diversas aplicaciones en Internet. El término establece una distinción entre la primera época de la Web (donde el usuario era básicamente un sujeto pasivo que recibía la información o la publicaba, sin que existieran demasiadas posibilidades para que se generara la interacción) y la revolución que supuso el auge de los blogs, las redes sociales y otras herramientas relacionadas.

La Web 2.0, por lo tanto, está formada por las plataformas para la publicación de contenidos, como Blogger, las redes sociales, como Facebook, los servicios conocidos como wikis (Wikipedia) y los portales de alojamiento de fotos, audio o vídeos (Flickr, YouTube). La esencia de estas herramientas es la posibilidad de interactuar con el resto de los usuarios o aportar contenido que enriquezca la experiencia de navegación.

Es importante tener en cuenta que no existe una definición precisa de Web 2.0, aunque

es posible aproximarse a ella estableciendo ciertos parámetros. Una página web que se limita a mostrar información y que ni siquiera se actualiza, forma parte de la generación 1.0. En cambio, cuando las páginas ofrecen un nivel considerable de interacción y se actualizan con los aportes de los usuarios, se habla de Web 2.0.

Con la llegada de la Web 2.0, se produjo un fenómeno social que cambió para siempre nuestra relación con la información, principalmente porque nos hizo parte de ella: en la actualidad, una noticia acerca de una manifestación en contra del maltrato animal no está completa sin mostrar cuántos usuarios de Facebook leyeron y disfrutaron de la misma, qué porcentaje de lectores está a favor del movimiento, y los comentarios, que muchas veces aportan datos importantes o señalan errores.

En base a las caracteristicas de Web 2.0, se pueden clasificar distintos tipo de servicios, los cuales son descritos a continuacion:

- **Blogs:** Corresponden a sitios web similares a una bitacora, en donde los usuarios generan contenido en forma de *entradas o posts* basados en contenido propio que es actualizado a menudo. Son sitios donde los usuarios pueden publicar opiniones, articulos o lo que deseen. Las entradas de los blogs se ordenan cronologicamente, y permiten la funcion de busqueda por fecha, tópicos, *keywords*, etc.
- **Wikis:** Son páginas web en las que varios autores pueden colaborar conjuntamente para editar la información y conformar un documento determinado. Además comprende herramientas que facilitan controlar las versiones y regenerar una version anterior en caso de errores. La mas conocida en el mundo es Wikipedia.
- **Foros:** Corresponden a sitios que representan escenarios de diálogos y sueln estar tematizados.
- **Redes Sociales:** Son sitios que permiten estructurar relaciones en los mas diversos ámbitos, desde profesionales hasta personales. Se genera comunicación e intercambio de información entre los usuarios. Ellos son los que generan el contenido del sitio.

2.2. Twitter

Twitter es un sitio web de *microblogging* creado en el año 2006. Tiene la principal caracteristica que los usuarios publican mensajes de texto acotados a 140 caracteres de longitud, llamados *tweets*. Los usuarios pueden suscribirse a a los tweets de otros usuarios, a lo que se llama "seguirz se les conoce como *friends*, y los usuarios suscritos se les conoce como "seguidores.º *followers*. Twitter es una red social que ha tenido un gran crecimiento, tiene 313 millones de usuarios activos y soporta mas de 40 idiomas a lo largo del mundo.

Como se mencionó, los tweets están compuestos por un máximo de 140 caracteres, por ende existen varios terminos y convenciones en los mensajes y en la red de usuarios que los permiten caracterizar.

- **Followers:** Representan a los seguidores de un usuario.
- **Following/Friends:** Corresponde a las cuentas que un usuario está siguiendo.

- **Status/ Tweets:** Son las actualizaciones de estado de los usuarios.
- **ReTweet (RT):** Consiste en publicar nuevamente un tweet, la acción se denomina *retwittear*.
- **Hashtags (simbolo):** Representan las etiquetas (son palabras escritas con el simbolo antepuesto) que usan para indexar palabras claves o temas de Twitter.
- **Usuario (simbolo @):** Se usa para indicar a un usuario dentro de un tweet. Los usuarios que se nombren en un tweet recibirán dicho mensaje.
- **Mensajes Directos/ Direct Messages (DM):** Son mensajes privados que se envían entre usuarios de Twitter. Se utilizan para mantener conversaciones privadas entre dos usuarios o un grupo de usuarios.
- **Menciones:** Corresponde a tweets en donde el mensaje contiene @nombredeusuario en el cuerpo del mensaje. El usuario mencionado recibe dicho mensaje.
- **Respuestas:** Corresponde a mensajes asociados a otros. Son similares a las menciones pero se diferencias en que @nombredeusuario se encuentra en el comienzo del tweet.

2.2.1. APIs de Twitter

Actualmente Twitter es una de las mayores fuentes de información en tiempo real de Internet, alimentada por millones de usuarios, reales y automáticos.

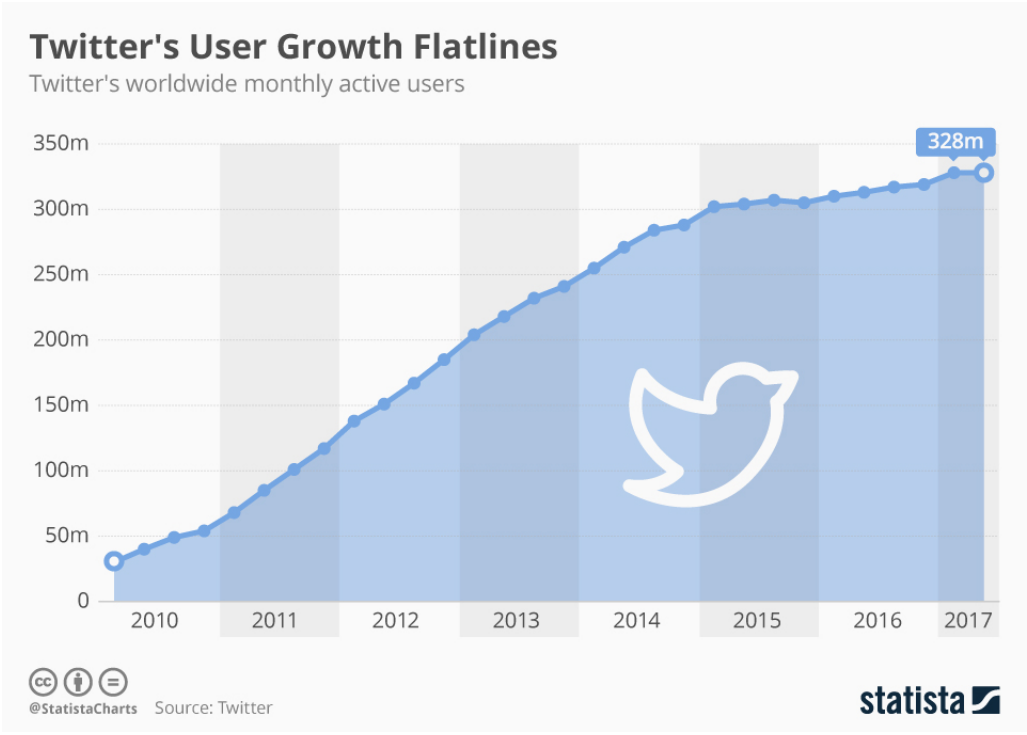

Figura 2.1: Crecimiento Twitter

El número de usuarios de Twitter tuvo un crecimiento espectacular hasta 2014, desacelerándose a partir de esa fecha. Parece que Twitter está cerca de su techo y que no consigue

atraer a nuevos usuarios. No obstante, es la plataforma de referencia para informarse en todo lo relacionado con los acontecimientos especiales. Su fortaleza radica en que la inmensa mayoría de sus mensajes son públicos lo que facilita su velocidad de propagación, llegando a muchas personas en muy poco tiempo.

Twitter ha ido variando sus APIs según sus versiones. Al principio ofrecía tres APIs: REST, Search y Streaming. Se ha reestructurado la documentación de sus APIs en un intento de simplificar (aunque creo que no está muy logrado) y de proporcionar información de los servicios estándar (gratis) o premium (de pago) que ofrece.

Todos los datos que suministran las APIs están en formato JSON. Los objetos más usados son el User object que contiene la información del perfil del un usuario y el tweet object que proporciona la información de contexto de cada tuit incluyendo el user object del autor, las entidades (hashtags, menciones a usuarios, URLs y multimedia) y la geo-localización.

- **REST API:** ofrece a los desarrolladores el acceso al core de los datos de Twitter. Todas las operaciones que se pueden hacer vía web son posibles realizarlas desde la API, como ver los perfiles de los usuarios, quienes son sus seguidores y seguidos, los tuits que publican, cuales son los trending topics, etc.
- **Search API:** suministra los tuits con una profundidad en el tiempo de 7 días que se ajustan a la query solicitada. Es posible filtrar por lenguaje y localización.
- **Streaming API:** proporciona un flujo de tuits en casi tiempo real al establecer una conexión permanente con los servidores de Twitter. Se puede obtener un filtrado (statuses/filter) por diez tipos de parámetros diferentes, siendo los más habituales palabras claves, usuarios y localizaciones. También es posible descargar una muestra aleatoria de tuits (statuses/sample).

2.3. Data Mining

El Data Mining es un conjunto de técnicas y tecnologías que permiten explorar grandes bases de datos, de manera automática o semiautomática, con el objetivo de encontrar patrones repetitivos que expliquen el comportamiento de estos datos.

A pesar de que la idea del Data Mining puede parecer una innovación tecnológica muy reciente, en realidad este término apareció en los años sesenta conjuntamente con otros conceptos como por ejemplo, el data fishing o data archeology. No obstante, no fue hasta los años ochenta cuando empezó su consolidación.

La minería de datos surgió con la intención o el objetivo de ayudar a comprender una enorme cantidad de datos, y que estos, pudieran ser utilizados para extraer conclusiones para contribuir en la mejora y crecimiento de las empresas, sobre todo, por lo que hace a las ventas o fidelización de clientes.

Su principal finalidad es explorar, mediante la utilización de distintas técnicas y tecnologías, bases de datos enormes de manera automática con el objetivo de encontrar patrones

repetitivos, tendencias o reglas que expliquen el comportamiento de los datos que se han ido recopilando con el tiempo. Estos patrones pueden encontrarse utilizando estadísticas o algoritmos de búsqueda próximos a la Inteligencia Artificial y a las redes neuronales.

Las personas que se dedican al análisis de datos a través de este sistema son conocidos como mineros o exploradores de datos, estos intentan descubrir patrones en medio de enormes cantidades de datos. Su intención es la de aportar información valiosa a las empresas para así, ayudarlas en la toma de decisiones futuras. Pero debemos tener claro que la elección del mejor algoritmo para una tarea analítica específica es un gran desafío, ya que podemos encontrar muchos patrones distintos, y además, dependerá de los problemas a resolver. Estos pueden ser la clasificación, regresión, segmentación, asociación y análisis de secuencias.

Los mineros o exploradores de datos a la hora de llevar a cabo un análisis de Data Mining, deberán realizar cuatro pasos distintos:

- **Determinación de los objetivos:** El investigador determina qué objetivos quiere conseguir gracias al uso del Data Mining.
- **Procesamiento de los datos:** Selección, limpieza, enriquecimiento, reducción y transformación de la base de datos.
- **Determinación del modelo:** Primero se debe hacer un análisis estadístico de los datos y después visualización gráfica de los mismos.
- **Análisis de los resultados:** En este paso se deberán verificar si los resultados obtenidos son coherentes.

Actualmente este tipo de trabajos se están realizando en seguridad de datos, finanzas, salud, marketing, detección de fraude, búsquedas online, procesamiento del lenguaje natural, coches inteligentes, entre otros. Es por este motivo, que la minería de datos se está convirtiendo en uno de los trabajos con mayor proyección para el futuro.

2.3.1. Procesamiento de datos

Convertir información en conocimiento tiene mucho que ver con entender las diferentes etapas de los procesos de datos. Extraer utilidad y generar valor no es un simple proceso de minería, sino que se trata, además, de una técnica especializada en la que hay que saber elegir entre una amplia gama de enfoques, herramientas y métodos.

Los datos en bruto entran al sistema y, a partir de ese momento, sólo de la organización depende el uso que se les dé. Aunque todos los procesos de datos son distintos, en todos se aprecia un esquema común: tras el acceso, se produce el despliegue a los largo de sistemas informáticos, software, aplicaciones varias... un recorrido que parte del dato y termina en la producción de conocimiento y contexto. El paso previo a la generación de valor.

Etapas del procesamiento del texto

Conocer las etapas de los procesos de datos es importante para optimizarlos, entenderlos mejor y lograr extraer todo el partido a cada bit de información que se introduce en el sistema. Así, puede hablarse de seis fases comunes a la mayoría de ellos:

- **Recogida:** Es la primera etapa del ciclo y la más relevante ya que la calidad de los datos recogidos afectará en gran medida a los productos que de ellos se obtengan. Esta etapa proporciona tanto la línea de base desde la que organizar el sistema de métricas, como la información necesaria acerca de los objetivos de los procesos de datos, que son los que ayudan a continuar mejorando.
- **Preparación:** En esta etapa se manipulan los datos, con el fin de que adquieran el formato adecuado para su posterior análisis y procesamiento. Los datos en bruto no pueden ser procesados y se debe comprobar su exactitud para evitar errores en fases posteriores.
- **Entrada de datos:** Ya se lleve a cabo de forma manual, digital o automatizada, este tipo de procesos de datos buscan convertir los datos en información procesable. Esta etapa, que consume bastante tiempo, requiere velocidad y precisión aunque se caracteriza por su intensidad en recursos.
- **Procesamiento:** Llegados a este punto, los datos se someten a diversos métodos, cada uno con sus instrucciones; a través de los que se intentan evaluar, clasificar y organizar para obtener información útil. Ésta es una de las metas de los procesos de datos y que, gracias a los avances tecnológicos, cada vez es posible conseguir en periodos de tiempo más reducidos.
- **Interpretación y análisis:** La información procesada se transmite al usuario de negocio, quien se ocupará de acceder a ella a través de informes, visualizaciones, montajes de vídeo o audio; para obtener el conocimiento que guiará las decisiones futuras de la empresa.

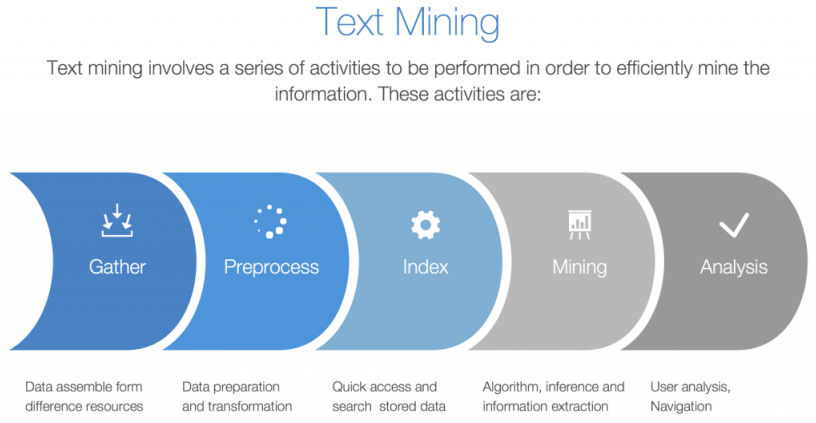

Figura 2.2: Esquema Data Mining

2.3.2. Preprocesamiento de texto

Existe una etapa en la que se debe preprocesar el texto recogido de forma que sea homogéneo y entendible para un programa en cuestión. En la siguiente figura se puede apreciar cuales son los pasos necesarios:

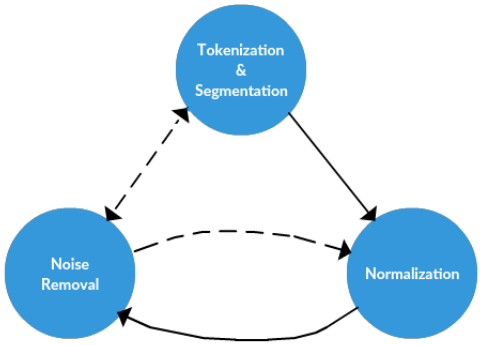

Figura 2.3: Esquema Preprocesamiento de Texto

Eliminación del ruido

Se define como la eliminación de elementos de texto que no aportan información relevante. Para el caso de este trabajo de memoria, se procederá a remover los siguientes elementos de ruido:

- **Hashtags #**
- **Menciones @**
- **Direcciones Web**
- **URLs**
- **Caracteres especiales**

Tokenización

La tokenización es un paso que divide cadenas de texto más largas en piezas más pequeñas o tokens. Los trozos de texto más grandes pueden ser convertidos en oraciones, las oraciones pueden ser tokenizadas en palabras, etc. El procesamiento adicional generalmente se realiza después de que una pieza de texto ha sido apropiadamente concatenada. La tokenización también se conoce como segmentación de texto o análisis léxico. A veces la segmentación se usa para referirse al desglose de un gran trozo de texto en partes más grandes que las palabras (por ejemplo, párrafos u oraciones), mientras que la tokenización se reserva para el proceso de desglose que se produce exclusivamente en palabras.

Normalización

La normalización generalmente se refiere a una serie de tareas relacionadas destinadas a poner todo el texto en igualdad de condiciones: convirtiendo todo el texto en el mismo caso (superior o inferior), eliminando la puntuación, convirtiendo los números a sus equivalentes de palabras, y así sucesivamente. La normalización pone todas las palabras en pie de igualdad, y permite que el procesamiento proceda de manera uniforme.

2.4. Word2Vec

Word2vec es una red neuronal que procesa texto. La entrada corresponde a un *corpus* de texto y la salida son un set de vectores: vectores de caracteristicas de las palabras presentes en el *corpus*. A pesas de que Word2vec no es una *Deep neural network*, convierte texto a números que las redes profundas pueden entender.

El propósito y utilidad de Word2vec es agrupar los vectores de palabras similares en el mismo espacio vectorial. Esto es, detectar matemáticamente similitudes en las palabras. Word2vec crea vectores que son representaciones numéricas distribuidas de las características de las palabras, que pueden ser el contexto o palabras individuales. Todo lo anterior sin intervención humana.

Con suficiente cantidad de datos, Word2vec puede hacer aproximaciones muy precisas sobre el significado de las palabras según las apariciones pasadas de las mismas.

La salida de Word2vec es un vocabulario en el cual cada objeto tiene un vector asociado, el cual puede ser introducido directamente en un red neuronal de *Deep Learning*.

Medir la similitud entre palabras representadas como vectores puede ser logrado a través la similitud del coseno. el maximo valor de similitud sería entonces 1 para palabras iguales y 0 corresponderia al minimo.

Existen dos tipos de Word2vec, Skip-gram y Continuos Bag of Words (CBOW).

Skip-Gram:

En el modelo Skip-gram, la entrada corresponde a una palabra concreta, mientras que la salida corresponden a las palabras que rodean la palabra dada. Por ejemplo, en la oración "Yo tengo un perro grande", la entrada podría ser ün", entonces la salida seria "Yo", "tengo", "perro", "grande", se asume que el tamaño de la ventana es 5. La red contiene una capa oculta cuya dimension es igual al tamaño del *embedding*, que es siempre menor al tamaño del vector de entrada/salida. El siguiente grafico da muestra de lo anterior. Con Skip-gram, la dimension de representacion decrece desde el tamaño del vocabulario (V) hasta el largo de la capa oculta (N).

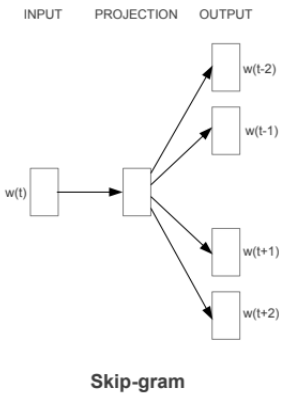

Figura 2.4: Skip-gram

CBOW

Continuos Bag of Words es muy similar a Skip-gram, exceptuando que se intercambian la salida y la entrada. La idea es, dado un contexto, entregar la palabra con mas posibilidades de aparecer.

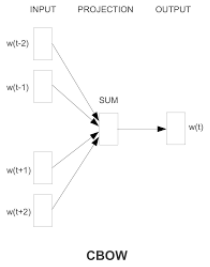

Figura 2.5: CBOW

La mayor diferencia se encuentra con Skip-gram es la forma en que los vectores son generados.

2.5. Machine Learning

Machine Learning es una disciplina científica del ámbito de la Inteligencia Artificial que crea sistemas que aprenden automáticamente. Aprender en este contexto quiere decir identificar patrones complejos en millones de datos. La máquina que realmente aprende es un algoritmo que revisa los datos y es capaz de predecir comportamientos futuros. Automáticamente, también en este contexto, implica que estos sistemas se mejoran de forma autónoma con el tiempo, sin intervención humana.

Como se establecio previamente, es un campo de las ciencias de la computación que, de acuerdo a Arthur Samuel en 1959, le da a las computadoras la habilidad de aprender sin ser

explícitamente programadas.

2.5.1. Funcionamiento

El principal objetivo de todo aprendiz (*learner*) es desarrollar la capacidad de generalizar y asociar. Cuando traducimos esto a una máquina o computadora, significa que éstas deberían poder desempeñarse con precisión y exactitud, tanto en tareas familiares, como en actividades nuevas o imprevistas.

Se logra haciendo que repliquen las facultades cognitivas del ser humano, formando modelos que "generalicen"la información que se les presenta para realizar sus predicciones. Y el ingrediente clave en toda esta cuestión son los datos.

En realidad, el origen y el formato de los datos no es tan relevante, dado que el machine learning es capaz de asimilar una amplia gama de éstos, lo que se conoce como big data, pero éste no los percibe como datos, sino como una enorme lista de ejemplos prácticos.

Podríamos decir que sus algoritmos se dividen principalmente en tres grandes categorías: *Supervised learning* (Aprendizaje supervisado), *Unsupervised learning* (Aprendizaje no supervisado) y *Reinforcement learning* (Aprendizaje por refuerzo). A continuación, detallaremos las diferencias entre éstas.

2.5.2. Tipos de Machine Learning

A continuación se presentan los tipos de algoritmos de Machine Learning desarrollados a día de hoy:

- **Supervised Learning:** Depende de datos previamente etiquetados, como podría ser el que una computadora logré distinguir imágenes de autos, de las de aviones. Para esto, lo normal es que estas etiquetas o rótulos sean colocadas por seres humanos para asegurar la efectividad y calidad de los datos.

  En otras palabras, son problemas que ya hemos resuelto, pero que seguirán surgiendo en un futuro. La idea es que las computadoras aprendan de una multitud de ejemplos, y a partir de ahí puedan hacer el resto de cálculos necesarios para que nosotros no tengamos que volver a ingresar ninguna información.

  Ejemplos: reconocimiento de voz, detección de spam, reconocimiento de escritura, entre otros.

- **Unsupervised Learning:** En esta categoría lo que sucede es que al algoritmo se le despoja de cualquier etiqueta, de modo que no cuenta con ninguna indicación previa. En cambio, se le provee de una enorme cantidad de datos con las características propias de un objeto (aspectos o partes que conforman a un avión o a un coche, por ej.), para que pueda determinar qué es, a partir de la información recopilada.

  Ejemplos: detectar morfología en oraciones, clasificar información, etc.

- **Reinforcement Learning:** En este caso particular, la base del aprendizaje es el refuerzo. La máquina es capaz de aprender con base a pruebas y errores en un número de diversas situaciones.

  Aunque conoce los resultados desde el principio, no sabe cuáles son las mejores decisiones para llegar a obtenerlos. Lo que sucede es que el algoritmo progresivamente va asociando los patrones de éxito, para repetirlos una y otra vez hasta perfeccionarlos y volverse infalible.

  Ejemplos: navegación de un vehículo en automático, toma de decisiones, etc.

Deep Learning

El Deep Learning es una forma especializada de aprendizaje automático. Un flujo de trabajo de Machine Learning empieza con la extracción manual de las características relevantes de los datos. Estas características se utilizan entonces para crear un modelo que categoriza los objetos de los datos. Con un flujo de trabajo de Deep Learning, las características relevantes se extraen directamente de los datos. Además, el aprendizaje profundo realiza un "aprendizaje completo", es decir, se proporcionan datos sin procesar y una tarea que realizar, como puede ser una clasificación, a una red, la cual aprende cómo hacerlo automáticamente.

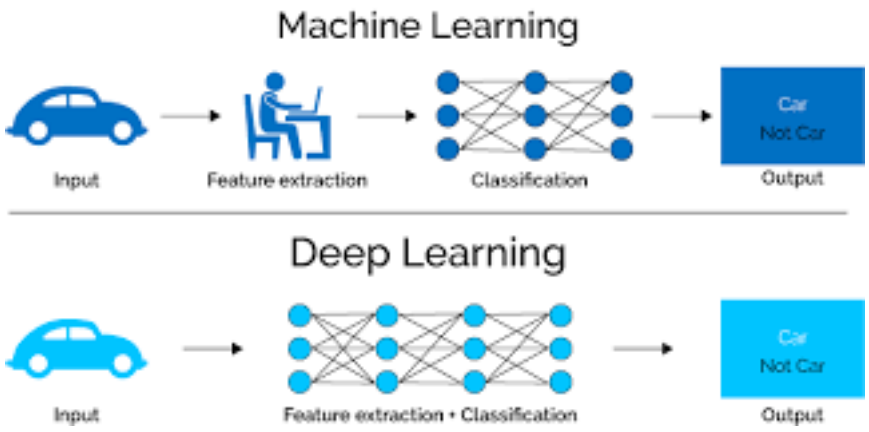

Figura 2.6: Machine Learning v/s Deep Learning

2.6. Redes neuronales

En esta sección se procede a definir lo que es una red neuronal artificial y el origen biológico de este modelo computacional. Además, se entregan el trasfondo matemático que subyace en las redes neuronales artificiales.

2.6.1. Neurobiología

Una neurona típica posee el aspecto y las partes que se muestran en la figura. Sin embargo, se debe observar que el dibujo no esta a escala, el axón alcanzo un largo normal de centímetros y a veces de varios metros, las dendritas también y las terminales sinápticas, son mas largas, numerosas y tupidas.

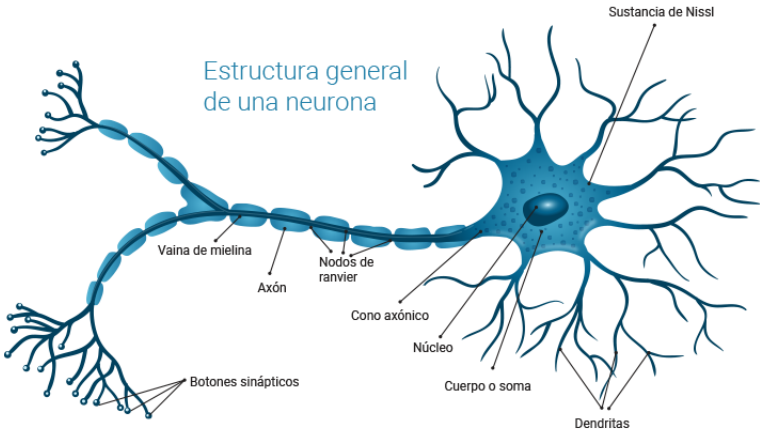

Figura 2.7: Neurona

Típicamente, las neuronas son 6 ó 5 órdenes de magnitud más lentas que una compuerta lógica de silicio, los eventos en un chip de silicio toman alrededor de nanosegundos, mientras que en una neurona este tipo de es del orden de milisegundos.

La mayoría de las neuronas codifican sus salidas como una serie de breves pulsos periódicos, llamados *potenciales de acción,* que se originan cercanos al soma de la célula y se propagan a través del axón. Luego, este pulso llega a las sinapsis y de ahí a las dendritas de las neuronas siguientes. Una *sinapsis* es una interconexión entre dos neuronas, en la parte inferior de estas lineas se incluye una figura de la misma. En ella, el botón sináptico corresponde al término del axón de una neurona pre-sináptica, y al dendrita es la correspondiente neurona post-sináptica.

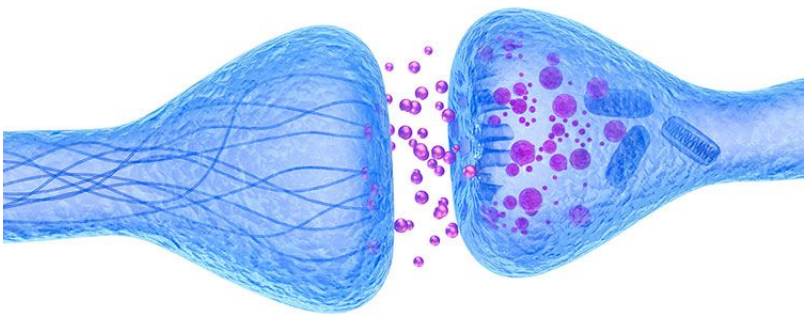

Figura 2.8: Salto Sináptico

En la neurona, hay dos comportamientos muy importantes:

1. El impulso que llega a una sinapsis y el que sale de ella no son iguales en general. El tipo de pulso que saldrá depende muy sensiblemente de la cantidad de neurotransmisor. Esta cantidad de neurotransmisor cambia durante el proceso de aprendizaje, es aquí donde se almacena la información. Una sinapsis modifica el pulso, ya sea reforzándolo o debilitándolo.
2. En el soma se suman las entradas de todas las dendritas. Si estas entradas sobrepasan un cierto umbral, entonces se transmitirá un pulso a lo largo del axón, en caso contrario no transmitirá. Después de transmitir un impulso, la neurona no puede transmitir durante un tiempo de entre 0.5 ms a 2 ms. A este tiempo se le llama período refractario.

En base a estas dos características, se construye el modelo de red neuronal artificial.

2.6.2. Modelo neuronal

En esta sección se desea introducir un modelo sencillo de la neurona para construir redes, nuestro fin ultimo es modelar correctamente el comportamiento global de toda la red.

El primer modelo matemático de una neurona artificial, creado con el fin de llevar a cabo tareas simples, fue presentado en el año 1943 en un trabajo conjunto entre el psiquiatra y neuroanatomista Warren McCulloch y el matemático Walter Pitts.

La siguiente figura muestra un ejemplo de modelo neuronal con n entradas, que consta de:

- Un conjunto de entradas $x_1, ...x_n$
- Los pesos sinápticos $w_1, ...w_n$, correspondientes a cada entrada.
- La función de agregación, Σ
- Una función de activación, f
- Una salida, Y

Las entradas son el estímulo que la neurona artificial recibe del entorno que la rodea, y la salida es la respuesta a tal estímulo. La neurona puede adaptarse al medio circundante y aprender de él modificando el valor de sus pesos sinápticos, y por ello son conocidos como los **parámetros libres** del modelo, ya que pueden ser modificados y adaptados para realizar una tarea determinada.

En este modelo, la salida neuronal Y está dada por:

$$Y = f(\sum_{\mathrm{i}=1}^{n} w_{\mathrm{i}} x_{\mathrm{i}})$$

La función de activación se elige de acuerdo a la tarea realizada por la neurona. Entre las más comunes dentro del campo de las RNAs podemos destacar:

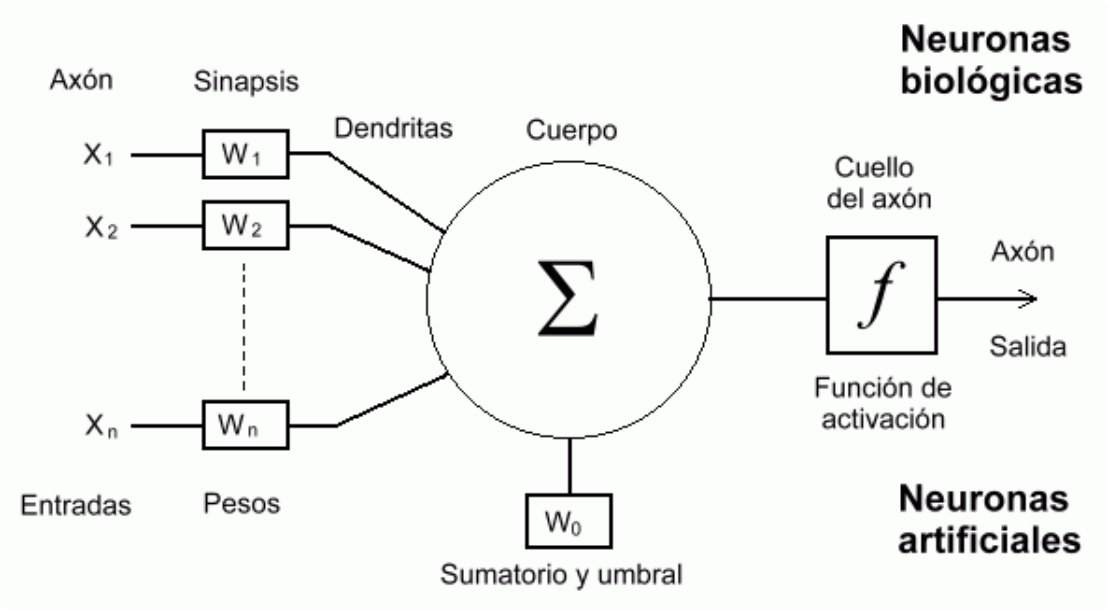

Figura 2.9: Esquema Neurona

	Función	Rango	Gráfica
Identidad	$y = x$	$[-\infty, +\infty]$	
Escalón	$y = sign(x)$ $y = H(x)$	$\{-1, +1\}$ $\{0, +1\}$	
Sigmoidea	$y = \frac{1}{1+e^{-x}}$ $y = tgh(x)$	$[0, +1]$ $[-1, +1]$	
Gaussiana	$y = Ae^{-Bx^2}$	$[0,+1]$	
Sinusoidal	$y = A\,\mathrm{sen}(\omega x + \varphi)$	$[-1,+1]$	

Figura 2.10: Funciones de activación

El Perceptrón multicapa

Un caso particular de perceptrón múltiple se puede formar organizando sus neuronas en capas. Así, tenemos la capa de entrada formada por las entradas a la red, la capa de salida formada por las neuronas que constituyen la salida final de la red, y las capas ocultas formadas por las neuronas que se encuentran entre los nodos de entrada y de salida. Una RNA puede tener varias capas ocultas o no tener ninguna de ellas. Las conexiones sinápticas (las flechas que llegan y salen de las neuronas) indican el flujo de la señal a través de la red, y tienen asociadas un peso sináptico correspondiente. Si la salida de una neurona va dirigida hacia

dos o más neuronas de la siguiente capa, cada una de estas últimas recibe la salida neta de la neurona anterior. La cantidad de capas de una RNA es la suma de las capas ocultas más la capa de salida. En el caso de existir capas ocultas nos referimos a la RNA como un Perceptron multicapa.

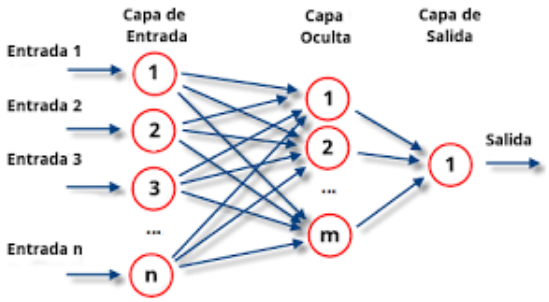

Figura 2.11: Perceptrón Multicapa

El problema habitual con este tipo de redes multicapa es el de, dados un conjunto de datos ya clasificados, de los que se conoce la salida deseada, proporcionar los pesos adecuados de la red para que se obtenga una aproximación correcta de las salidas si la red recibe únicamente los datos de entrada. A mediados de los años 80 se ofreció un algoritmo, llamado de Propagación hacia atrás, que aproxima en muchos casos los pesos a partir de los datos objetivo. Este algoritmo de entrenamiento de la red se puede resumir muy brevemente en los siguiente puntos:

- Empezar con unos pesos sinápticos cualesquiera (generalmente elegidos al azar).
- Introducir datos de entrada (en la capa de entrada) elegidos al azar entre el conjunto de datos de entrada que se van a usar para el entrenamiento.
- Dejar que la red genere un vector de datos de salida (propagación hacia delante).
- Comparar la salida generada por al red con la salida deseada.
- La diferencia obtenida entre la salida generada y la deseada (denominada error) se usa para ajustar los pesos sinápticos de las neuronas de la capa de salidas.
- El error se propaga hacia atrás (back-propagation), hacia la capa de neuronas anterior, y se usa para ajustar los pesos sinápticos en esta capa.
- Se continua propagando el error hacia atrás y ajustando los pesos hasta que se alcance la capa de entradas.
- Este proceso se repetirá con los diferentes datos de entrenamiento.

Empezar con unos pesos sinápticos cualesquiera (generalmente elegidos al azar). Introducir datos de entrada (en la capa de entrada) elegidos al azar entre el conjunto de datos de entrada que se van a usar para el entrenamiento. Dejar que la red genere un vector de datos de salida (propagación hacia delante). Comparar la salida generada por al red con la salida deseada. La diferencia obtenida entre la salida generada y la deseada (denominada error) se usa para ajustar los pesos sinápticos de las neuronas de la capa de salidas. El error se propaga hacia atrás (back-propagation), hacia la capa de neuronas anterior, y se usa para ajustar los pesos sinápticos en esta capa. Se continua propagando el error hacia atrás y ajustando los pesos hasta que se alcance la capa de entradas. Este proceso se repetirá con los diferentes datos de entrenamiento.

2.6.3. Redes Neuronales Convolucionales

Una red neuronal convolucional (*Convolutional Neural Networks* en inglés, con los acrónimos CNNs o ConvNets) es un caso concreto de redes neuronales Deep Learning, que fueron ya usadas a finales de los 90 pero que en estos últimos años se han popularizado enormemente al conseguir resultados muy impresionantes en el procesamiento de texto, impactando profundamente en el área de visión por computador.

Las redes neuronales convolucionales son muy similares a las redes neuronales del capítulo anterior: están formadas por neuronas que tienen parámetros en forma de pesos y sesgos que se pueden aprender. Pero un rasgo diferencial de las CNN es que hacen la suposición explícita de que las entradas son texto, cosa que nos permite codificar ciertas propiedades en la arquitectura para reconocer elementos concretos en las oraciones.

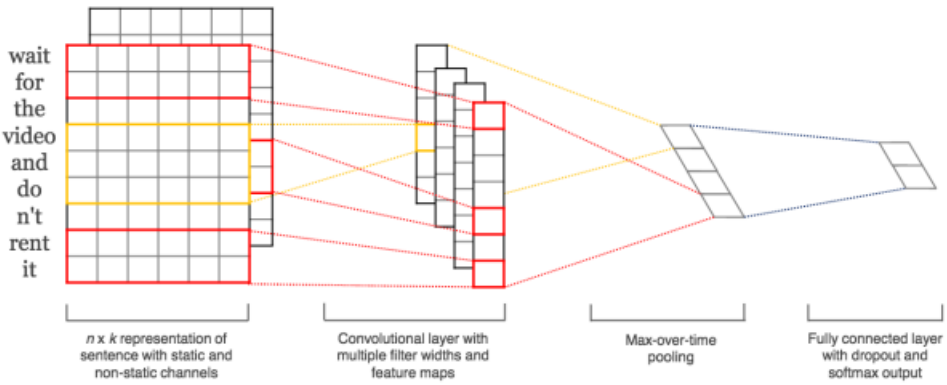

Figura 2.12: CNN para clasificación de texto

Convolución

La diferencia fundamental entre una capa densamente conectada y una capa especializada en la operación de convolución, que llamaremos capa convolucional, es que la capa densa aprende patrones globales en su espacio global de entrada, mientras que las capas convolucionales aprenden patrones locales en pequeñas ventanas de dos dimensiones.

De manera intuitiva, podríamos decir que el propósito principal de una capa convolucional es detectar características o rasgos sintácticos en las oraciones como distancia entre palabras, nombres, etc. Esta es una propiedad muy interesante, porque una vez aprendida una característica en un lugar concreto de la oración la puede reconocer después en cualquier parte de la misma. En cambio, en una red neuronal densamente conectada tiene que aprender el patrón nuevamente si este aparece en una nueva localización de la oracion.

Otra característica importante es que las capas convolucionales pueden aprender jerarquías espaciales de patrones preservando relaciones espaciales. Por ejemplo, una primera capa convolucional puede aprender elementos básicos como verbos, y una segunda capa convolucional puede aprender patrones compuestos de elementos básicos aprendidos en la capa anterior. Y

así sucesivamente hasta ir aprendiendo patrones muy complejos. Esto permite que las redes neuronales convolucionales aprendan eficientemente conceptos sintácticos y semánticos cada vez más complejos y abstractos.

Lo explicado, visualmente, se podría representar como:

Figura 2.13: Reduccion de *features* o caracteristicas

Pooling

Además de las capas convolucionales que acabamos de describir, las redes neuronales convolucionales acompañan a la capa de convolución con unas capas de *pooling*, que suelen ser aplicadas inmediatamente después de las capas convolucionales. Una primera aproximación para entender para qué sirven estas capas es ver que las capas de *pooling* hacen una simplificación de la información recogida por la capa convolucional y crean una versión condensada de la información contenida en estas.

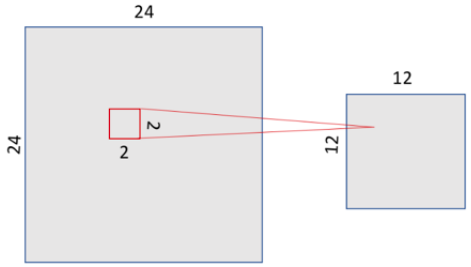

Figura 2.14: Ejemplo *Pooling*

Hay varias maneras de condensar la información, pero una habitual, y que usaremos en nuestro ejemplo, es la conocida como *maxpooling*, que como valor se queda con el valor máximo de los que había en la ventana de entrada de 22 en este caso. En este caso dividimos por 4 el tamaño de la salida de la capa de *pooling*, quedando una imagen de 1212.

Es interesante remarcar que con la transformación de *pooling* mantenemos la relación espacial. Para verlo visualmente, se toma el siguiente ejemplo de una matriz de 1212 donde se tiene representado un "7" (Se imagina que los píxeles donde pasamos por encima contienen

un 1 y el resto 0; no lo hemos añadido al dibujo para simplificarlo). Si aplicamos una operación de *maxpooling* con una ventada de 22 (que se representa en la matriz central que divide el espacio en un mosaico con regiones del tamaño de la ventana), obtenemos una matriz de 66 donde se mantiene una representación equivalente del 7 (en la figura de la derecha donde aquí se ha marcado en blanco los ceros y en negro los puntos con valor 1):

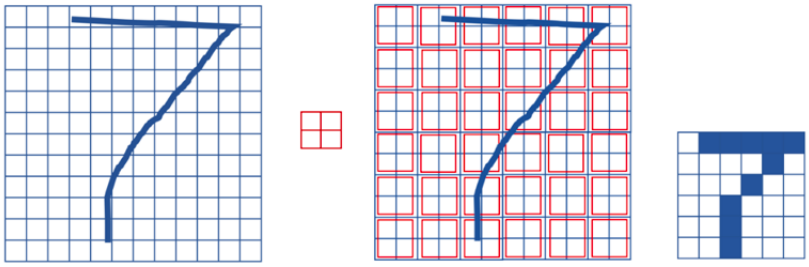

Figura 2.15: Ejemplo *Pooling* con numero 7

Capítulo 3

Materiales y Métodos

3.1. Materiales

Los materiales utilizados corresponden a la base de datos, proporcionada por la Universidad de Stanford, y los distintos *softwares* y bibliotecas utilizados en todo el desarrollo del trabajo de memoria.

3.1.1. Base de datos

Con el objetivo de entrenar el modelo de una forma óptima, es necesario un set de datos lo suficientemente amplio para que el sistema tenga a sus disposición la suficiente cantidad de ejemplos para alcanzar la mayor precisión posible.

La base de datos utilizada para este trabajo de memoria es *Sentiment140*. Este set de datos perteneciente a la Universidad de Stanford que consta de 1.6 millones de tweets clasificados según su polaridad:

- **Positivo:** Aquellos tweets cuyo mensaje tenga una connotación positiva estas marcados con polaridad=4
- **Neutral:** En este caso, cuando el mensaje no puede ser clasificado de forma ni positiva ni negativa es marcado con polaridad=2
- **Negativo:** Finalmente, aquellos tweets con un mensaje negativo son marcados con polaridad=0

3.2. Software

El lenguaje escogido para este trabajo de memoria es *Python* por su gran versatilidad y porque tiene a su disposición librerías muy útiles para el análisis de lenguaje natural como

las que se muestran a continuación:

- Libreria *scikit-learn*
- Libreria *pandas*
- Libreria *Keras*
- Libreria *Tensorflow*

3.3. Procedimientos

A continuación se muestran los procedimientos seguidos para realizar este trabajo de memoria. Bajo estas lineas, se detallan la serie de procedimientos necesarios para completar el sistema:

1. Elección del mejor procesador de lenguaje natural.
2. Preparación de los datos.
3. Construcción de la red neuronal convolucional.
4. Ajuste de parámetros para mejorar el *performance* de la red neuronal.
5. Escoger la métrica de evaluación del modelo.

3.3.1. Procesador de lenguaje natural

Actualmente, Python es uno de los lenguajes mas populares para trabajar en el campo de la Inteligencia Artificial. Para abordar los problemas relacionados con el Procesamiento de Lenguaje Natural; Python nos propociona la siguientes librerias:

- **NTLK:** Es la librería líder para el Procesamiento del Lenguaje Natural. Proporciona interfaces fáciles de usar a más de 50 corpus y recursos léxicos, junto con un conjunto de bibliotecas de procesamiento de texto para la clasificación, tokenización, el etiquetado, el análisis y el razonamiento semántico.
- **TextBlob:** TextBlob simplifica el procesamiento de texto proporcionando una interfaz intuitiva a NLTK. Posee una suave curva de aprendizaje al mismo tiempo que cuenta con una sorprendente cantidad de funcionalidades.
- **Stanford CoreNLP:** Paquete desarrollado por la universidad de Stanford, para muchos constituye el estado del arte sobre las técnicas tradicionales de Procesamiento del Lenguaje Natural. Si bien esta escrita en Java, posee una interface con Python.
- **Spacy:** Es una librería relativamente nueva que sobresale por su facilidad de uso y su velocidad a la hora de realizar el procesamiento de texto.
- **Textacy:** Esta es una librería de alto nivel diseñada sobre Spacy con la idea de facilitar aun más las tareas relacionadas con el Procesamiento del Lenguaje Natural.
- **Gensim:** Es una librería diseñada para extraer automáticamente los temas semánticos de los documentos de la forma más eficiente y con menos complicaciones posible.

- **pyLDavis:** Esta librería está diseñado para ayudar a los usuarios a interpretar los temas que surgen de un análisis de tópicos. Nos permite visualizar en forma muy sencilla cada uno de los temas incluidos en el texto.

Deep Learning y Procesamiento del Lenguaje

Durante mucho tiempo, las técnicas principales de Procesamiento del Lenguaje Natural fueron dominadas por métodos de aprendizaje automático que utilizaron modelos lineales como las máquinas de vectores de soporte o la regresión logística, entrenados sobre vectores de características de muy alta dimensional pero muy escasos. Recientemente, el campo ha tenido cierto éxito en el cambio hacia modelos de deep learning sobre entradas más densas.

Las redes neuronales proporcionan una poderosa maquina de aprendizaje que es muy atractiva para su uso en problemas de lenguaje natural. Un componente importante en las redes neuronales para el lenguaje es el uso de una capa de word embedding, una asignación de símbolos discretos a vectores continuos en un espacio dimensional relativamente bajo. Cuando se utiliza word embedding, se transforman los distintos símbolos en objetos matemáticos sobre los que se pueden realizar operaciones. En particular, la distancia entre vectores puede equipararse a la distancia entre palabras, facilitando la generalización del comportamiento de una palabra sobre otra. Esta representación de palabras como vectores es aprendida por la red como parte del proceso de entrenamiento. Subiendo en la jerarquía, la red también aprende a combinar los vectores de palabras de una manera que es útil para la predicción. Esta capacidad alivia en cierta medida los problemas de dispersión de los datos. Hay dos tipos principales de arquitecturas de redes neuronales que resultan muy útiles en los problemas de Procesamiento del Lenguaje Natural: las Redes neuronales prealimentadas y las Redes neuronales recurrentes.

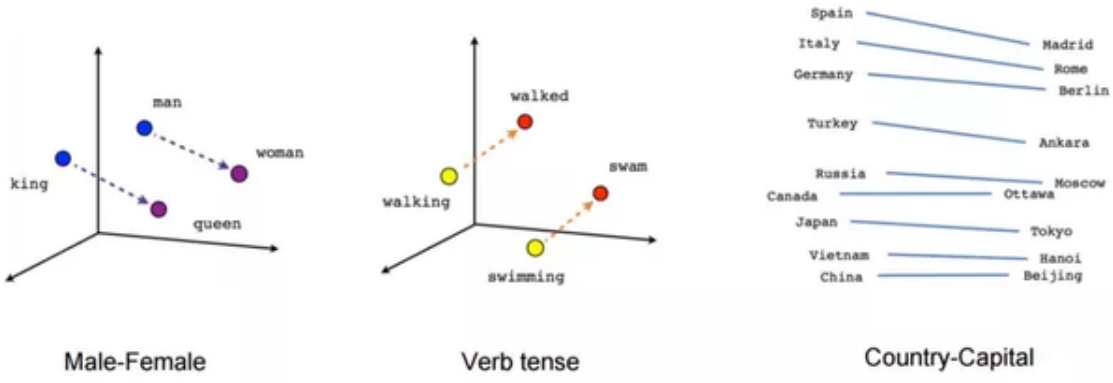

Figura 3.1: Ejemplo Word Embedding

Gensim cuenta con herramientas de *embeddding* muy purificadas y se adapta perfectamente al trabajo que se realizará en este trabajo de memoria.

3.3.2. Preparación de los datos

En esta sección se especifican todos los pasos a seguir para preparar adecuadamente los datos y entregarlos como entrada a la CNN.

1. **HTML:** El codigo HTML presenta problemas en la conversión a texto y termina como '&amp', '&Quot', etc. Se debe decodificar el codigo HTML como primer paso en el proceso de limpieza de los datos. Se usará la librería BeautifulSoup de Python para esta tarea.
2. **@menciones:** El siguiente paso de la preparación de los datos es lidiar con las @menciones. A pesar que las @menciones traen consigo cierta información, esta no añade valor para el modelo de Sentiment analysis.
3. **Links URL:** El tercer paso de la limpieza es tratar con los links URL. Estos tienen algo de información, pero para este trabajo de memoria será ignorado.
4. **UTF-8:** .[El] UTF-8 BOM es una secuencia de bytes que permite al lector identificar un archivo como UTF-8". Decodificando el texto con unicode se sustituyen los caracteres irreconocibles por ¿", que puede ser procesado y eliminado.
5. **Hashtags/Numeros:** Algunas veces el texto usado en los hashtags provee de información muy útil con respecto a tweet. Es por ello que se dejará intacto y solo se eliminara el caracter #. Además, se eliminaran todos los caracteres que no correspondan a letras, incluidos números.
6. **Guardado de los datos:** Una vez definida la función de limpieza; cada tweet sera guardado en un archivo csv en el que solo conservará su ID, el texto limpiado y la polaridad del tweet.

3.3.3. Vectorización del Corpus

Para poder alimentar la red neuronal convolucional, se deben realizar operaciones previas en corpus del documento con el objetivo de que la entrada de la red neuronal sea un conjunto de datos entendibles para la misma.

Tokenizer

La líbreria Keras cuenta con una funcion llamada *Tokenizer* que se encarga de separar cada palabra de una oración.

Text to sequences

Una vez obtenidos los tokens, el siguinte paso es obtener una representacion numérica de cada uno de los tokens generados en la sección anterior. Esto es posible gracias a función "Text_to_sequences"; esta función también puede recibir como argumento el numero má-

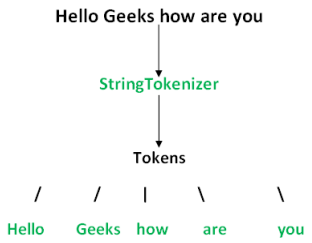

Figura 3.2: Tokenización

ximo de palabras del vocabulario que se quieren usar, para este trabajo de memoria sera 10000.

Padding

Cada tweet posee un numero diferente de palabras, o lo que es lo mismo, distinto largo. Por esta razón, se debe realizar una normalización del largo. El largo maximo de los tweets se define en 45, usando la funcion 'pad_sequences' que rellena con 0 el inicio del tweet hasta llegar a un largo de 45 para el array.

Capa de *Embedding*

El ultimo procedimiento que debe llevarse a cabo es crear una matriz de embedding con los vectores asociados a cada palabras. Para este trabajo de memoria, se decidio alimentar una red neuronal con los vectores predifinidos y entrenar un modelo para refinar la representación del corpus. Por medio de la función 'Sequential()' de Keras se define una Red neuroanal con los siguientes parametros:

- Tamaño del vocabulario: 100000. Corresponde a numero de palabras del vocabulario.
- Dimensión de los vectores: 200. Corresponde a la dimensión de los vectores de palabras.
- Los pesos asociados corresponde a la matriz d elos vectores predifinidos.
- Largo de la entrada: 45.

3.3.4. Construcción CNN

Se construye la CNN con un capa convolucional y una capa de pooling que estará completamente conectada a una capa oculta, para después conectarse a la capa de salida y realizar finalmente la predicción.

Capa convolucional

La capa convolucional que se implementará tiene las siguientes características:

- Filtros:100
- Tamaño del kernel o ventana de convolución: 2
- Se activará la función padding.

3.3.5. Métricas de rendimiento

Medir de forma matemática el rendimiento de la CNN es de suma importancia si se quiere entender como mejorar la predicción obtenidas. Se elige como metrica la curva ROC porque da una buena aproximación de nivel de acierto en la prediccion de sentimientos.

Capítulo 4

Análisis y Resultados

En este capítulo se presentan los resultados obtenidos en el trabajo de memoria por medio de la metodología expuesta en el capitulo 3. Además, se realiza un análisis de los resultados conseguidos y se realiza la correlación entre las variables comprometidas.

4.1. Visualización de la base de datos

Para poder trabajar la base de datos *Sentiment140*, que se encuentra en formato csv, se debe usar la libreria *Pandas* que cuenta con metodos para leer archivos csv.

Utilizando la función Pd.read_csv se crea un objeto pandas con ciertos atributos y funciones asociadas. A continuación, para comprobar que los datos fueron traspasados correctamente al objeto pandas se utiliza otra función de pandas llamada pd.head() con parámetro igual a 10 para ver las primeras 10 primeras entradas del objeto pandas.

	sentimiento	id	fecha	query_string	usuario	texto
0	0	1467810369	Mon Apr 06 22:19:45 PDT 2009	NO_QUERY	_TheSpecialOne_	@switchfoot http://twitpic.com/2y1zl - Awww, t...
1	0	1467810672	Mon Apr 06 22:19:49 PDT 2009	NO_QUERY	scotthamilton	is upset that he can't update his Facebook by ...
2	0	1467810917	Mon Apr 06 22:19:53 PDT 2009	NO_QUERY	mattycus	@Kenichan I dived many times for the ball. Man...
3	0	1467811184	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	ElleCTF	my whole body feels itchy and like its on fire
4	0	1467811193	Mon Apr 06 22:19:57 PDT 2009	NO_QUERY	Karoli	@nationwideclass no, it's not behaving at all....
5	0	1467811372	Mon Apr 06 22:20:00 PDT 2009	NO_QUERY	joy_wolf	@Kwesidei not the whole crew
6	0	1467811592	Mon Apr 06 22:20:03 PDT 2009	NO_QUERY	mybirch	Need a hug
7	0	1467811594	Mon Apr 06 22:20:03 PDT 2009	NO_QUERY	coZZ	@LOLTrish hey long time no see! Yes.. Rains a...
8	0	1467811795	Mon Apr 06 22:20:05 PDT 2009	NO_QUERY	2Hood4Hollywood	@Tatiana_K nope they didn't have it
9	0	1467812025	Mon Apr 06 22:20:09 PDT 2009	NO_QUERY	mimismo	@twittera que me muera ?

Figura 4.1: 10 primeras entradas *Sentiment140*

El sentimiento de cada tweet se indica en la columna "sentiemientoz puede corresponder a 3 valores distintos que son:

- 0=Negativo
- 2=Neutral

- 4=Positivo

El siguiente paso es comprobar que el objeto pandas y la base de datos tienen el mismo numero de entradas, es decir, el objeto contiene toda la información proporcionado por la base de datos.

```
4    800000
0    800000
Name: sentimiento, dtype: int64
```

Figura 4.2: Información sobre la cantidad de entradas que contiene el objeto pandas

Como se puede apreciar en la imagen superior, no existen entradas con sentimiento neutral, aún cuando el set de datos lo anuncia en su leyenda.

Es necesario descartar ciertas columnas que no aportan información relevante para el desarrollo de este trabajo de memoria; estas columnas corresponden a:

- Fecha
- Query
- Usuario
- Id

4.1.1. Limpieza de datos

Una vez cargados los datos en un objeto pandas, el siguiente paso es limpiar el texto del tweet de información irrelevante como la descrita en la sección de metodología. Para este propósito se definió una función encargada de remover las partes del tweet descritas anteriormente. Los resultados obtenidos se muestran a continuación:

	sentimiento	texto
0	0	@switchfoot http://twitpic.com/2y1zl - Awww, t...
1	0	is upset that he can't update his Facebook by ...
2	0	@Kenichan I dived many times for the ball. Man...
3	0	my whole body feels itchy and like its on fire
4	0	@nationwideclass no, it's not behaving at all....
5	0	@Kwesidei not the whole crew
6	0	Need a hug
7	0	@LOLTrish hey long time no see! Yes.. Rains a...
8	0	@Tatiana_K nope they didn't have it
9	0	@twittera que me muera ?

Figura 4.3: Tweets negativos

	sentimiento	texto
800000	4	I LOVE @Health4UandPets u guys r the best!!
800001	4	im meeting up with one of my besties tonight! ...
800002	4	@DaRealSunisaKim Thanks for the Twitter add, S...
800003	4	Being sick can be really cheap when it hurts t...
800004	4	@LovesBrooklyn2 he has that effect on everyone
800005	4	@ProductOfFear You can tell him that I just bu...
800006	4	@r_keith_hill Thans for your response. Ihad al...
800007	4	@KeepinUpWKris I am so jealous, hope you had a...
800008	4	@tommcfly ah, congrats mr fletcher for finally...
800009	4	@e4VoIP I RESPONDED Stupid cat is helping me ...

Figura 4.4: Tweets positivos

	text	target
0	awww that bummer you shoulda got david carr of...	0
1	is upset that he can not update his facebook b...	0
2	dived many times for the ball managed to save ...	0
3	my whole body feels itchy and like its on fire	0
4	no it not behaving at all mad why am here beca...	0
5	not the whole crew	0
6	need hug	0
7	hey long time no see yes rains bit only bit lo...	0
8	nope they did not have it	0
9	que me muera	0

Figura 4.5: Entradas con texto limpio

4.1.2. Wordcloud

Aunque es una herramienta muy discutida en el ámbito de visualizacion de datos, Wordcloud es una poderosa herramienta cuando se quiere determinar de forma rápida y sencilla cuales son las palabras mas frecuentes en un corpus dado.

Wordcloud contabiliza la frecuencia de cada palabra en el corpus y despliega aquellas que tienen mayor frecuencia en un arreglo aleatorio. A continuación, se muestran los resultados obtenidos para el corpus separado según polaridad del tweet, es decir, se muestran dos Wordclouds; una para los tweets positivos y otra para los negativos.

Tweets Negativos

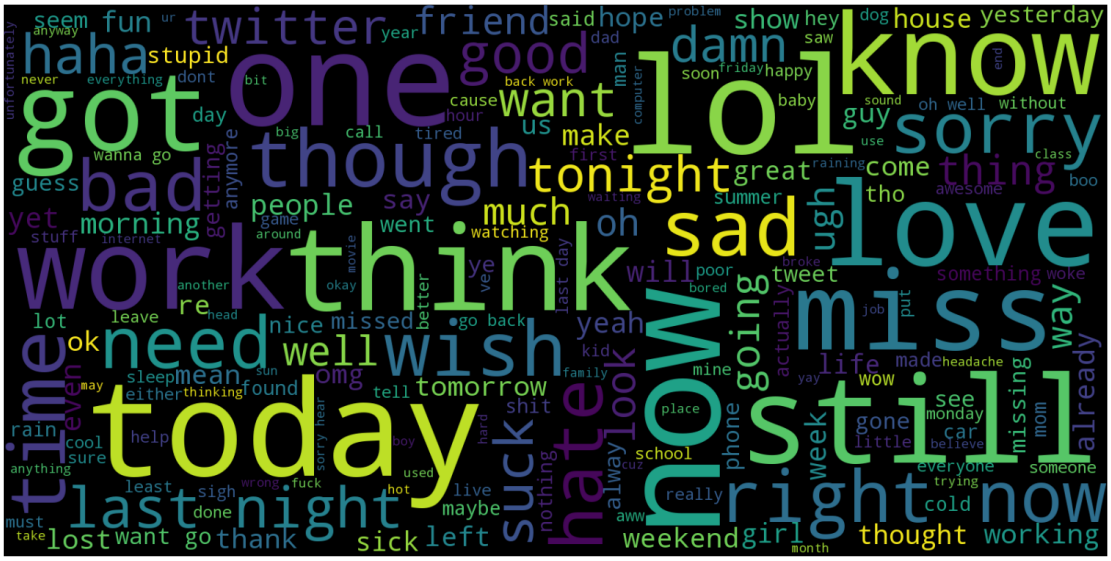

Figura 4.6: Wordcloud tweets negativos

Lo primero que llama la atención es que palabras *love* aparezcan en la wordcloud de

los tweets negativos; así tambien palabras que a primera vista se podrian catalogar como neutras, ejemplo *think*, *now* y *today*. Para determinar la razón por la cual estas palabras hacen aparición en el grafico, se realiza un *print* de aquellos tweets que contengan la palabra *love*:

```
ahh ive always wanted to see rent love the soundtrack
meh almost lover is the exception this track gets me depressed every time
awe love you too am here miss you
damn the grind is inspirational and saddening at the same time do not want you to stop cuz like what do much love
missing you babe but as long as your alive happy ya tired my love imma try to sleep hopefully you had headstart
love the french tell people here in the south qtr french and they snarl at me french are beautiful people
```

Figura 4.7: Tweets negativos que contienen la palabra *love*

Como se puede apreciar, aunque los tweets contengan la palabra *love*, el significado es de un sentimiento claramente negativo. Otras veces puede ser usado de forma sarcastica.

Tweets Positivos

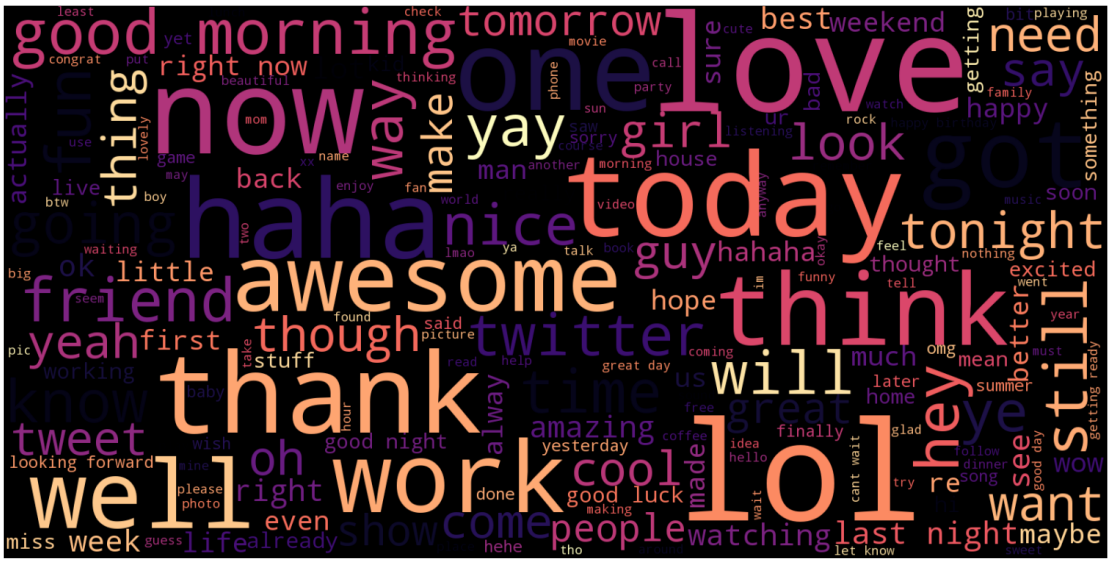

Figura 4.8: Wordcloud tweets positivos

Se puede observar un fenómeno parecido al visto en la nube de los tweets negativos; la aparición de palabras neutrales en la nube.

4.2. Construcción de Vocabulario

En esta sección se muestran los resultados obtenidos al construir un vocabulario usando Word2vec. El procedimiento consiste en alimentar una red neuronal con los tweets presentes en la base de datos, con sus correspondientes *labels*, para que esta sea capaz de inferir relaciones semánticas entre las palabras presentes en el dataset.

A continuación, se construyen dos modelos distintos gracias a la libreria gensim: Skip-gram y Continuos Bag of Words. Las diferencias entre ambos modelos se encuentran disponibles en el marco teórico de este trabajo de memoria.

Se divide la base de datos en 3 partes:

- **Set de entrenamiento:** Esta división consta de 1.564.120 tweets; de los cuales el 50.02 % son negativos y el 49.98 % positivos.
- **Set de Validación:** Este set contiene 15.960 entradas; con 49.45 % negativos y 50.55 % positivos
- **Set de Testeo:** Finalmente, el ultimo set consta de 15.961 entradas; 49.68 % negativas y 50.32 % positivas.

Se define el numero de *epochs* en 30. Con el objetivo de comprobar el nexo semántico que cada modelo ha logrado establecer entre cada una de las palabras del set de datos, se elige la palabra *"friend"* y se le exige a cada modelo que entregue las 5 palabras mas similares a la palabra seleccionada.

Como se puede observar en las imágenes bajo estas líneas, ambos modelos concuerdan al decir que *cousin* es la palabra más similar al *friend*. Sin embargo, se puede apreciar que a partir de ahí, los resultados difieren notablemente. Esto es debido a la forma que tiene cada modelo de analizar la oración y el método que utiliza cada uno para predecir las palabras en función de una entrada dada.

```
[('cousin', 0.7497243285179138),
 ('bestiest', 0.7204349040985107),
 ('bestfriend', 0.697901725769043),
 ('friends', 0.6908015012741089),
 ('aunt', 0.6563115119934082)]
```

Figura 4.9: Resultados Skip-Gram

```
[('cousin', 0.7472054958343506),
 ('friends', 0.7284735441207886),
 ('bff', 0.7195557355880737),
 ('girlfriend', 0.7182607054710388),
 ('gf', 0.704704761505127)]
```

Figura 4.10: Resultados Continuos Bag of Words

4.3. Capa *Embedding*

Un paso previo que debe realizarse antes de alimentar la red neuronal, es convertir el texto a un formato numérico que tenga sentido para la red. Para esta tarea se utiliza la clase *Tokenizer* de Keras que cuenta con varios métodos que permiten la conversión de texto a un formato numérico.

```
love eryn your melodies note selections were very very dope the song is hot as well wish had the full version
playing couple of sit gos on ps because still wanna be poker player thus no wsop for me
```

Figura 4.11: Ejemplo 2 tweets del dataset

En las imágenes superiores puede verse un ejemplo de 2 tweets en dos formatos distintos: LA primera de ellas representa dos tweets sin procesar, mientras que la segunda imagen corresponde a los mismos tweets pero convertidos en secuencias numéricas donde cada numero corresponde al lugar en el cual fue vista por primera vez la palabra en cuestión.

```
[[42,
  29626,
  39,
  22261,
  1102,
  16332,
  143,
  113,
  113,
  3349,
  2,
  262,
  8,
  237,
  77,
  68,
  106,
  57,
  2,
  443,
  975],
 [355,
  601,
  11,
  978,
  24864,
  12,
  854,
  197,
  66,
  161,
  21,
  2522,
  1733,
  3796,
  35,
  10859,
  10,
  14]]
```

Figura 4.12: Ejemplo 2 tweets convertidos a sequences

4.3.1. *Padding*

Se determina por medio de la función *Lenght* que el numero máximo de palabras presentes en un tweet corresponde a 40. Se decide fijar una extensión máxima de 45 palabras para todos los tweets.

Gracias a la función *pad_sequences* logramos esto ultimo y cada tweet se ven como el siguiente ejemplo:

```
array([    0,     0,     0,     0,     0,     0,     0,     0,     0,
           0,     0,     0,     0,     0,     0,     0,     0,     0,
           0,     0,     0,     0,     0,     0,    42, 29626,    39,
       22261,  1102, 16332,   143,   113,   113,  3349,     2,   262,
           8,   237,    77,    68,   106,    57,     2,   443,   975],
      dtype=int32)
```

Figura 4.13: Tweet con *padding*

4.3.2. Matriz de *Embedding*

Finalmente, el ultimo paso consiste en combinar los vectores del vocabulario de cada modelo: Skip-Gram y CBOW. Esta matriz tiene una dimensión de 200X45 (Cada modelo genera un vector de 100X1 para cada palabra, por lo tanto, al combinar los vectores de los dos modelos se obtienen vectores de 200X1; mientras que 45 corresponde a la extensión máxima definida).

4.4. Implementación de la CNN

El modelo final de CNN implementado en este trabajo de memoria consta de los siguientes elementos:

- Una entrada en formato secuencial con una dimensión de 45X1
- Una capa de *embedding* formada por la matriz de *embedding*.
- Una capa de bigramas con su respectiva capa de convolución y *pooling*. La funcion de activacion es *relu"*
- Una capa de Trigramas con las mismas características que la de bigramas.
- Otra capa de Cuatrigramas similar a las anteriores.
- Una capa oculta con 256 neuronas con función de activacion *relu"*
- Una capa de salida con 1 neurona. Su función de activación es *"sigmoid"*

La métrica seleccionada es *accuracy* y el optimizador *adam*.

4.5. Resultados obtenidos

Después de entregarle los parámetros a la CNN, se inicia su entrenamiento por medio de 5 epocas en las cuales la red neuronal intenta mejorar su rendimiento a través de prueba y error. Cuando se logra un mejor desempeño, se deben guardar los pesos que asociados a la capa de *embedding* y también los de la capa oculta. Los resultados finales se muestran en la siguiente tabla:

Epoch	Tiempo	Loss	Accuracy	Val_loss	Val_acc
1	1552	0.4033	0.8182	0.3839	0.8302
2	1561	0.3653	0.8401	0.3798	0.8345
3	1560	0.3383	0.8544	0.3817	0.8345
4	1559	0.3090	0.8691	0.3967	0.8318
5	1560	0.2782	0.8838	0.4275	0.8256

Tabla 4.1: Resultados entrenamiento CNN

- Tiempo medido en segundos.

- Acc, Loss corresponden a porcentajes sobre el total de los datos de entrenamiento.
- Val_acc y Val_loss son porcentajes pero referidos al set de validación.

Se puede inferir de a tabla que mejor valor de accuracy alcanzado se logra en la quinta época, con un valor igual al 0.8838 %; mientras que por otro lado el valor de Loss es el mas bajo alcanzado e igual a 0.2782 %.

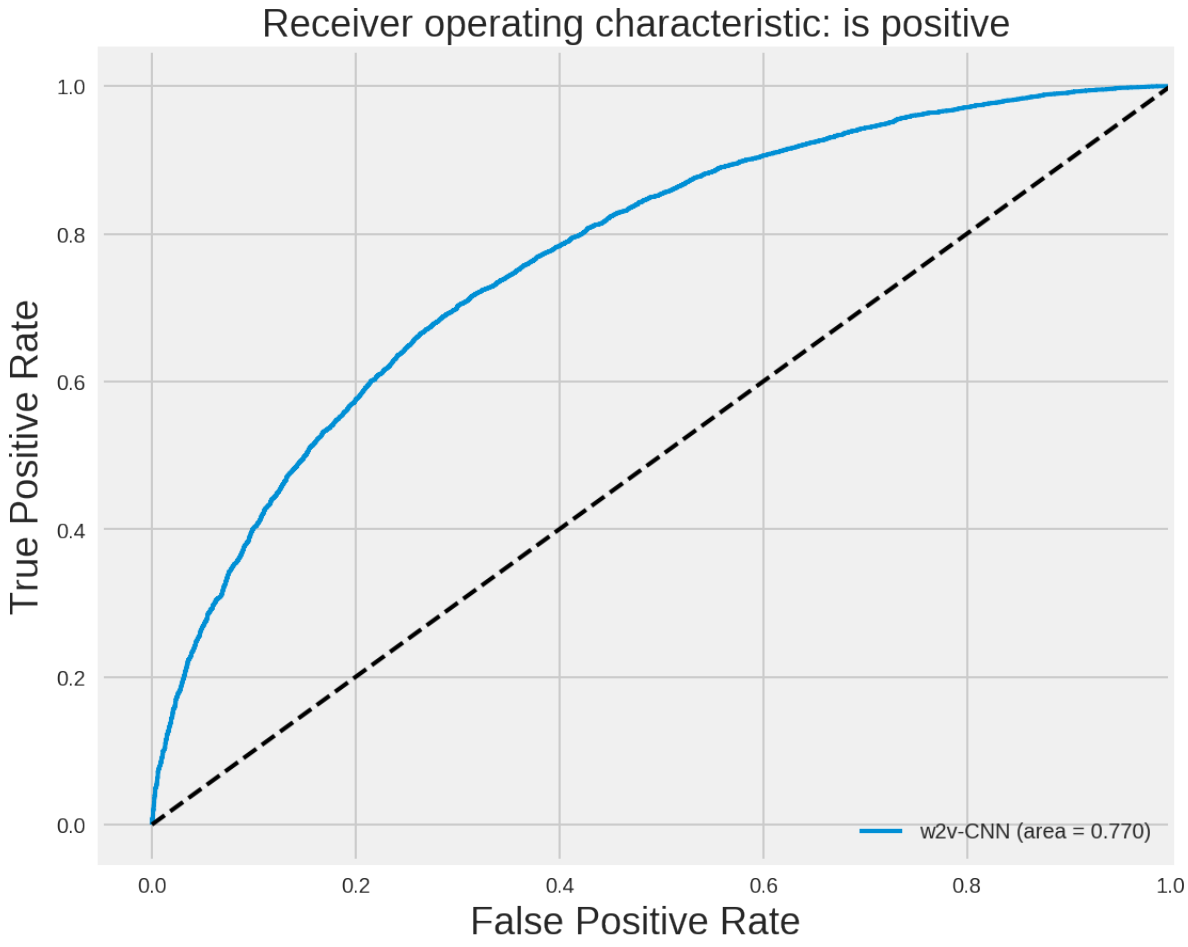

Figura 4.14: Curva ROC

La curva ROC mostrada sobre estas lineas se realizo a partir del set de testeo previamente definido en las primeras secciones del presente capítulo. Se puede desprender de la curva que la tasa de verdaderos positivos supera ampliamente a la tasa de falsos positivos, lo que da una noción del rendimiento del clasificador binario. El área bajo la curva tiene un valor de 0.770 por lo que se puede concluir que el resultado del test ha sido bastante satisfactorio.

Capítulo 5

Conclusiones y Trabajo futuro

5.1. Conclusiones

Este trabajo de memoria consiste en comprobar el funcionamiento de redes neuronales convolucionales en el análisis de texto. Las CNN han sido ampliamente usadas en procesamiento de imágenes, sin embargo en el ámbito del lenguaje natural se ha experimentado relativamente poco.

El uso de la ironía, así como matices en el uso de las palabras y el la presencia de elementos no verbales (emoticono) supone todo un desafió a la hora de clasificar las oraciones. Esto es debido a que todos los elementos antes mencionados tienen la facultad de cambiar radicalmente el valor semántico de la oración. En este trabajo se han ignorado para focalizarse solo en los aspectos puramente verbales del texto.

Por medio de trabajos como el expuesto en este documento, las maquinas se acercan de forma lenta pero constante a la posibilidad de que un Software tenga la capacidad de comprender de forma correcta el lenguaje natural escrito y hablado de los seres humanos.

Gracias a los resultados obtenidos, se comprueba que es posible implementar un sistema de gestión de reclamos para el sistema de salud publico y privado. Esto permitirá mejorar la calidad del sistema de salud en general; es posible empezar a detectar de forma automática instituciones que pudieran estar vulnerando los derechos de los usuarios del sistema de salud.

5.2. Trabajo Futuro

Se propone crear un sistema capaz de comprender y clasificar audio. Esto representa un tarea mas ardua ya que no solo hay que considerar todo los factores incluidos en este trabajo de memoria, sino que se deben tomar en cuenta variables como entonación, pausas y volumen del sonido.

Debido a la falta de bases de datos clasificadas en español, se optó por realizar el trabajo en Inglés. Sería de mucha la creación de set de datos en el idioma español. El Español cuenta con una cantidad altísima de tiempos verbales, también muchos dialectos distintos, etc. El desarrollo de sistemas inteligentes de análisis de texto permitiría un impulso en la ciencias computacionales hispanas.

Bibliografía

[1] Multiples autores. Twitter. Online, March 2006.

[2] L. Arco C. Torres. Representación textual en espacios vectoriales semánticos. *Revista Cubana de Ciencias Informáticas*, 2016.

[3] M. Congosto. Lo que siempre quiso saber del api de twitter y nunca se atrevió a preguntar. Online, October 2017.

[4] Andrés Gonzaléz. ¿qué es machine learning? Online, July 2014.

[5] A. Gardey J. Perez. Definicion de web 2.0. Online, June 2010.

[6] Y. Kim. Convolutional neural networks for sentence classification. *IEEE*, 2014.

[7] M. Mayo. Preprocesamiento de datos de texto. Online, May 2018.

[8] A. Ramiréz. Procesos de datos: sus fases y la gereración de valor. Online, July 2016.

[9] E. Ribas. ¿qué es el data mining o minería de datos? Online, January 2018.

[10] G. Corrado J. Dean T. Milokov, K. Chen. Efficient estimation of word representations in vector space. *IEEE*, 2013.

[11] K. Chen G. Corrado J. Dean T. Milokov, I. Sutskever. Distributed representations of words and phrases and their compositionality. *IEEE*, 2013.

[12] G. Hinton Y. LeCun, Y. Bengio. Deep learning. *Nature*, 2015.