CIMAT Centro de Investigación en Matemáticas, A.C.

Detección de depresión en redes sociales con modelos de aprendizaje profundo: Una revisión de su desempeño e interpretabilidad

T E S I S

Para obtener el grado de

Maestro en Cómputo Estadístico

Presenta:

Román Castillo Casanova

Director de Tesis:

Dr. Adrián Pastor López Monroy

Co-director de Tesis:

Dr. Víctor Muñiz Sánchez

Autorización de la versión final

Monterrey, Nuevo León, 15 de octubre de 2021

Para papá y mamá, por supuesto ...

Resumen

En este trabajo exploramos el desempeño de clasificadores basados en redes neuronales profundas en la tarea de detección de depresión a partir de textos obtenidos de redes sociales. La relevancia se encuentra en que siendo la depresión la principal causa de discapacidad y suicidio a nivel mundial, es necesario contar con herramientas que auxilien en la correcta detección de la misma. Estudios han hallado que la depresión afecta la forma en la que las personas se comunican, y también que personas que las padecen suelen usar las redes sociales para informarse o hablar sobre su condición debido a la seguridad que el anonimato les confiere. Desde hace unos años, ha sido una tarea de interés el desarrollo de mecanismos eficientes para la detección de depresión a partir de textos provenientes de redes sociales, unido a este esfuerzo en este trabajo se explora el desempeño que diferentes arquitecturas de aprendizaje profundo pueden tener en esta tarea, ya que estos modelos profundo han demostrado una alta capacidad de descubrir patrones o características de los datos que no suelen ser evidentes a primera vista. Es por eso que en los últimos años han ganado popularidad como propuestas de solución en problemas complejos. Obtuvimos resultados competitivos para las diferentes redes neuronales profundas puestas a prueba, con resultados mejores con respecto a otros métodos de clasificación populares. Por otra parte, también exploramos la posibilidad de interpretar la forma que los modelos dirigen su aprendizaje, en este punto logramos encontrar patrones interesantes sobre los que las modelos consideran relevante.

Palabras clave: Depresión, Redes sociales, Aprendizaje profundo, Mecanismos de atención

Agradecimientos

Sin duda, totalmente agradecido a mis padres por todo el esfuerzo realizado conmigo y todo el amor que me han regalado. Mamá, gracias por guiarme desde pequeño, por tu empeño en ponerme metas para buscar ser mejor en todo momento, gracias por hacerme un hombre de valores. Papá, gracias por ser mi ejemplo de trabajo , constancia, por enseñarme el valor de perseguir en lo que crees hasta conseguir la meta. A mis hermanos: Claudia, Judith, Dario y Martín por siempre creer en mí aún en los momentos en los que yo mismo no podía hacerlo. Familia, mi cariño y mis logros son suyos.

A mis asesores, Dr. Adrián Pastor López Monroy y el Dr. Victor Muñiz Sánchez por su paciencia, disposición para enseñar, calidez humana, empatía y por estar siempre pendientes de mis avances en este proyecto. Ha sido una gran experiencia poder trabajar con ustedes y de esta me llevo mucho aprendizaje.

Al CIMAT por todo el apoyo brindado durante mis estudios. Por hacerme sentir y saber que estaba en casa.

A mis amigos, mi pequeña familia CIMAT MTY con quienes pude crecer profesionalmente y como persona. Ester :) gracias por tu apoyo y amistad cuando más necesitaba de ella. Amigos, me llevo todo tipo de recuerdos de este par de años juntos, sé que aún nos queda mucho que contar, esto apenas comienza.

Finalmente, agradezco al CONACYT por la beca que me permitió sostener mis estudios de maestría.

Índice general

Resumen III

Agradecimientos V

1. Introducción 1
- 1.1. e-Risk . . . 2
- 1.2. Motivación . . . 2
- 1.3. Objetivo . . . 4
- 1.4. Organización de la tesis . . . 4

2. Trabajo relacionado 7

3. Marco teórico 9
- 3.1. Clasificación de textos . . . 9
- 3.2. Representación vectorial de palabras y textos . . . 11
  - 3.2.1. Modelo neuronal del lenguaje . . . 13
  - 3.2.2. Word-Embbedings . . . 16
- 3.3. Métodos de clasificación . . . 19
  - 3.3.1. Máquinas de soporte vectorial . . . 20
  - 3.3.2. Redes Neuronales . . . 20
  - 3.3.3. Mecanismos de atención . . . 26

4. Metodología 31
- 4.1. Descripción de los datos . . . 31
- 4.2. Clasificadores . . . 33
  - 4.2.1. Baselines . . . 33
  - 4.2.2. Redes convolucionales . . . 34
  - 4.2.3. Redes recurrentes . . . 34
  - 4.2.4. Modelos de fusión . . . 36
- 4.3. Ajustes y configuraciones de los experimentos . . . 36
  - 4.3.1. Preprocesamiento . . . 36
  - 4.3.2. Análisis exploratorio . . . 39
  - 4.3.3. Ajuste y detalles de experimentos . . . 41
  - 4.3.4. Métricas . . . 44

5. Resultados **47**
5.1. Distribución de sentimientos 47
5.2. Resultados de los clasificadores 48
5.2.1. Baselines 48
5.2.2. Redes convolucionales 50
5.2.3. Redes recurrentes 50
5.2.4. Modelos de fusión 51
5.2.5. Visualización del aprendizaje arquitectura BiLSTM (SP) . . . 52

6. Conclusiones y trabajo futuro **57**

Índice de figuras

3.1. Clasificación supervisada de texto. 11
3.2. Representación vectorial de documentos. 12
3.3. Arquitectura del modelo neuronal. 15
3.4. Representación 2D de Word Embbedings. 17
3.5. Entrenamiento Word2Vec 18
3.6. Modelo Glove para obtención de word embeddings 18
3.7. Clasificación binaria con máquinas de soporte vectorial. 19
3.8. Kernel Trick. 20
3.9. Funciones de activación. 21
3.10. Modelo de una red neuronal simple. 22
3.11. Estructura clásica del CNN 23
3.12. Estructura clásica RNN 24
3.13. Estructura clásica LSTM 26
3.14. Mecanismo de atención 27
3.15. Mecanismo de *Self attention* 28
3.16. Estructura de modelo de de BiLSTM self-attention 28

4.1. Estructura de red convolucional para la clasificación de texto 34
4.2. BiLSTM con Mecanismo de atención para la clasificación de texto . . 35
4.3. Estructura de los modelos de fusión 36
4.4. Procesamiento del conjunto de datos 37
4.5. Distribución de cantidad de palabras de los textos 40
4.6. Distribución de largo de textos entre el grupo de interés y el grupo de control 42
4.7. Estructura de CNN 43
4.8. Estructura de BiLSTM con mecanismo de atención 44

5.1. Distribución de emociones entre clases 48
5.2. Visualización de la atención 53
5.3. Visualización de relevancia de las palabras 53
5.4. Efecto del etiquetado de emociones en la atención 54
5.5. Nube de palabras relevantes para la clasificación de la clase con depresión 54

Índice de tablas

4.1. Características de los datos 39

5.1. Desempeño en conjunto de prueba de clasificadores SVM. Nivel Usuario 49
5.2. Desempeño en conjunto de prueba de clasificadores SVM. Nivel Post . 49
5.3. Resultados de Experimentos CNN 50
5.4. Resultados de Experimentos Bi-LSTM-Att 51
5.5. Desempeño modelos de fusión 51
5.6. Palabras agrupadas por tópicos 56

Capítulo 1

Introducción

Las redes sociales se han convertido en una pieza fundamental en las interacciones de las personas. Se estima que los usuarios dedican cerca de dos horas al día a sus redes sociales (Statista)[62] . En estas plataformas, las personas interactuan con otras, ya sea contacto directo por mensajeria, o 'muros' o 'foros' donde comparten texto, audio o vídeo. La mayoria de las personas usa estos canales para expresar sus pensamientos y sentimientos.

Por lo anterior, resulta natural el creciente interés en usar el material proveniente de redes sociales como una fuente de información para distintas áreas de interés público. Se ha mostrado eficacia en diferentes estudios que van desde la identificación de la propagación de los síntomas de la gripe (Sadilek y col.)[59] hasta la construcción de ideas sobre enfermedades basadas en publicaciones en Twitter (Paul y col., Katikalapudi y col.)[49, 23]. En el contexto del estudio de trastornos mentales, estudios como (Moreno y col.)[43] encontraron que las actualizaciones de estado en Facebook podrían revelar síntomas de episodios depresivos mayores. También por su parte (Park y col.)[47] encontraron evidencia inicial de que las personas publican sobre su depresión e incluso su tratamiento en las redes sociales.

1.1. e-Risk

Desde 2017, se celebra el laboratorio denominado *e-Risk*, (Losada y col., Losada y col., Losada y col.)[34, 35, 36] cuyo objetivo principal es promover la discusión sobre la creación de puntos de referencia reutilizables para evaluar algoritmos de detección temprana de riesgos, explorando temas de evaluación metodología, métricas de efectividad y otros procesos relacionados con la creación de colecciones de datos para pruebas de la detección temprana de riesgos. (Losada y col.)[33]

Entre los desafíos propuestos por el laboratorio se encuentran la detección temprana de depresión [1]. El desafío consiste en procesar secuencialmente piezas de evidencia y detectar los primeros indicios de depresión lo antes posible. La tarea se centra principalmente en evaluar soluciones de minería de texto y, por lo tanto, se concentra en textos escritos en redes sociales. Los textos deben procesarse en el orden en que fueron creados. De esta forma, los sistemas que realizan esta tarea de manera efectiva podrían aplicarse para monitorear secuencialmente las interacciones de los usuarios en blogs, redes sociales u otros tipos de medios en línea. En la versión 2018 (Losada y col.)[35] del laboratorio se agregó la tarea de la detección de anorexia y en 2019 (Losada y col.)[36], la detección de autolesiones, en el mismo formato de la primera tarea.

Entre las soluciones propuestas en el E-Risk, se encuentran algunas que mezclaron enfoques de procesamiento de lenguaje natural, *machine learning*, búsqueda y recuperación de información. El desempeño en esta tarea ha sido relativamente bajo, esto muestra su nivel de complejidad y un espacio de oportunidad para proponer y probar métodos de clasificación.

1.2. Motivación

La depresión es una enfermedad frecuente en todo el mundo, y se calcula que afecta a más de 300 millones de personas, cantidad que representa un poco más del

[1] Aunque gran parte del trabajo relacionado está enfocado en la detección temprana de la depresión, cabe recordar al lector que este no es el enfoque de esta tesis

4 % de la población. Si bien las formas de depresión son más comunes entre las mujeres (5.1 %) que entre los hombres (3.6 %) y la prevalencia difiere entre las regiones del mundo, ocurre en cualquier grupo de edad y no se limita a ninguna situación de vida específica (Organización Mundial de la salud)[46]. La depresión es distinta de las variaciones habituales del estado de ánimo y de las respuestas emocionales breves a los problemas de la vida cotidiana, puede convertirse en un problema de salud serio, especialmente cuando es de larga duración e intensidad moderada a grave, y puede causar gran sufrimiento y alterar las actividades laborales, escolares y familiares. En el peor de los casos puede llevar al suicidio. Cada año se suicidan cerca de 800 000 personas, y el suicidio es la segunda causa de muerte en el grupo etario de 15 a 29 años (Organización Mundial de la salud)[46].

No es sencillo definir a la depresión, no solo por los diferentes subtipos que se han descrito y que han cambiado en a lo largo de los años (Paykel)[50], sino también la expresión 'estar deprimido' se ha adoptado al lenguaje cotidiano. En general, se puede describir que la depresión como un estado de ánimo alterado que puede ir acompañada, por ejemplo, con una imagen negativa de sí mismo, distorsión en la percepción de hechos cotidianos, deseos de escapar o de aislarse y cambios súbitos en patrones de conducta. Los síntomas experimentados por las personas deprimidas pueden afectar gravemente su capacidad para hacer frente a cualquier situación en la vida diaria y, por lo tanto, difieren drásticamente de las variaciones normales del estado de ánimo que cualquiera experimenta

Aunque hay tratamientos eficaces para la depresión, más de la mitad de los afectados en todo el mundo (y más del 90 % en muchos países) no los recibe. Además, existen otros obstáculos para la atención eficaz tales como la falta de recursos y de personal sanitario capacitado en diagnóstico y tratamiento, la fuerte estigmatización social hacia los trastornos mentales y el hecho de que en muchos la evaluación clínica es inexacta o incluso errónea. Actualmente los tratamientos eficientes se basan en la detección temprana de la depresión, tales como métodos de auto-reporte, cuestionarios , encuestas, entrevistas guiadas y valoración clínica. Sin embargo, estos métodos

adolecen en el proceso de captura y procesamiento retrasando el diagnóstico[2]. En definitiva es necesario explorar otras fuentes de información que puedan asistir en la tarea de detectar la depresión de forma acertada, en menor tiempo, que carezca del sesgo natural de un clasificador humano. En este contexto es evidente que procesar textos de redes sociales con métodos de aprendizaje automático es un área de oportunidad en la lucha en contra de este padecimiento.

1.3. Objetivo

El objetivo principal de este trabajo, es explorar la efectividad y la interpretabilidad de modelos de aprendizaje profundo entrenados con texto proveniente de redes sociales en la tarea de identificación de signos de depresión, esto nos conduce a dos objetivos específicos:

1. Evaluar la efectividad de modelos de aprendizaje profundo usados en tareas de clasificación de textos y comparar el desempeño con otros clasificadores usados en la tarea de detección de depresión con textos de redes sociales

2. A partir de modelos entrenados en la tarea descrita que poseen un buen desempeño, obtener representaciones que permitan entender el proceso de aprendizaje del modelo.

1.4. Organización de la tesis

En el Capítulo 2 se haya una relatoría de trabajo previos que sentaron las bases en las actuales investigaciones relacionadas con la tarea de detección de depresión, esto para poder brindar un contexto del estado en el que se encuentra la tarea, también enumeramos algunos primeros acercamientos y propuestas para la tarea. En el Capítulo 3 se introduce los conceptos técnicos básicos que permiten desarrollar este trabajo, primeramente se describe la tarea de clasificación de textos y los retos propios

[2]También hay que señalar que en muchos es necesario un grupo de expertos que puedan realizar un diagnóstico confiable.

de la tarea, también son descritas algunas representaciones útiles de los texto usadas, así como una breve revisión de clasificadores populares en la clasificación de texto así como la intuición inherente a ellos. Posteriormente en el capitulo 4, se describe las arquitecturas de aprendizaje profundo de interés en este trabajo, así como herramientas adicionales que serán usadas para el enriquecimiento de los datos, ajustes a los modelos utilizados y el análisis exploratorio del conjunto de datos.

En el capítulo 5 se describe los resultados obtenidos en los diferentes experimentos, así como resultados de interés relativos al aprendizaje de los modelos entrenados. Finalmente en el capitulo 6 se realiza un resumen de los hallazgos de interés de este trabajo, así como una reflexión sobre el trabajo futuro.

Capítulo 2

Trabajo relacionado

En los últimos años, con el desarrollo de las redes sociales y la era del internet la cantidad de trabajo relacionado con el uso de de redes sociales en tareas de detección o modelo de fenómenos de interés social ha ido en aumento. Sitios como Facebook, Twitter, Reddit se han convertido en plataformas de interés para la investigación innovadora que pueda contar con fuentes ricas de información que permitan capturar comportamientos o características de los usuarios de estas redes. Previamente se ha mencionado el estudio (Sadilek y col.)[59] que usaron datos de Twitter sobre la geolocalización de su usuarios para modelar los patrones de propagación de gripa. También (Paul y col.)[49] modelaron el estado de salud de las personas usando sus tuits, buscando detectar de manera oportuna factores de riesgo. (Earle y col.)[15] usaron las interacciones en Twitter para predecir eventos sísmicos. (D'Andrea y col.)[10] también con datos de Twitter pronostican embotellamientos de tráfico. Estos son solo algunos ejemplos de cómo las redes sociales se convierten en fuentes valiosas de información.

Con respecto a tareas de detección podemos mencionar el trabajo (Yin y col.)[73] que se centró en la detección de acoso dentro foros *online*, los autores extrajeron características de los mensajes en los foros y plantearon un clasificador basado en una máquina de soporte vectorial. Por su parte (Lin y col.)[31] detectan estrés psicológico con información de textos provenientes de redes sociales usando redes convolucionales. El trabajo de (Del Bosque y col.)[13] se centra en la detección agresividad, ciberacoso usando redes neuronales y máquinas de soporte vectorial. También el uso

de redes profundas en tareas de detección se ha popularizado. Por ejemplo (Pitsilis y col.)[52] usan redes profundas para la detección de discursos de odio, (Ma y col.)[37] para detección de rumores, (Chen y col.)[8] para detección de agresiones verbales y (Akhtyamova y col.)[1] para reacciones adversas a drogas .

En este mismo contexto, podemos mencionar trabajos orientados a detección de trastornos mentales de interés en la salud pública. Por ejemplo (Astoveza y col.)[2] detectan comportamiento suicida. En la detección temprana de la anorexia podemos mencionar a (Wang y col.)[67]. Con respecto a la tarea de interés, es decir la detección de depresión previamente referimos al trabajo de (Park y col.)[47] que encontró diferencia en la forma de comunicarse en las personas que sufren o habían sufrido algo episodio depresivo. (Karmen y col.)[21] detectan signos de depresión en foros de internet montando diferentes métodos de NLP.(Katchapakirin y col.)[22] Detecta signos de depresión dentro la comunidad Thai, a partir de las interacciones en Facebook, usando diferentes métodos de NLP para la extracción de características, y entrado el clasificador con una maquina de soporte vectorial. [11] extraen características lingüísticas de tuits para detectar depresión posparto. También [12] realiza un estudio de detección de depresión en redes sociales, donde identifica a los tópicos característicos del grupo en depresión. [43] evalúa y predice grados de depresión en alumnos usando información sobre sus cambios de estado dentro de la red social e información demográfica proporcionada por la misma plataforma. En años recientes en marco del Foro y Laboratorio para la predicción temprana de riesgos (CLEF eRisk por sus siglas en inglés) varios grupos de investigación han presentado sus propuestas para la solución a la tarea planteada dentro del foro de detección temprana de la depresión [35, 36]. También en tiempos recientes el trabajo de (Tadesse y col.)[64] donde además de explorar la identificacion depresion, realizan un análisis de tópicos obtenidos.

Capítulo 3

Marco teórico

En este capítulo se introducen los conceptos básicos necesarios para abordar los siguientes capítulos. Primeramente en la Sección 3.1 se describe la naturaleza de la tarea de clasificación automática de texto, en la Sección 3.2 se presenta algunas formas usadas para la representación de texto y las ventajas que proporcionan. Finalmente,en la Sección 3.3 se describe algunos métodos de aprendizaje supervisado que son usados en este trabajo

3.1. Clasificación de textos

La gestión de documentos basados en su contenido que en general se conoce como recuperación de información (IR, *information retrieval* en inglés) se destaca en el campo de los sistemas de información, debido la gran cantidad de documentos digitales disponibles y la necesidad acceder a su contenido de manera flexible (Sebastiani)[61]. Dentro de esta gestión está la clasificación de texto, es decir la tarea de asignar un documento a categorías temáticas de un conjunto predefinido, en función del lenguaje en ellos. Aunque esta tarea se remonta a los años 60, solo solo a principios de los años 90 se convirtió en un subcampo importante de la disciplina de los sistemas de información, gracias al mayor interés aplicativo y a la disponibilidad de hardware más potente. En la actualidad la clasificación de textos se está aplicando en diferentes escenarios, desde la indexación de documentos (Baharudin y col.)[3], filtrado (Fumera

y col.)[16], la generación automatizada de metadatos (Hess y col., Lerman y col.)[20, 29], la desambiguación del sentido de las palabras (Bollegala y col., Turdakov)[7, 66].

En sus inicios la clasificación de textos era una tarea bastante operativa, ya que era necesario un experto o conjunto de expertos que definía manualmente el conjunto de reglas para clasificar un texto en una categoría en particular. Desde los años 90, este enfoque perdió popularidad a favor del aprendizaje máquina (*machine learning* en inglés, ML), donde ahora se construye un clasificador con la capacidad de aprender de forma automática las características de las categorías de interés, a partir de un conjunto de documentos previamente clasificados. Este enfoque llega a tener una precisión comparable a la lograda por expertos humanos, además que representa un ahorro considerable en términos de mano de obra y de tiempo.

Entonces si consideramos los pares $< d_j, c_i >_{\in \mathbf{DxC}}$, donde $\mathbf{D}$ es el conjunto de documentos y $\mathbf{C}$ un conjunto predefinido de *categorías* y T un booleano asignado a cada par $< d_j, c_i >$ que indica la decisión de clasificar o no al documento j dentro de la categoría i. Con mayor formalidad la tarea de la clasificación de textos consiste en aproximar la función objetivo desconocida $\Phi : \mathbf{DxC} \rightarrow T$, (la cuál describe la forma en la que los documentos deben ser clasificados) por medio de la función $\tilde{\Phi} : \mathbf{DxC} \rightarrow T$ que nombramos como *clasificador* tal que $\tilde{\Phi}$ y Φ 'coinciden lo mayor posible' (Sebastiani)[61].

En la Figura 3.1, se presenta el proceso de clasificación de textos. En primer lugar, se debe contar con un conjunto de documentos clasificados manualmente, llamado *conjunto de entrenamiento*, del cual se extraen las características de interés y se aplica el algoritmo de aprendizaje seleccionado, con esto, el clasificador queda entrenado (Liu y col.)[32] . Por último, se evalúa el desempeño del clasificador con un conjunto de documentos nuevos (documentos nunca antes vistos), llamado *conjunto de prueba*

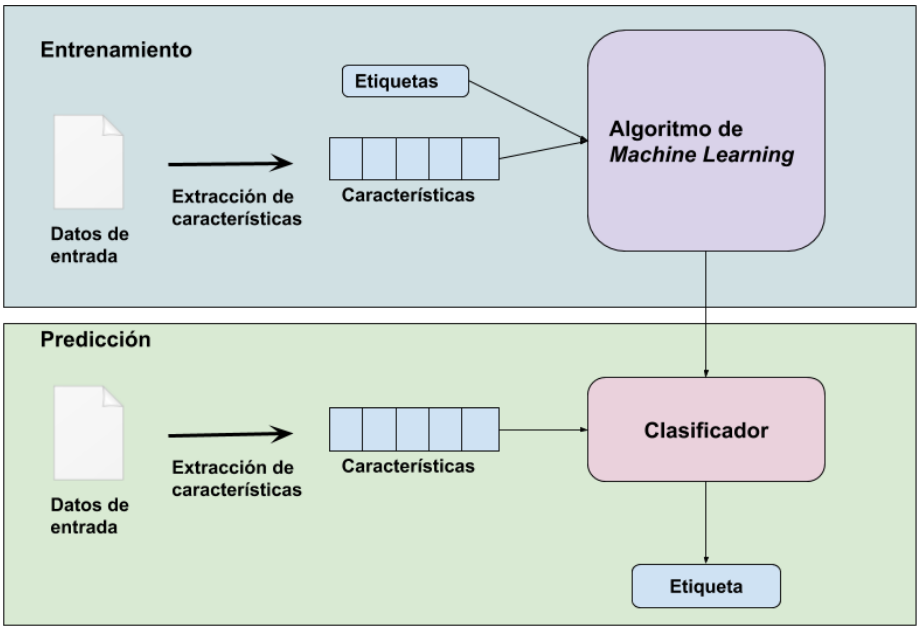


Figura 3.1: Clasificación supervisada de texto. Durante el entrenamiento se obtienen los vectores de características que capturan la información relevante de los datos, estos junto con las etiquetas son las entradas en el algoritmo de aprendizaje para obtener el clasificador entrenado. En la fase de predicción se obtienen características de ejemplos no observados en el entrenamiento, con las cuales el clasificador entrenado predice sus etiquetas.Reproducido de (Steven Bird y col.)[63]

3.2. Representación vectorial de palabras y textos

Existen varias maneras de representar un documento para ser procesado por un clasificador, siendo la más usada el modelo de espacio vectorial (Salton y col.)[60] . En este modelo, los documentos son representados por vectores de palabras en un espacio de n dimensiones, siendo n el número de palabras en un vocabulario definido por los documentos. De esta manera, los documentos quedan representados como un vector $d = (w_1, ..., w_n)$, donde cada término indexado corresponde a una palabra en el texto y tiene un peso asociado a él, que refleja la importancia del término ya sea para el documento o para la colección completa de documentos. El peso de la i-ésima palabra o término del documento se representa por w_i. En la Figura 3.2 se muestra la representación de documentos en el modelo vectorial.

	eveyrthing	interesting	learning	lerning	like	Machien	machine	not	predicts	problems	solving	sure	What
1	0	1	0	0	1	0	0	0	0	1	1	0	0
2	0	0	1	0	0	0	1	0	0	0	0	0	1
3	0	0	0	0	0	0	0	1	0	0	0	1	0
4	1	0	0	1	0	1	0	0	1	0	0	0	0

Figura 3.2: Representación vectorial de documentos.(One-Hot). Cada columna representa una palabra del vocabulario y cada renglón representa un documento. Cada celda indica la presencia o no de la palabra en el documento, al final cada documento es representando por un vector numérico del tamaño del vocabulario. Reproducido de (Paruchuri)[48]

Existen distintos esquemas para obtener el peso de cada palabra dentro del documento, los esquemas más utilizados en el área de clasificación automática son: *One-Hot*, Frecuencia del Término y Frecuencia del Término-Frecuencia Inversa del Documento. Estos se detallan a continuación

- *One-Hot*: Puede ser 1 ó 0. Depende si la palabra aparece o no en el documento.

$$w_i = \begin{cases} 1, & \text{si el i-ésimo término aparece en el documento} \\ 0, & \text{otro caso} \end{cases} \tag{3.1}$$

- *Frecuencia del Término (TF)* : Frecuencia del término i en el documento (f_i). Bajo este enfoque, las palabras más repetidas son las que tienen mayor relevancia dentro del documento. Este enfoque también se conoce como bolsa de palabras (*Bag of Words* BoW) .

$$w_i = f_i \tag{3.2}$$

- *Frecuencia del Término-Frecuencia Inversa del Documento (TFIDF)*: Combina la frecuencia del término en el documento (f_i) con la frecuencia de éste en el resto de los documentos de la colección. Donde N es el tamaño de la colección de documentos y n_i es el número de documentos en los que aparece el término i-ésimo.

$$w_i = f_i \cdot \log\left(\frac{N}{n_i}\right) \tag{3.3}$$

La representación de la forma 'bolsa de palabras', es una de las más populares en la actualidad, debido que a pesar de ser bastante sencilla en la práctica permite recoger suficiente información del texto para un obtener un clasificador bueno.

Estas representaciones únicamente se basan en la frecuencia en la que aparece un palabra dentro del documento, así que no consigue rescatar características más abstractas del lenguaje, tales como la semántica y la relación entre palabras. Ahora se presenta una forma de representación de las palabras que busca rescatar esas características propias del lenguaje

3.2.1. Modelo neuronal del lenguaje

Un modelo del lenguaje μ asigna probabilidades a una secuencia de palabras a partir de la estimación de la probabilidad de transición. Así, dado una secuencia $w_1...w_m$ en un lenguaje determinado, un modelo del lenguaje busca estimar la probabilidad $p(w_1...w_m)$.

Los modelos del lenguaje tradicionales, se basan en la estimación de las probabilidades de transición y las probabilidades iniciales a partir del método de Máxima Verosimilitud, lo que determina un cálculo de probabilidades basado en las frecuencias relativas de los n-gramas (Manning y col.)[39].

Podemos expresar $p(w_1...w_m)$ como el producto de probabilidades condicionales de la forma:

$$p(w_1 \dots w_m) \approx p(w_1) \prod_{i=2}^{m} p(w_i|w_{i-1} \dots w_1) \tag{3.4}$$

Estas probabilidades condicionales representan la transición desde el inicio de la cadena en el símbolo w_1 (cuya probabilidad inicial está dada por $p(w_1)$) hasta el fin de la cadena en el símbolo w_m. La probabilidad de observar el símbolo w_m dado que se ha recorrido la cadena está representada por la probabilidad condicional $p(w_m|w_{m-1} \dots w_1)$. La transición a cada símbolo esta representada por una probabilidad del tipo $p(w_i|w_{i-1} \dots w_1)$, con $i = 1, \dots, m$. La probabilidad de la cadena esta dada, entonces, por el producto de estas probabilidades de transición.

La estimación de las probabilidades de transición no es una tarea sencilla, pues cadenas del lenguaje con una extensión considerable no serán observadas con suficiente frecuencia para estimar una probabilidad adecuada. Por tanto, los modelos del lenguajes suelen asumir la independencia de transiciones pasadas. El modelo mas común para solucionar este problema es el modelo de n-gramas. Este modelo busca estimar las probabilidades condicionales de transición a partir de asumir la dependencia únicamente de los n elementos anteriores a una palabra dada. (Manning y col.)[39].

En este sentido, las probabilidades de transición se aproximan a partir de probabilidades de n-gramas dadas por:

$$p(w_i|w_{i-1}...w_1) \approx p(w_i|w_{i-1}...w_{i-n+1}) \quad \text{con } i = 1, \ldots, m \text{ y } n \leq m \tag{3.5}$$

Entonces, la probabilidad de transitar a una palabra w_i está determinada solamente por los n elementos anteriores; es decir, los elementos previos a wi_{i-n+1} son ignorados. Esta forma de representar las probabilidades proviene de asumir la propiedad de Markov (de orden n), la cual consiste, precisamente, en asumir que las transiciones en un proceso estocástico son dependientes únicamente de los n estados anteriores. La ventaja de los modelos de n-gramas es que simplifican el calculo de probabilidades, reduciendo el tamaño del contexto en que los símbolos se presentan. Esto permite observar con mayor frecuencia las cadenas contextuales que definen las transiciones.

Puede notarse que los cálculos necesarios para la estimación de las probabilidades $p(w_i|w_{i-1}, \ldots, w_{1-n+1})$ crece al aumentar n. Es decir existe un problema de capacidad computacional. En este sentido (Bengio y col.)[6] abordan este problema y proponen una solución para la estimación de parámetros del modelo a partir de un modelo continuo basado en redes neuronales. (En la sección 3.3.2 se detalla lo qué son las redes neuronales)

Las idea principal de la solución de (Bengio y col.)[6] puede resumirse en tres puntos:

1. Asociar un vector en $\mathbb{R}^d$ a cada palabra en el vocabulario

2. Expresar la probabilidad conjunta de secuencias de palabras en términos de los vectores de las palabras en la secuencia

3. Aprender al mismo tiempo, la representación vectorial de las palabras y los parámetros de las probabilidades

En la Figura 3.3 puede observarse la arquitectura de la red neuronal para la solución propuesta por Bengio. La entrada de la arquitectura son vectores *one-hot* en $\mathbb{R}^N$, es decir que cada palabra w_i en el vocabulario, con $i = 1, \ldots, N$, es representada con unos de estos vectores que tiene un 1 en la posición i de la palabra y 0 en el resto de sus entradas.

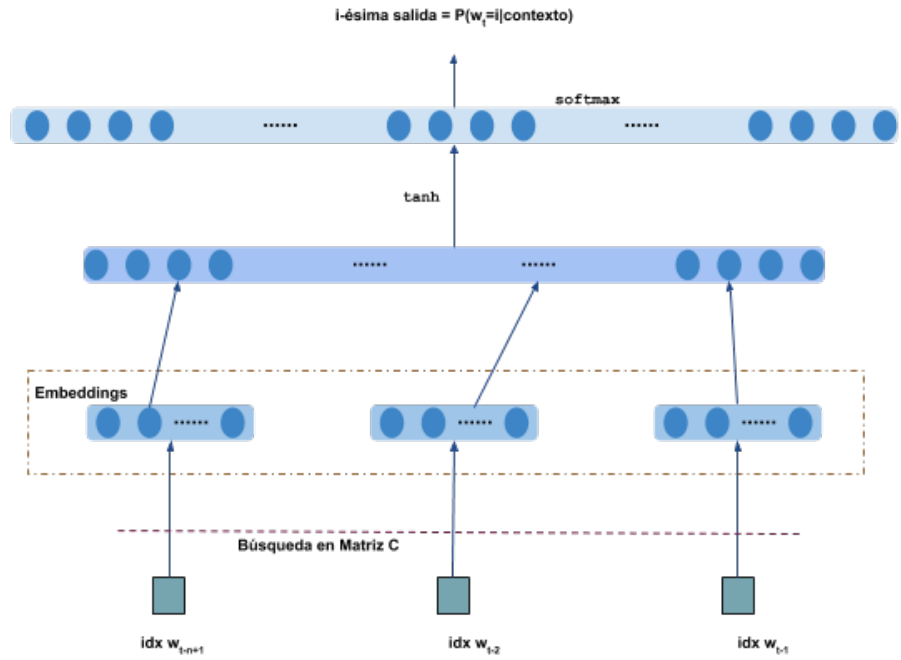


Figura 3.3: Arquitectura del modelo neuronal. Ingresan las palabras de contexto como *one-hot vectors* donde sus representaciones densas son concatenadas y activadas con una la función tanh, finalmente a traves una representación densa se obtienen las probabilidades condicionales. Basado en (Bengio y col.)[6]

La primera capa de la arquitectura, *capa de embedding*, corresponde a las representaciones vectoriales de las palabras. Siendo la entrada w_i el vector *one-hot*, su representación densa esta dada por la transformación lineal:

$$v_i = \mathbf{W} w_i \tag{3.6}$$

Los vectores $v_{i-1}, \ldots, v_{1-n+1}$ del contexto son concatenados y pasan por una función de activación de tangente hiperbólica.El objetivo de esta capa oculta es obtener representaciones vectoriales de las palabras del vocabulario:

$$h_i = \tanh(\mathbf{W}w_i + b) \tag{3.7}$$

Finalmente la capa de salida se define por :

$$y_i = \mathbf{U}h_i + c \tag{3.8}$$

Finalmente para la estimación de las probabilidades condicionales:

$$p(w_j|w_i) = \text{Softmax}(y_i) \tag{3.9}$$

3.2.2. Word-Embbedings

Actualmente los *word-embeddings* son una de las representaciones más populares del vocabulario en el procesamiento del lenguaje natural, sobre todo para el enriquecimiento de datos (Rezaeinia y col., Xiong y col., Onan)[54, 71, 45]. Los *word embbedings* son capaces de capturar el contexto de una palabra dentro de un documento, la semántica, su sintaxis y su relación con otras palabras. En términos sencillos convertimos cada palabra en un vector donde se encuentran representadas todas las características antes mencionadas. En la Figura 3.4 vemos unas representación gráfica de cómo estos vectores logran capturar características semánticas.

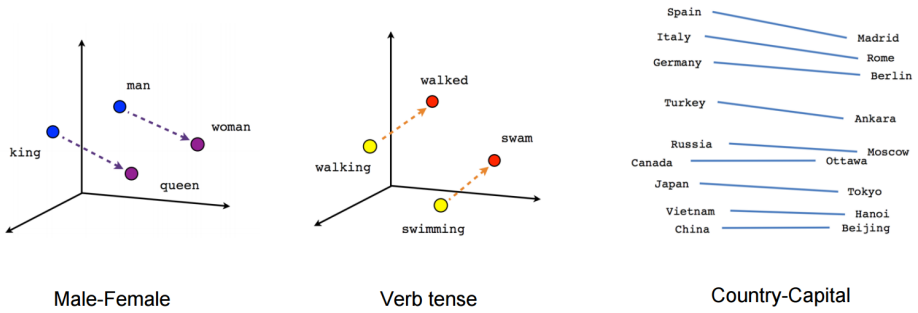


Figura 3.4: Representación 2D de Word Embbedings. Se observa cómo las estructuras semánticas entre palabras relacionadas se mantiene, vemos que palabras en el mismo campo semántico tales como *king* y *queen* mantiene la misma relación que *male* y *female*. También podemos verque se mantiene la relación entre formas conjugadas de verbos, y lugares. Reproducido de (Khandelwal)[24]

Desde la publicación de Word2vec (Mikolov y col.)[40] en 2013, las representaciones vectoriales de las palabras comenzaron a popularizarse y han surgido una gran cantidad de métodos diferentes para generarlos *word embeddings*. A continuación, se mencionan algunos métodos pupulares para su obtención; hemos escogido los word embeddings 'más conocidos' y que tiene buenos resultados en la tarea de clasificación de textos [69, 38, 72, 28, 30].

Wor2Vec Word2Vec es un enfoque que logra obtener vectores similares para palabras similares (Mikolov y col.)[40] . Las palabras que están relacionadas entre sí se asignan a puntos más cercanos entre si en un espacio de dimensión predefinida.

Estas representaciones se obtienen del entrenamiento de una red neuronal con una sola capa oculta. (Mikolov y col.)[40] proponen dos arquitecturas para obtener estos *Word Embbedings*, el modelo *C-BoW* y el modelo *Skip-Gram*. Estas arquitecturas pueden observarse en la Fígura 3.6. En el modelo *C-BoW (Continuos Bag of words)* el modelo trata de predecir la palabra de interés a partir de su contexto, mientras que en el modelo *Skip-Gram* se trata de predecir las palabras del contexto a partir de cada palabra.

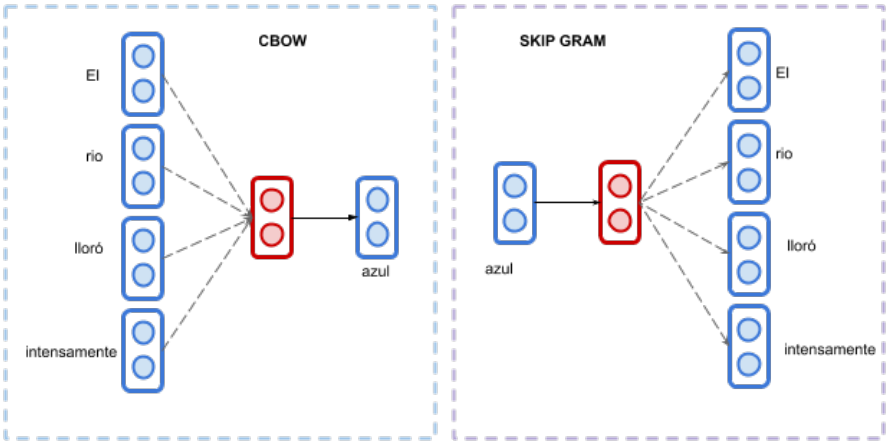


Figura 3.5: Entrenamiento Word2Vec: a) Modelo C-BoW: A la red entran las representaciones *one-hot-encoding* de las palabras de contexto y como salida se obtiene la palabra a la que pertenece este contexto b) Skip-Gram: A la red entra la representación de la palabra de interés y la salida es las palabras que forman parte del contexto. Reproducido de (Ganesan)[17]

Glove Glove (por *Global Vectors*) es un enfoque , presentado por el equipo de NLP de Stanford (Pennington y col.)[51] , en que se aprende las representaciones de las palabras a partir de una matriz de co-ocurrencia, es decir, con qué frecuencia aparecen juntas. GloVe es un modelo basado en conteo. El modelo se basa en una idea bastante simple de que las proporciones de probabilidades de co-ocurrencia palabra-palabra tienen el potencial de codificar alguna forma de significado que se puede codificar como diferencias vectoriales.

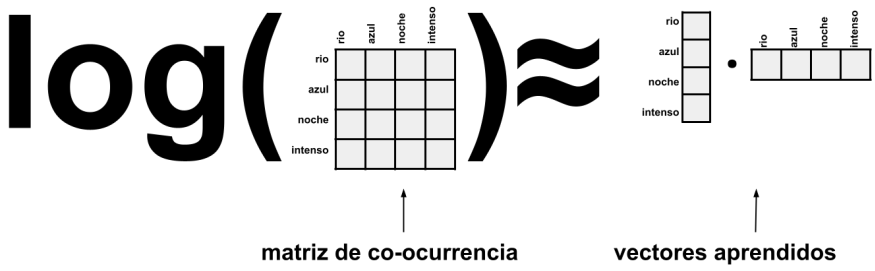


Figura 3.6: Modelo Glove para obtención de word embeddings. El modelo estima representaciones vectoriales de las palabras a traves de la estimación de la matriz de co-ocurrencia

Esto nos lleva a un problema de optimización, el cual tiene la forma la suma de errores cuadrados ponderados, donde $X_{i,j}$ son las veces que la palabra i aparece en

el contexto de j, y w_i, wj sus representaciones vectoriales, que van a ser aprendidas. Puede notarse la inclusión de los *biases* y la función f que tiene como función ponderar ratios que sean extremos ya sea muy cercanos a cero o que tiendan al infinito.

$$\text{mín } J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T w_k + b_i + b_k - \log(X_{ik}))^2 \tag{3.10}$$

3.3. Métodos de clasificación

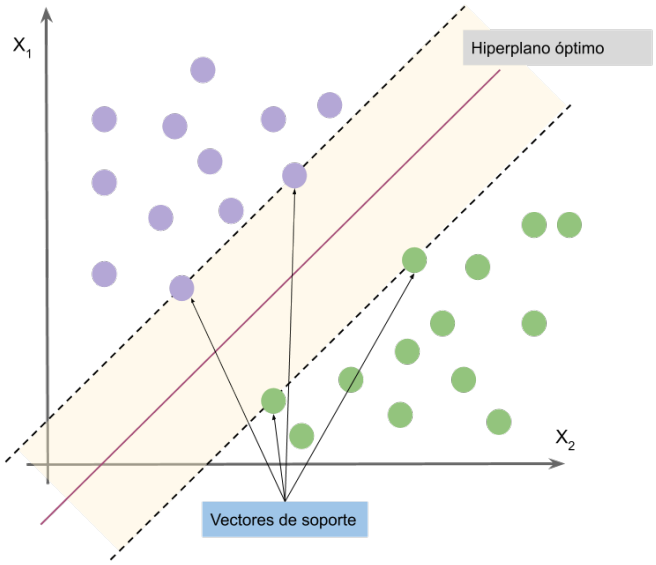


Figura 3.7: Clasificación binaria con máquinas de soporte vectorial.El modelo separa las clases a 2 regiones lo más amplias posible mediante un hiperplano de separación definido por el vector soporte.

Los algoritmos de clasificación permiten encontrar un modelo de clasificación válido que distinga clases para futuras predicciones. Se trata de que el modelo sea capaz de aprender las características que permitan generalizar la clasificación a partir de una poca cantidad de ejemplos. Entre los algoritmos más populares en la clasificación automática de texto se encuentran: los árboles de decisión, las Máquinas de Soporte Vectorial (SVM, Support Vector Machine), los algoritmos basados en reglas, los probabilísticos como las redes bayesianas y los de máxima entropía. A continuación, se describe brevemente las máquinas de soporte vectorial que son de interés en este trabajo.

3.3.1. Máquinas de soporte vectorial

Las máquinas de vectores de soporte (*Support Vector Machine*SVM,) es un método de clasificación supervisada propuesto por (Cortes y col.)[9]. Este método tiene su base en la teoría de aprendizaje estadístico. La idea detrás de este método consiste en buscar hiperplanos que maximicen el margen entre dos clases, separando los ejemplos de entrenamiento positivos de los negativos.

La Figura 3.7 muestra la idea principal en datos linealmente separables, la distancia que existe entre las líneas punteadas se le llama margen y los puntos más cercanos al hiperplano son llamados vectores de soporte.

Un punto importante relacionado con SVM, es que pueden mapear los datos del espacio original a otro espacio de alta dimensionalidad llamado espacio de características, donde se busca maximizar la distancia entre los puntos negativos de los positivos (*kernel trick*), esta idea puede observarse en la Figura 3.8

En general las máquinas de soporte vectorial han mostrado un buen desempeño sobre una amplia variedad de problemas, en particular dentro de la clasificación de textos también se han conseguido buenos resultados

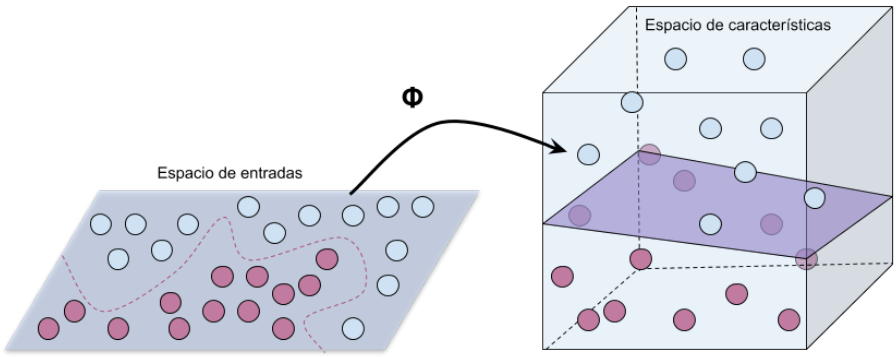


Figura 3.8: Kernel Trick.En el espacio de las características las clases no son linealmente separables, sin embargo se observa que a través de un tranformación a un espacio de mayor dimensionalidad se lograr la separación

3.3.2. Redes Neuronales

El primer modelo de neurona artificial fue propuesto por Warren McCulloch y Walter Pitts en 1943 (Kohonen)[27], conocido como el modelo de McCulloch & Pitts

(MCP). En la literatura de redes neuronales artificiales, el perceptron es el modelo matemático de una neurona más estudiado. Este fue introducido por Frank Rosenblatt en 1957 (Rosenblatt)[56].

Las redes neuronales artificiales son un modelo computacional donde un conjunto de unidades llamadas neuronas (Goodfellow y col.)[18] se encuentran interconectadas entre sí, compartiendo información entre ellas. La información de entrada atravesará a la red, sufriendo transformaciones a su paso y produciendo valores de salida. Cada neurona recibe información ya sea los datos de entrada o salidas de otras neuronas conectada con ella, este información es pesada dentro de la neurona mediante una multiplicación de pesos que son aprendidos por la misma, también puede existir una función transformadora a la salida de la neurona conocida como *función de activación* que permite que la red pueda resolver problemas más complejos. La figura 3.9 muestra algunas de las funciones de activación más usadas en la tarea de clasificación de texto

Función	Ecuación	Gráfica
Sigmoide	$f(x)=1/(1+e^{-x})$	
ReLu	$f(x)=max(x,0)$	
Tangente hiperbólica	$f(x)=(1-e^{-2x})/(1+e^{-2x})$	

Figura 3.9: Funciones de activación. Estas se utilizan para dar una 'no linealidad' al modelo y que la red sea capaz de resolver problemas más complejos. Si todas las funciones de activación fueran lineales, la red resultante sería equivalente a una red sin capas ocultas

En al Figura 3.10 puede observarse la estructura de una red neuronal simple con

dos capas ocultas

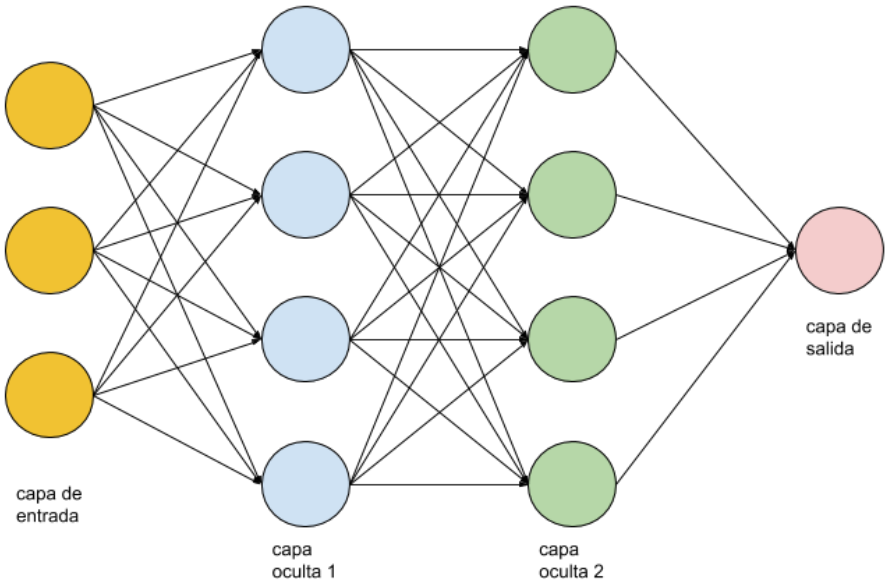


Figura 3.10: Modelo de una red neuronal simple.Se observa el conjunto de neuronas que se encuentran conectadas entre sí, se ejemplifica una red con dos estados ocultos con tres neuronas de entrada y una única salida.

Como todo modelo de aprendizaje automático, existe una función de pérdida a minimizar que permite el aprendizaje de todos los pesos de cada una de las neuronas, el método de ajuste más usado en la actualidad es conocido como retropropagación (*backpropagation en inglés*) (Rumelhart y col.)[58]. La complejidad de una red neuronal pude valorarse en función a la cantidad de capas y unidades ocultas que contenga, y el tipo tareas que sean realizadas en cada una de esas capas, así como la forma que están conectadas entre sí (Goodfellow y col.)[18]. El aprendizaje profundo, considera redes con capas ocultas más especializadas que las capas densas tales como capas de convolución, o las recurrentes. Este tipo de modelos profundos (*deep learning*) están teniendo mucha repercusión actualmente para proponer soluciones a problemas complejos tales como la segmentación de objetos en imágenes o videos (Molina Cabello y col.)[42], la descripción de escenas (Pusiol, Rincón Núñez y col.)[53, 55] o la extracción de la semántica/contexto de palabras y frases (Torres López y col.)[65].

Redes Convolucionales

Una red Convolucional CNN o ConvNet (Convolutional neuralnetwork) se caracterizan por realizar operaciones de convolucion, que es una transformación lineal de dos funciones en una tercera que puede representar la magnitud, de valor real, en la que se superponen las anteriores. Una CNN es una red multicapa especializada principalmente en extraer características de estructuras de datos en forma de *grid*, tales como imágenes lo que permite mejorar ciertas propiedades respecto de otras arquitecturas. Sobre el conjunto de datos se desliza un filtro que resume la información de cada entrada junto con su contexto espacial, es habitual el uso de diferentes filtros sobre una misma fuente de datos para enriquecer el resumen (Goodfellow y col.)[18]. La integración de la información relevante se realiza en una capa de *pooling*, posteriormente pueden definirse más capas de ya sea de convolución u otra que realice alguna tarea especializada o en su defecto a una capa densa.

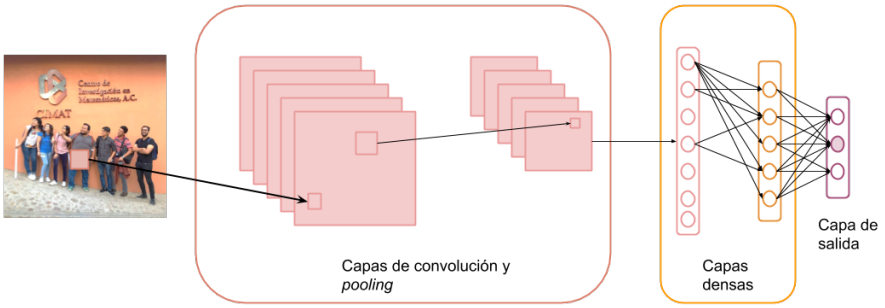


Figura 3.11: Modelo clásico de una Red Convolucional usada para la clasificación de imágenes, primero las capas de convolución extraen información espacial que es resumida con las capas de pooling, posteriormente se conectan las capas densas para realizar la clasificación

Redes Recurrentes

Las redes neuronales recurrentes (Recurrent Neural Networks en inglés RNN) son una clase de redes para analizar datos secuenciales. Aunque fueron descritas por primera vez en la década de 1980, ganaron popularidad en los últimos años debidos a los avances tecnológicos actuales que ahora permiten su implementación. Estas redes tienen la característica de retroalimentarse, es decir cada neurona se encuentra

conectada a la neurona anterior a ella y tambien a la inmediata a ella. Así que cada instante de tiempo (suele llamarse *timestep*), cada neurona recibe la entrada de la capa anteror, así como su propia salida del instante del tiempo anterior para generar su propia salida. En la Figura 3.12 observamos la estructura de una red recurrente clásica

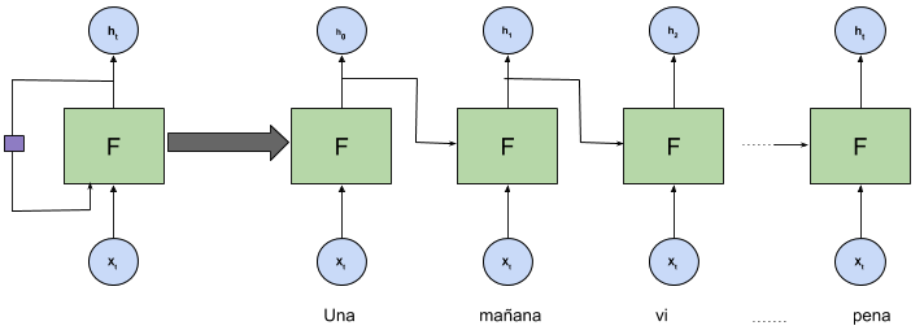


Figura 3.12: Estructura clásica RNN. Se observa el desdoble de la red, cada rezago tiene una neurona de ingreso que recibe el estado oculto anterior

Entonces podemos decir que la salida en un momento t de la red tiene la forma:

$$h_t = f(\mathbb{W}x_t + \mathbb{U}h_{t-1} + b) \tag{3.11}$$

De la ecuación 3.11 observamos que una neurona recurrente tiene cierto grado de memoria, que le permite 'recordar' información relevante de momentos anteriores. Esto también muestra una deficiencia de estas arquitecturas debida a la forma que comparten parámetros, ya que durante el entrenamiento al efectuarse el proceso de *back-propagation* es muy probable que los gradientes se desvanezcan o por el contrario 'exploten' en función del tamaño de la secuencia.

Una de la soluciones a este problema es el uso de celdas LSTM (*Long short Term Memory* en inglés) que introducen compuertas que permiten decidir la cantidad de información que pasa en cada instante del tiempo. C_i es un peso recurrente que conserva información de los estados anteriores, y en cada celda se actualiza para mantener o desechar información del pasado y agregar información de estado actual. La estructura general de una celda LSTM puede observarse en la Figura 3.13.

A cada celda entra el estado oculto anterior (h_{t-1}), el dato en el tiempo corres-

pondiente (x_t) y la memoria contenida en la celda de estado (C_{t-1}). f_t es el peso que se da a la información pasada contenida en (C_{t-1}), esta es calculada en la llamada compuerta de olvido:

$$f_t = \sigma(x_t \mathbf{U}^f + h_{t-1}\mathbf{W}^f + \mathbf{b}^f) \tag{3.12}$$

Para actualizar la memoria que viaja entre las celdas con el estado anterior, se calcula el peso (i_t) para la información ($\tilde{C}_t$) en el tiempo actual:

$$i_t = \sigma(x_t \mathbf{U}^i + h_{t-1}\mathbf{W}^i + \mathbf{b}^i) \tag{3.13}$$

$$\tilde{C}_t = \tanh(x_t \mathbf{U} + h_{t-1}\mathbf{W} + \mathbf{b}) \tag{3.14}$$

El peso recurrente es actualizado de la siguiente forma:

$$C_t = \sigma(f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t) \tag{3.15}$$

C_t es un valor númerico entre 0 y 1, por tanto indica que tan relevante es estado en la predicción. Para el cálculo del estado oculto, consideramos la información en tiempo actual, el estado oculto anterior y la información en la celda de memoria:

$$o_t = \sigma(x_t \mathbf{U}^o + h_{t-1}\mathbf{W}^o + \mathbf{b}^o) \tag{3.16}$$

$$h_t = \tanh(C_t) \cdot o_t \tag{3.17}$$

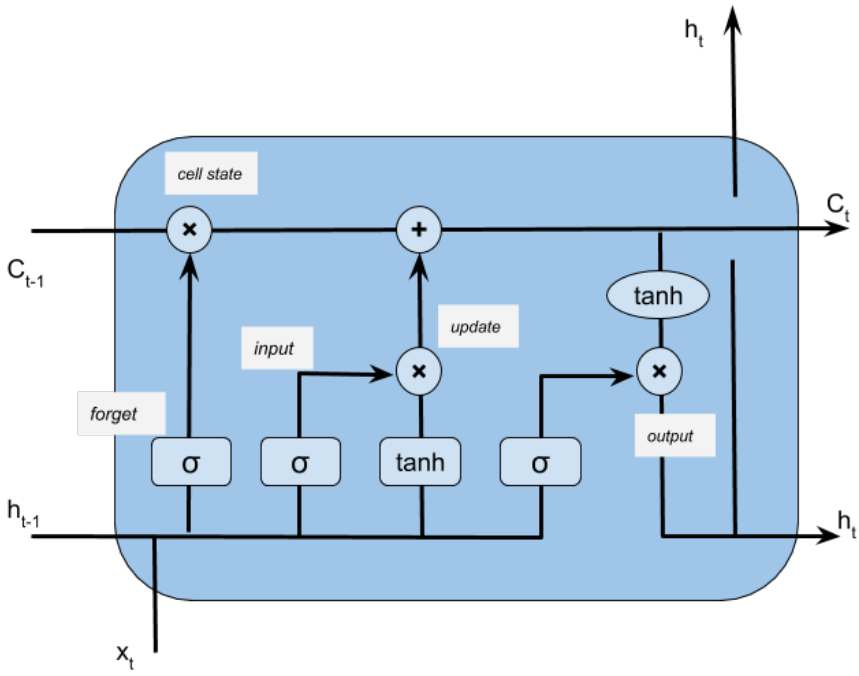


Figura 3.13: Estructura clásica LSTM. En la *cell state* se transmite la información de memoria de la red. En la compuerta de olvido (*forget*) se actualiza la relevancia de la información anterior. En las compuertas *input, update* se determina cuánto de la información actual se propaga hacia el momento actual y posteriores. Finalmente en *output* se obtiene el estado oculto actual. Adaptado de (Olah)[44]

f_t es el peso o relevancia que le damos a la información pasada, (de hecho la sigmoide de donde se obtiene se le suele llamar compuerta de olvido), mientras que i_t es el peso de la información actual $\tilde{C}_t$. Por último h_t, la salida de la compuerta se obtiene de nuestra celda de estado después de un par de activaciones.

3.3.3. Mecanismos de atención

Los mecanismos de atención permiten dar importancia a elementos de la secuencia sobre otros, y por tanto en tareas de clasificación son una manera de guiar a la red sobre qué términos de la secuencia son verdaderamente relevantes para emitir una clasificación. Por ejemplo,en la oración: 'Todo estuvo mal desde el lunes pasado que viajé a la ciudad y perdí mi equipaje, esta sin duda ha sido la peor semana desde que comencé con el nuevo empleo', si deseamos clasificar la emoción asociada no fijamos nuestra en la frase completa si no basta con secciones de la oración como

'Comenzó mal...'perdí' ..el peor dia...' , es decir que prestamos atención únicamente en los elementos que resultan relevantes para la tarea que nos encontramos realizando, esta es la intuición detrás del mecanismo de atención

En la actualidad la redes con atención tienen resultados exitosos en tareas tan diversas como *machine translation, speech recognition, image captioning* (Bahdanau y col.)[4] (Xie y col.)[70]

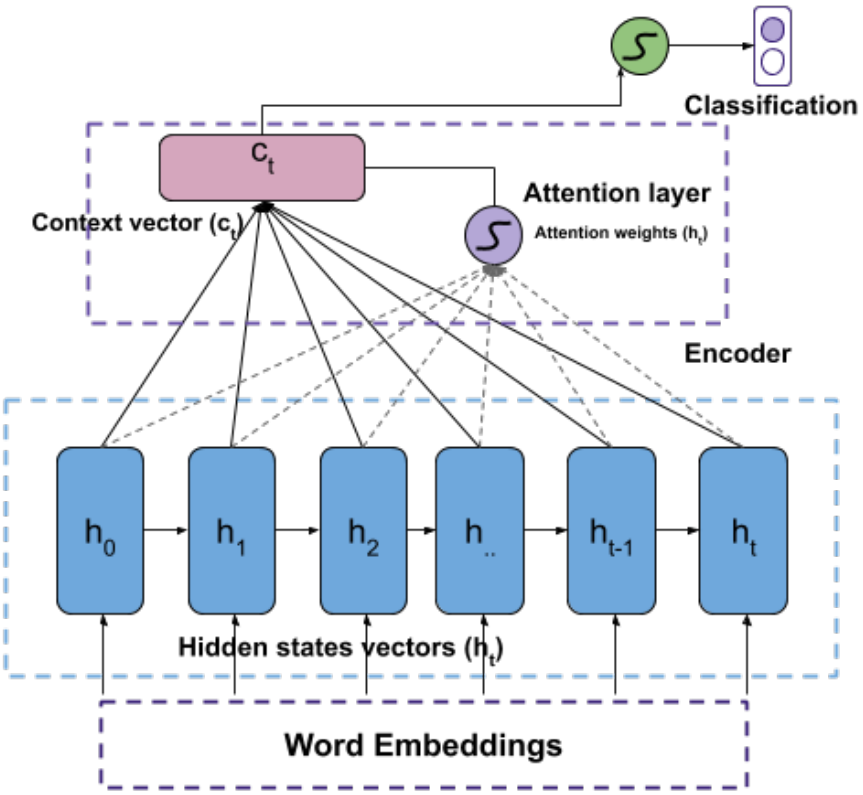


Figura 3.14: Mecanismo de atención para clasificación binaria de texto. Se usa la representación de cada *timestep* para calcular su peso (o relevancia) en el vector de contexto. Reproducido de (Weng)[68]

En la Figura 3.14 se observa cómo se incorpora el mecanismo de atención sobre una red recurrente. A cada estado oculto (h_t) se conecta a una neurona con función de activación sigmoide para obtener su peso de atención (a_t) correspondiente. Obtenemos el vector de contexto $C = \sum_t a_t h_t$ que se conecta a una capa densa para obtener la clasificación.

Self Attention

Una red recurrente aprende la representación de cada estado oculto usando los rezagos anteriores h_t, sea en el contexto de este trabajo, la representación de la palabra actual se obtiene usando el conjunto de las palabras anteriores a ella.

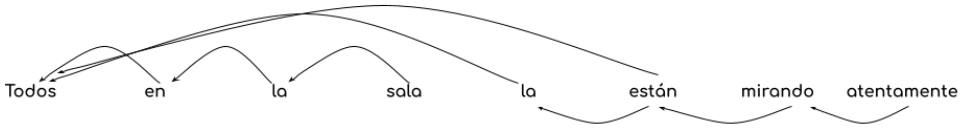


Figura 3.15: Mecanismo de *Self attention*. Las flecha indican las palabras a las que se centró la atención en el procesamiento de la secuencia. El mecanismo permite mantener la atención en palabras que no sean inmediatas a la actual

El mecanismo de *self-attention* permite que la red centre mayor atención en la palabras más correlacionadas al aprender la representación. En la figura 3.15 se muestra la intuición detrás del mecanismo.La figura 3.16 muestra una arquitectura clásica usada en la tarea de clasificación de texto que incorpora este mecanismo de atención.

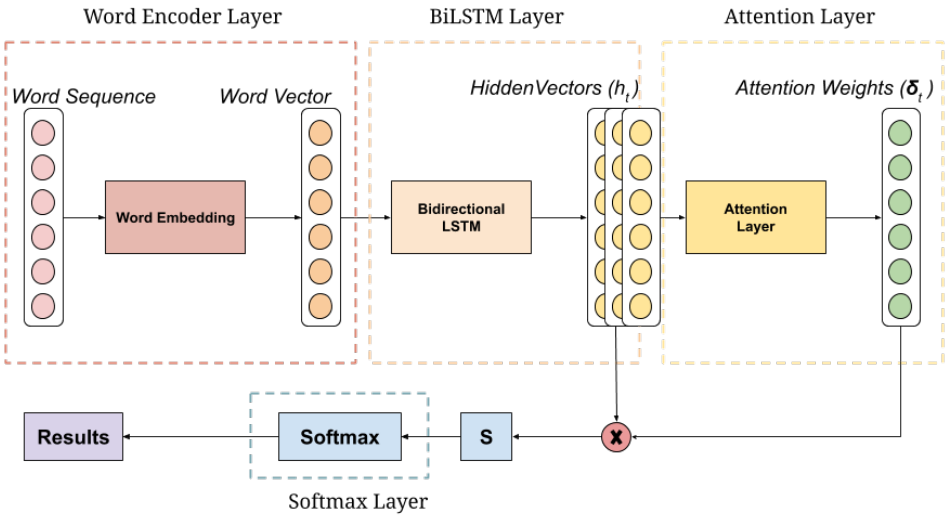


Figura 3.16: Estructura de modelo de de BiLSTM self-attention. La estructura tiene 4 módulos a) `encoder` donde se obtienen las representaciones de las palabras b)`BiLSTM` donde se procesan las representaciones vectoriales de las palabras c)`Attention Layer` donde se obtiene los pesos para estado oculto d)`Softmax` donde se asignan probabilidades a partir del vector de contexto S (Zhou y col.)[74]

Observamos que la salida de la `BiLSTM` es la colección de vectores h_t que representan el estado oculto de la red, los cuales son las entrada de la capa de atención que cuya salida es un vector de pesos $\boldsymbol{\delta_t}$. que multiplica entrada por entrada a los estados ocultos para obtener el vector $\mathbf{s}$ (Vector de contexto) que es el resumen pesado de los estados ocultos que son inputs para una capa densa y obtener la clasificación de la secuencia.

Se describe ahora el conjunto de operaciones que permiten obtener el vector de pesos $\boldsymbol{\delta_t}$.

$$u_t = \tanh(W_w h_t + b_w) \tag{3.18}$$

$$\delta_t = \frac{\exp u_t^T u_w}{\sum_t \exp u_t^T u_w} \tag{3.19}$$

$$s = \sum_t \delta_t h_t \tag{3.20}$$

Como se muestra en 3.18 primeramente el estado oculto h_t ingresa a una red neuronal simple, donde se obtiene una nueva representación u_t del mismo. Vemos en 3.19 el cálculo del peso que representa la importancia del estado oculto h_t representado por u_t con la función softmax , también puede notarse que se aprende un vector de contexto u_w que tiene como objetivo juzgar que tan importante son las palabras dentro de la oración. Este vector u_w se inicializa de forma aleatoria y se aprende conjuntamente en el entrenamiento. Finalmente en 3.20 vemos que se realiza el pesaje de los estados ocultos con los pesos calculados con anterioridad.

Capítulo 4

Metodología

En este capítulo se describe la metodología que dirige este trabajo. Primeramente en la Sección 4.1 damos contexto del origen de los datos que son utilizados en los experimentos.En la Sección 4.2 describimos las referencias utilizadas para los clasificadores utilizados en este trabajo.

4.1. Descripción de los datos

Este trabajo usa los datos proporcionados para *Task-1* del *Erisk-2018*,(Losada y col.)[35] esta tarea de carácter exploratorio sobre la *detección temprana de depresión*. Para esta tarea se debía procesar secuencialmente piezas de evidencia y detectar tempranamente indicios de depresión tan pronto como sea posible. Los textos deben ser procesados en el orden cronológico que fueron creados, de esta forma los sistema que tengan un desempeño efectivo pueden ser utilizados para monitorear las interacciones de los usuarios en blogs, redes sociales u otro tipo de dato *online*.

Los colectores del conjunto de datos (Losada y col.)[33] escogieron a *Reddit* sobre otras comunidades digitales (e.g. Twitter, MTC's A Thin line), principalmente considerando cuatro factores : 1) extensión y calidad de las fuentes de información, 2) Disponibilidad de un historial lo suficientemente largo de los individuos que formen parte de la colección, 3) El grado de dificultad de distinguir los casos de depresión de los que no, 4) Los términos y condiciones para la redistribución del conjunto de

datos, punto importante para hacer accesible el conjunto de datos a más personas.

Reddit es una plataforma abierta fundada 2005 en en la que los miembros de su comunidad (*redditors*) pueden enviar contenido (publicaciones, comentarios, vínculos a otros sitios), votar en publicaciones. El contenido dentro la plataforma se encuentra organizado en diferentes áreas de interés, entre los que se encuentran diferentes condiciones médicas, tales como depresión y anorexia. *Reddit* tiene una gran comunidad de usuarios y muchos de ellos tienen un gran historial de publicaciones en la plataforma (muchos de ellos, historial que cubre una gran cantidad de años). *Reddit* permite el uso de su contenido con fines de investigación.

Para obtener el conjunto de sujetos que tienen o tuvieron depresión, los autores optaron por un método manual, ya que era imposible aplicar las encuestas clásicas para detección de depresión. Con búsquedas en el portal se detectaron a individuos que hayan expresado explícitamente diagnóstico de depresión (e.g. *I was diagnosed with depression*) y una posterior revisión del histórico de escritos del usuario, la selección fue estricta en el cuanto la declaración de diagnóstico de depresión, ya que frases tales como *I think I have depression* o *I am depressed* no se consideraron como expresiones explicitas del diagnóstico. Solo fueron considerados dentro del grupo de depresión a aquellos *redditors* que clara y explícitamente hicieron mención de su diagnóstico

Por limitaciones del API de *reddit* de cada usuario seleccionado fueron obtenidos a lo más 2000 envíos (1000 publicaciones, 1000 comentarios). Es decir es la máxima cantidad de publicaciones que los usuarios más activos tendrán. Las publicaciones son del tipo texto y pueden pertenecer a cualquiera de los temas en los que el usuario suele participar. Los autores organizaron los envíos por usuario en orden cronológico permitiendo el análisis de la diferencia del lenguaje entre el grupo con depresión y el control, además de la posibilidad de analizar la evolución de la escritura del grupo de interés. Del grupo de depresión se elimina la publicación donde los usuarios manifestaron explícitamente su condición.

El grupo de control fue seleccionado aleatoriamente entre los miembros de la comunidad, incluyendo una cantidad de miembros que participaban activamente en el foro de depresión, pero que no había indicios de que la padecieran o que se les ha-

ya diagnosticado. Muchos de estos casos, se trata de personas que tiene un familiar cercano que sufre o sufrió depresión en algún momento, y por tanto suelen hablar de depresión con frecuencia.

El conjunto de datos fue dividido en un conjunto de archivos XML, un archivo por usuario. Cada XML contiene el histórico de los envíos de un solo usuario. Cada envio contiene únicamente su titulo, el contenido y la fecha en la que se creo. No son proporcionados más metadatos. La estructura de los XML se muestra a continuación:

```
<INDIVIDUAL>
<ID>test_subject96</ID>
<WRITING>
<TITLE> Comcast Burned </TITLE>
<DATE> 2015-04-27 18:03:50 </DATE>
<INFO> reddit post </INFO>
<TEXT> It´s Horrible </TEXT>
</WRITING>
</INDIVIDUAL>
```

cada bloque `<INDIVIDUAL>` contiene el conjunto de entradas `<WRITING>` pertenecientes al individuo que se le identifica por `<ID>`. Las etiquetas `<TITLE>`,`<DATE>`, `<INFO>` contienen información adicional de los escritos

4.2. Clasificadores

4.2.1. Baselines

Para tener un punto de comparación del desempeño de los diferentes clasificadores, usamos como *baseline*, clasificadores 'clásicos' con desempeños que se encuentran en el estado del arte. Para la representación del texto usaremos la representación de *Bolsa de palabras* y la representación *TFIDF*, como clasificador una maquina de soporte vectorial (SVM) con un kernel lineal.

4.2.2. Redes convolucionales

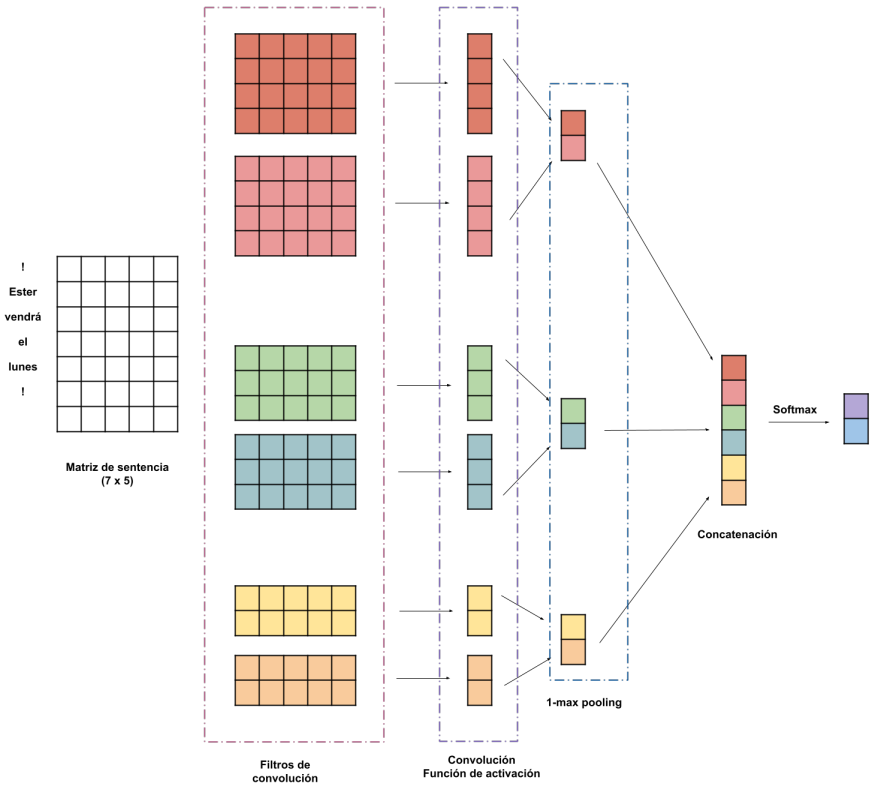


Figura 4.1: Arquitectura de red convolucional propuesta por (Kim)[25] para la clasificación d texto. Cada frase es procesada por filtros de diferente tamaño, las representaciones son resumidas en la capa de *pooling*, posteriormente concatenadas para ser conectadas a una capa densa para obtener finalmente la clasificación mediante una función sigmoide

En los últimos años, dentro del campo del procesamiento del lenguaje natural, se han obtenido buenos resultados en la tarea de clasificación de texto usando redes convolucionales bastante simples. A partir de la arquitectura propuesta por (Kim)[25], basado en con una red convolucional poco profunda donde las palabras están vectorizados por word2vec , se han propuestos arquitecturas más sofisticadas de redes convolucionales. En la figura 4.1 puede observarse los detalles de esta arquitectura

4.2.3. Redes recurrentes

Es clara la naturaleza secuencial del lenguaje, donde cada palabra está relacionada con las que se comunican previamente y las siguientes a ella. Por tanto es natural usar

redes recurrentes para el trabajo con el texto.

Anteriormente se ha expuesto el buen desempeño de los modelos que usan redes recurrentes para el análisis de datos de naturaleza secuencial, tal como lo son los textos. También ha sido expuesto cómo los mecanismos de atención potencializan el desempeño de estas arquitecturas. Muchos trabajos han usados diferentes tipos de redes recurrentes en la tarea de clasificación de texto, usamos como trabajo de referencia el trabajo de (Zhou y col.)[74], donde proponen una arquitectura con una red Bi-LSTM dotada de un mecanismo de atención. La Figura 4.2 muestra la arquitectura de la red propuesta. Podemos notar las mejoras de esta arquitectura con respecto a una LSTM ordinaria, ya que esta cuenta con dos LSTM que aprenden los pesos en direcciones contrarias, lo que en la práctica ha mostrado lograr un mejor entrenamiento. También vemos que se integra el mecanismo de atención para ayudar a discernir la relación entre los términos y su contexto.

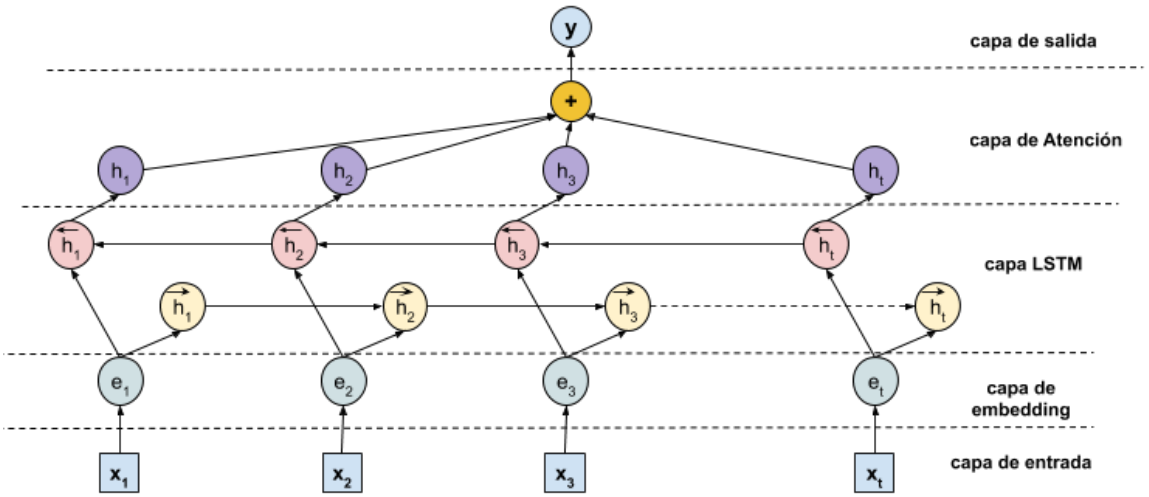


Figura 4.2: BiLSTM con Mecanismo de atención para la clasificación de texto. La arquitectura incorpora una LSTM bidireccional que aprende los estados ocultos en los sentidos del texto. Los estados ocultos entran a capas densas que calculan la atención para cada uno de ellos. Finalmente el promedio ponderado con los pesos de atención entran a una capa densa para obtener la clasificación (Zhou y col.)[74]

4.2.4. Modelos de fusión

Dado que los clasificadores 'más sencillos' tales como las máquinas de soporte vectorial que usan representaciones vectoriales como bolsa de palabras o TFIDF suelen tener resultados competitivos en tareas de clasificación de texto, una práctica que gana popularidad es la fusión de características (Basly y col.)[5] . En la Figura 4.3 se muestra cómo se obtiene esta fusión. Primeramente se debe contar con una red neuronal preentrenada en la tarea, de la cual se obtiene la representación de los datos previa a la capa de clasificación, por otra parte se obtiene una representación vectorial sencilla del mismo conjunto de datos. Ambos conjuntos de características se fusionan a nivel ejemplo, donde cada uno de ellos cuenta con las características obtenidas de la red y la representación vectorial. El conjunto de fusión se usa para entrenar un modelo de aprendizaje automático.

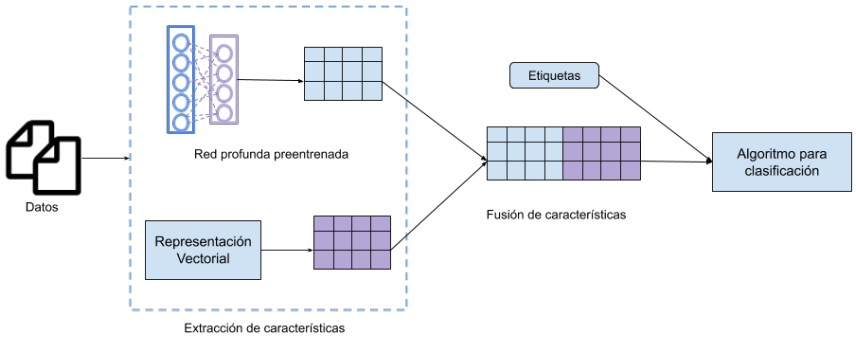


Figura 4.3: Modelos de fusión de características. Los datos pasan por dos procesos de extracción de características, primero con una red preentrenada con el conjunto de datos que extraiga las características previas a la capa final y una segunda extracción de características a través de una representación vectorial clásica. Las representaciones se fusionan y se usan para alimentar un algoritmo de clasificación

4.3. Ajustes y configuraciones de los experimentos

4.3.1. Preprocesamiento

Los XML fueron procesados a una estructura de archivos que permita un manejo fluido de la información. Para el procesamiento se escribe un *parser* en código Python 3.x. Este procesa el conjunto de XML's y concentra la información dentro de un

diccionario, que es estructura de datos en Python que permite el fácil acceso de la información para alimentar los modelos de interés en este trabajo. Para el proceso de ingesta y transformación de los datos se usan las siguientes librerias:

1. **XML, BeautifulSoup**: Para procesamiento de los XML
2. **NLTK, SckitLearn**: Funciones relativas al procesamiento de texto
3. **Pickle**: Serialización de datos

Aunque se procesa por separado el conjunto de datos de entrenamiento y el de prueba, se realiza el mismo procesamiento a ambos conjuntos.

Con respecto al preprocesamiento, se realizan los siguientes tratamientos a los textos:

1. Se eliminan todas las secuencias no alfanuméricas, tales como hipervínculos, emojis, etc
2. Todas las palabras son convertidas a minúsculas
3. Se conservan las llamadas *stopwords*
4. No se realiza proceso de *stemming*

La figura 4.4 muestra el flujo de procesamiento de los datos

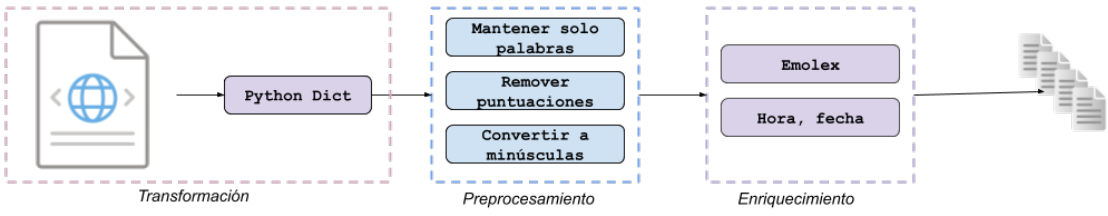


Figura 4.4: Procesamiento del conjunto de datos. `a)Transformación:` Los archivos XML son procesados a diccionarios para una manipulación ágil. `b)Preprocesado:` Limpieza mínima de los datos eliminando caracteres numéricos, símbolos y puntuaciones y normalizando a minúsculas. `c)Enriquecimiento:` Agregamos información adicional que pueda ser útil para la tarea, tales como emociones asociadas a la palabra y marcas de tiempo relativas al momento que se realiza la publicación.

Etiquetado de sentimientos: Emolex

Emolex Es un lexicón creado por Saif M. Mohammad para el análisis de sentimientos (Mohammad y col.)[41], que permite asociar palabras con 8 emociones (*anger, fear, anticipation, trust, surprise, sadness, joy, and disgust*) y que cuenta con un vocabulario de 16,862 palabrás en inglés. Este diccionario de palabras se obtuvo de tuits que tuvieran hashtags con emociones (e.g. Hola #happy #anger), las cuales sirvieron para asignar las emociones correspondientes a cada palabra.

En la literatura se ha demostrado de que el discurso de las personas con depresión tienen una fuerte carga de emociones negativas (Rottenberg)[57], por tanto el el texto es enriquecido con los sentimientos asociados con la hipótesis de que el desempeño de los clasificadores puede ser mejorado.

Con respecto al etiquetado de emociones, *Emolex* fue utilizado de dos maneras:

1. Las palabras en el texto que estuvieran en el diccionario de *emolex* se les agrega la emoción o emociones asociadas a ella, las palabras que no se encontrarán en el diccionario del lexicón fueron conservadas, es decir que si la publicación original es: *'I think you´re overthinking the situation don´t worry he´ll be ok* esta se cambia por: *'I think you´re overthinking fear anticipation the situation don´t worry fear sadness he´ll be trust'*

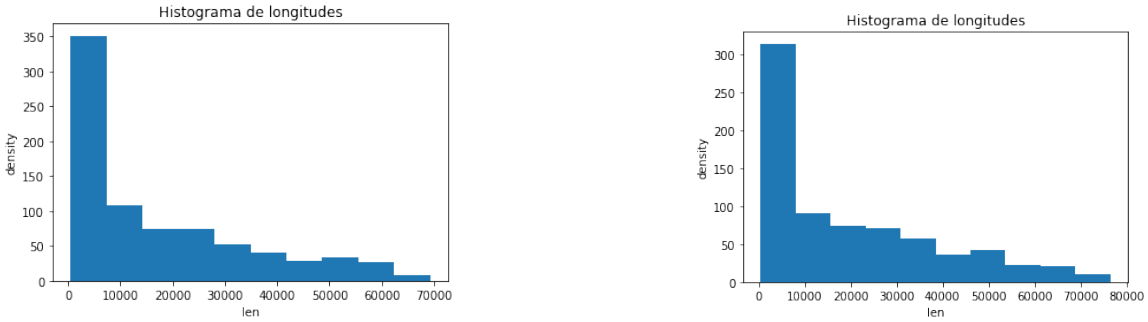
2. Las palabras en el texto que estuvieran en el diccionario de *emolex* son sustituidas por las emociones asociadas y eliminar las que no se encuentren en él, es decir que si la publicación original es: *'I think you´re overthinking the situation don´t worry he´ll be ok'* esta se cambia por: *'fear sadness trust'*

4.3.2. Análisis exploratorio

Tabla 4.1: Características de los datos

	Depressed	Control
Sujetos	214	1493
Num. de entradas	99114	963674
Media entradas por subj.	361.90	638.20
Media de días de actividad	578.30	625.30
Media palabras por entrada	27.40	36.70

En la tabla 4.1, se presentan algunos estadísticos de interés del conjunto de datos. El tiempo promedio entre el primer y ultimo escrito (publicación o comentario) de un usuario es de alrededor de 18 meses. Los escritos por usuario tienen un alta variabilidad en cuestión a su longitud de palabras, encontramos por un lado escritos 'particularmente cortos' con menos de 100 palabras en toda la colección, y por escritos de usuarios con miles de palabras, en promedio los textos por usuario tienen 10225 palabras. A nivel escrito, las entradas van entre una y 8000 palabras por envió. El promedio de palabras de cada envió, después del preproceso ronda alrededor de 30,la figura 4.5 muestra la distribución del largo de palabras de los escritos. El conjunto de datos se encuentra separado en *entrenamiento* y *prueba*, en proporciones 3:2 tanto para el conjunto de personas con depresión como el de control. La tarea original para la que se compilo este conjunto datos es la detección temprana de la depresión, así que los datos en su versión original se encuentran divididos en diez partes (*chunks*) que contienen aproximadamente la misma cantidad de escritos por usuario



(a) Nivel usuario

(b) Nivel post

Figura 4.5: Distribución de cantidad de palabras de los textos. Al observar la distribución del largo de palabras por los textos, se observa que a pesar de existir escritos de longitud extensa, el mayor peso recae sobre escritos de longitud menor. Las entradas en promedio tienen 30 palabras, mientras que los escritos por usuario ronda en promedio 10225

Consideramos dos formas 'alimentar'a nuestros clasificadores con los datos:

1. Concatenar todo el texto de un usuario en una sola entrada, es decir que nuestro clasificador tendrá que aprender con la totalidad del texto en un sola vista. En esta versión el número de ejemplos de los cuales el clasificador aprende corresponde a la cantidad de usuarios que se encuentran en el conjunto de entrenamiento. En esta versión, la cantidad de observaciones en el conjunto de entrenamiento y de prueba es de 887 casos (135 positivos-752 control) y 820 casos (79 positivos- 741 control), respectivamente. Preparamos dos conjuntos de datos, uno donde al texto solo se le trata con el procesamiento descrito en 4.3.1 y en el otro, además del procesamiento se etiquetan emociones usando emolex. Estos conjuntos de datos se codifican como **RU** y **SU** respectivamente. Donde R (raw) indica que el texto solo fue preprocesado y S (sentiment) para indicar que se enriquece con emolex, U indica que el texto se divide a nivel usuario.

2. Presentar cada entrada como un ejemplo al clasificador, es decir que nuestro entrenamiento se hace a nivel *post*. Para las etiquetas que corresponden al post, repetimos la etiqueta del usuario que lo realiza. En esta versión, la cantidad de observaciones en el conjunto de entrenamiento y de prueba es de 531397 casos (49557 positivos-481837 control) y 544447 casos (40665 positivos- 503782

control), respectivamente. También tenemos dos versiones, **(RP)** donde al texto solo se le trata con el procesamiento descrito en 4.3.1 y en **(SP)** además del procesamiento se etiquetan emociones usando emolex. La codificación incluye P para indicar que la división es a nivel post.

4.3.3. Ajuste y detalles de experimentos

Selección de Word Embbedings

En todos los experimentos usamos *embbeddings* `Glove` preentrenados para la representación vectorial de las palabras. [1]. Estos fueron entrenados con textos de Twitter, con cerca de 2 billones de tuits en inglés, el vocabulario contiene cerca de 1.2 millones de palabras. Escogimos los embedding de dimensión 100 para cada palabra. (Pennington y col.)[51]

Ajustes Baselines

Para cada una de las diferentes versiones de datos, entrenamos con la representación de bolsa de palabras y la representación TFIDF una máquina de soporte vectorial usando un kernel lineal. En en el ajuste del modelo consideramos únicamente tomar los 3000 términos relevantes dentro del vocabulario.

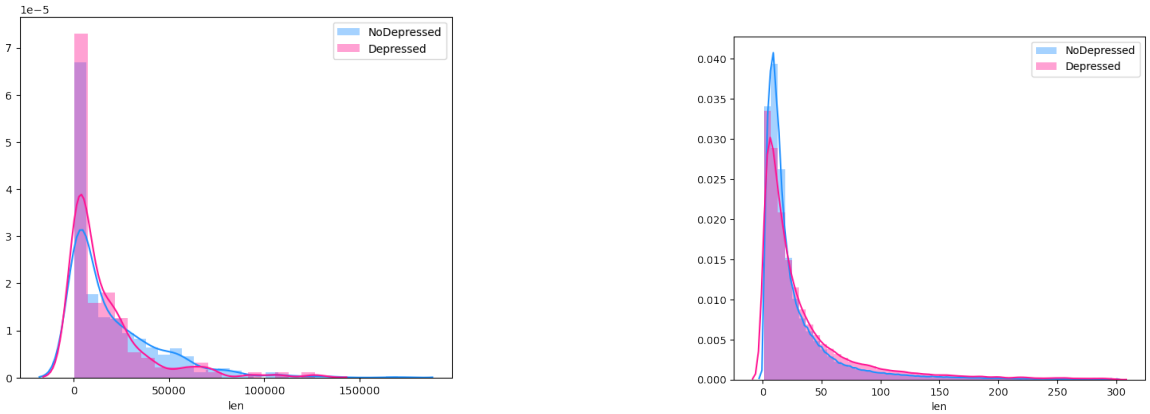
Selección de largo de secuencia

Un hiperparámetro importante en el diseño de modelos con redes neuronales, es definir es el largo de la secuencia de texto de palabras que será recibidas en la capa de entrada. En la figura 4.6 se muestra la distribución del largo de secuencias de texto del conjunto de entrenamiento obtenido de los datos a nivel usuario(4.6a) y a nivel de post (4.6a),presenta las distribuciones para el grupo de interés y el grupo de control.

Para el conjunto de datos a nivel usuario la media de largo de secuencia es 38915.4 palabras con un rango entre las 73 y 170053 palabras. Para el conjunto a nivel post, la media es de 39.4 palabras por post con un rango entre 1 y 8316 palabras. Con respecto

[1]Disponibles en `https://nlp.stanford.edu/projects/glove/`

a la distribución del largo de secuencia, no se observa una diferencia entre el grupo de interés y el grupo de control. Ahora, es evidente la alta variabilidad en la extensión de las secuencias. Lo ideal es entrenar con la totalidad del texto disponible y no sacrificar información, es decir tomar como largo de secuencia la longitud mayor observada en el conjunto de datos. Pero esto implicaría que una gran cantidad de ejemplos tendrían que ser rellenados con un *padding* (ceros) para tener el tamaño deseado de secuencia, y por tanto procesar gran cantidad de datos sin contenido lo que es un gran costo computacional, estudios como (Dwarampudi y col.)[14] revelan que el *padding* puede tener efectos artificiales en el entrenamiento de la red. En el caso de las versiones de datos a nivel usuario (**Ru, Su**), después de experimentar con diferentes longitudes y considerar los costos computacionales así como el desempeño del clasificador, se fija una secuencia de entrada de tamaño 10000 tokens. Esta secuencia cubre en totalidad al 55 % de los escritos. Para las versiones a nivel post, fijamos el largo de secuencia a 61 tokens cubriendo con ello el 95 % del total de los casos. En todos los casos si la secuencia era menor a la longitud fijada se agregaba un padding al final hasta completar la secuencia, pero si la secuencia es mayor que largo fijado, se tomaba el largo de secuencia con los últimos elementos.



(a) Nivel usuario. **(b)** Nivel post.

Figura 4.6: Distribución de largo de textos entre el grupo de interés y el grupo de control. En ambos casos, no se observa diferencia entre los grupos y se encuentran sesgadas hacia la izquierda, por tanto tomar una secuencia menor no sacrifica mucha información

Ajustes para redes convolucionales

En la Figura 4.7 se muestra la conectividad de las redes convolucionales usadas en los experimentos. Previamente se ha discutido el ajuste que se da la dimensión de las secuencias de entrada y el uso de *embeddings* preentrenados en la capa correspondiente. Para la capa de convolución 1D se fija un kernel de dimensión 4, con un *padding* ajustado en ambos lados de las secuencia y usando como función de activación a la función *ReLu*(3.9), se fijan 32 filtros de convolución. Las capas de *max-pooling* tienen un filtro de dimensión 2. Se fija 0.30 para el porcentaje de desconexión de la capa de *Drop-out* par evitar el sobreajuste. La capa densa se fija a 256 unidades con función de activación *relu*. La función de pérdida usada es la *binary crossentropy* y usamos como algoritmo de optimización a *adam*, que es un algoritmo de gradiente estocástico con estimación de momentos adaptativa (Kingma y col.)[26].

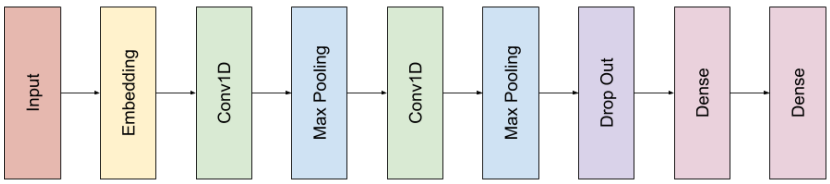


Figura 4.7: Estructura de CNN. Usamos una configuración con doble capa de convolución conectada a una capa de *max pooling* para comprimir la información, posteriormente una capa de *drop-out* antes de ser conectada a las capas densas

Ajustes para redes recurrentes

El diseño de la red BiLSTM puede consultarse en la Figura 4.8. Se ajustó la cantidad de unidades de las celdas LSTM a 100 unidades. Manteniendo la dimensión de los embeddings de entrada

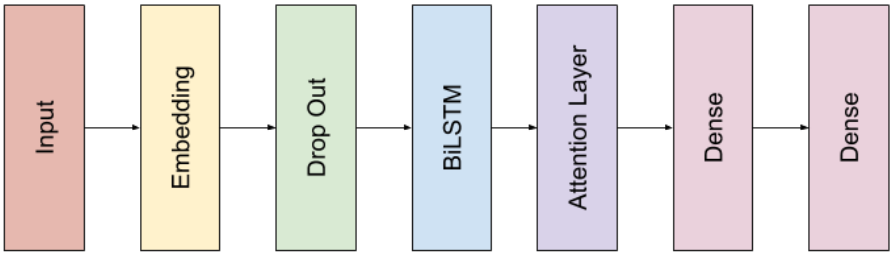


Figura 4.8: Estructura de BiLSTM con mecanismo de atención. Previo a la celda Bidireccional la celda *Drop-out* desconecta algunas entradas. El mecanismo de atención se conecta a la celda bidireccional para ingresar a la densa

Entrenamiento

Todos los experimentos usaron *batchsize* de tamaño 16, como conjunto de validación se toma el 30 % de los datos de entrenamiento. El número de épocas de entrenamiento fue definido a través de *callbacks*. El código de todos experimentos fue escrito en Python usando las librerias `Sckitlearn` y `tensorflow` Todas las redes fueron escritas y entrenadas en la plataforma `Google Colaboratory` (Google)[19] .

4.3.4. Métricas

A continuación se describen las métricas usadas para la comparación de los modelos :

- Accuracy

  El *accuracy* se define como el número de predicciones correctas entre el número de predicciones totales, o dicho de otra forma:

$$\text{Accuracy} = \frac{\text{Verdaderos positivos} + \text{Verdaderos negativos}}{\text{Total de casos}}$$

- Precisión

  La precisión se define como la proporción de predicciones positivas que fue correcta, es decir:

$$\text{Precision} = \frac{\text{Verdaderos positivos}}{\text{Total de predicciones positivas}}$$

- Recall

  El recall se define como la proporción de positivos reales que se identificó correctamente, es decir:

$$\text{Recall} = \frac{\text{Verdaderos positivos}}{\text{Total de casos positivos}}$$

- F1-score

  El F1-score se define como la media armónica entre la precisión y el recall. Muestra que tan preciso y robusto es el clasificador:

$$\text{F1 score} = 2\frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Capítulo 5

Resultados

En este capítulo describimos los resultados obtenidos en este trabajo. Primeramente en la Sección 5.1 se muestran los resultados relacionados con el análisis de emociones relevantes en el conjunto de datos, que fueron obtenidas a través del etiquetado con *Emolex*. Finalmente en la Sección 5.2 mostramos los resultados obtenidos para los diferentes clasificadores que fueron entrenados y las métricas de interés para cada uno.

5.1. Distribución de sentimientos

En la Sección 4.3.1 se presentó el uso de *Emolex* para el etiquetado de emociones. Es de interés observar si existe alguna tipo diferencia en la marca emocional de los escritos provenientes de personas con depresión con respecto al grupo de control. Con este fin, se realiza el etiquetado de emociones de la totalidad del conjunto de datos, y observamos la proporción con la que cada emoción aparece en cada grupo. En la Figura 5.1 puede observarse esta distribución.

La cuatro emociones predominantes en el conjunto de datos son *trust,anticipation,joy* y *fear*. La emoción menos representada es *disgust*. No observamos diferencia en la huella de emociones entre el grupo control y el de personas con depresión. Se esperaba observar que el conjunto de interés tenga una carga mayor sobre emociones negativas pero este comportamiento no fue observado lo que indica la necesidad de tener un

método de aprendizaje máquina que lo descubra.

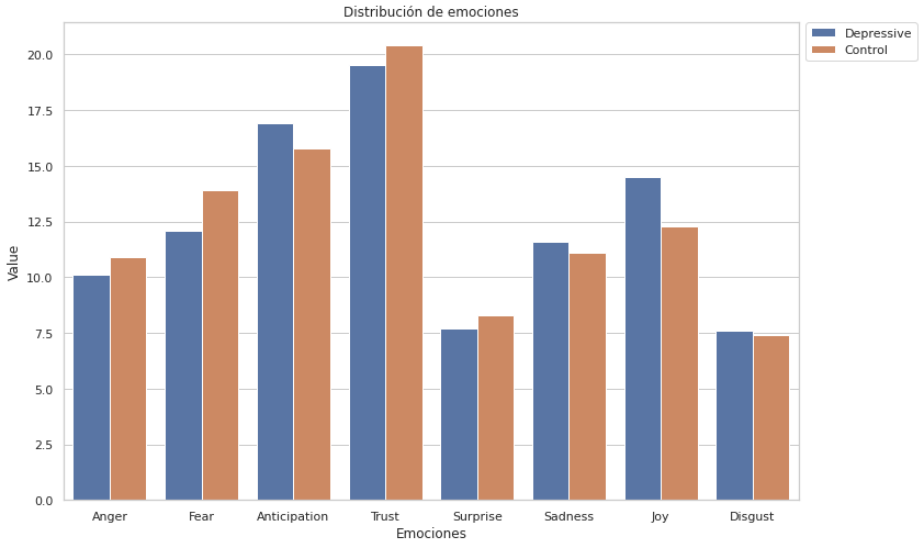


Figura 5.1: Distribución de emociones entre clases. No se observa diferencia entre la proporción de las emociones en los grupos. Con respecto a las emociones predominan confianza (*trust*), anticipación *(anticipation)*, alegria (*joy*)

5.2. Resultados de los clasificadores

En esta Sección presentamos los resultados obtenidos en la tarea de detección de depresión con los diferentes clasificadores descritos en el Cápitulo 4. Se reportan las métricas descritas en la Sección 4.3.4.

5.2.1. Baselines

Se presenta las métricas sobre el conjunto de prueba en las versiones a nivel usuario, del conjunto de datos (**RU,SU**)[1] obtenidos con los clasificadores usados como *baseline* como en la Sección 4.3.3 fue indicado. Estos resultados pueden consultarse en la Tabla 5.2.1

Aunque no parece ver una diferencia notable en el desempeño de los clasificadores, podemos tomar como valor base el obtenido por el clasificador que usa la representa-

[1]Como se indica en la sección 4.3.2, codificamos los cuatro conjuntos de datos con la letra R para indicar que el texto solo fue preprocesado, S si fue enriquecido con *emolex*, y las letras U y P para indicar si se encuentran a nivel usuario o post

ción **TFIDF** donde al texto fueron agregadas las emociones con *Emolex*. Recordemos que hay un desbalanceo entre la clase de interés y el grupo de control (1:8 aprox.), vemos que este afecta al desempeño del clasificador, ya que el resultado se sesga hacia la clase mayoritaria. Por eso tal como ya se indicó en el Apartado 4.3.4 usamos la métrica F1 y no el *accuracy* como una medida de desempeño discriminatoria. El modelo con la representación TFIDF que es enriquecido con *Emolex* tiene el mejor resultado con un F1 de 0.514. Al comparar el desempeño entre versiones de datos similares, vemos mejora en clasificadores que usan *Emolex* que los que no y en los que usan la representación TFIDF sobre la bolsa de palabras (BoW)

La Tabla 5.2 presenta el resultado de los clasificadores ahora entrenando a nivel post.(**RP,SP**). El *accuracy* es menor al compararse al conjunto de datos a nivel usuario, así que el sesgo del clasificador hacia la clase mayoritaria es menor. Vemos también que el F1 es menor. En particular observando el resto de las métricas, la precisión decae en comparación al conjunto en nivel usuario. Esto lo interpretamos que ahora al clasificador es menos acertado en sus predicciones de clase de interés.

Tabla 5.1: Desempeño en conjunto de prueba de clasificadores SVM. Nivel Usuario

	RU		**SU**	
	BoW	TFIDF	BoW	TFIDF
Acc	0.875	0.875	0.878	**0.875**
Precission	0.382	0.410	0.389	**0.412**
Recall	0.481	0.670	0.468	**0.683**
F1 Score	0.426	0.509	0.425	**0.514**

El valor de accuracy alto se debe al desbalanceo de clases, el clasificador sesga hacia la clase mayoritaria. Tomamos con baseline TFIDF

Tabla 5.2: Desempeño en conjunto de prueba de clasificadores SVM. Nivel Post

	RP		**SP**	
	BoW	TFIDF	BoW	TFIDF*
Acc	0.672	0.658	0.674	**0.658**
Precission	0.124	0.128	0.125	**0.127**
Recall	0.565	0.616	0.564	**0.618**
F1 Score	0.204	0.212	0.205	**0.212**

5.2.2. Redes convolucionales

La Tabla 5.3 muestra los resultados de desempeño de las redes convolucionales entrenadas con las diferentes versiones de los datos en la tarea de detección de depresión. Vemos que las versiones de datos a nivel usuario tienen valores altos de *accuracy* **(RU:0.885, SU:0.875)** este efecto podemos atribuirlo al desbalance de clases, lo que hace que el clasificador tienda a clasificar los ejemplos como la clase mayoritaria, aunque utilizamos pesos para las clases durante el entrenamiento este efecto no pudo ser evitado completamente. Con respecto al F1, el conjunto **SU** obtuvo 0.479. El conjunto a nivel post, obtuvo resultados menores en comparación con al nivel usuario. De las métricas *precission y recall* observamos que en los conjuntos a nivel post, el clasificador se equivoca con mayor frecuencia en sus predicciones positivas pero tiene mayor sensibilidad para identificar los casos positivos. En este conjunto, el mejor resultado para el F1 se obtuvo del conjunto (**SP:0.210**)

Tabla 5.3: Resultados de Experimentos CNN

	RU	**SU**	**RP**	**SP**
Acc	0.885	0.875	0.523	0.649
Precission	0.427	0.401	0.109	0.126
Recall	0.519	0.594	0.752	0.624
F1 Score	0.465	0.479	0.191	0.210

Se observa un mejor desempeño en los modelos que son enriquecidos con emociones sobre los que no, el rendimiento general de la arquitectura es semejante al obtenido con el clasificador SVM

5.2.3. Redes recurrentes

La Tabla 5.4 muestra los resultados con la redes BiLSTM dotadas con mecanismo de atención en la tarea de detección de depresión. Con respecto a la métrica de interés F1-score los mejores resultados son 0.535 y 0.291 para el conjunto de datos que presenta etiquetado de emociones (**SU,SP**), estos resultados resultan mejores que los obtenidos con la máquina de soporte vectorial y la red convolucional. Con respecto al comportamiento de los conjuntos, se observa patrones semejantes a los obtenidos con la redes convolucionales. El conjunto a nivel usuario presenta mejores resultado

que el a nivel post y se nota de nuevo el efecto de la clase mayoritaria. El conjunto a nivel post presenta mayor dificultad para acertar las predicciones positivas. También se observa una mejora en los resultados en los conjuntos de datos en los que se agrega las emociones con respecto a los que no.

Tabla 5.4: Resultados de Experimentos Bi-LSTM-Att

	RU	**SU**	**RP**	**SP**
Acc	0.822	0.873	0.499	**0.508**
Precission	0.324	0.413	0.170	**0.175**
Recall	0.785	0.759	0.854	**0.867**
F1 Score	0.459	0.535	0.284	**0.291**

5.2.4. Modelos de fusión

En la Sección 5.2.1 se muestra que los conjuntos de datos enriquecidos con el etiquetado de emociones, en lo que se uso la representación TFIDF y la máquina de soporte vectorial como clasificador se obtuvieron los mejores resultados. También con este conjunto de datos se obtuvo los mejores resultados en la redes neuronales usadas. Ahora para explorar, la posible sinergia entre los resultados. Obtenemos del conjunto de datos (**SU**, **SP**) las características que cada una de las redes neuronales extrae de ellos y las fusionamos con la representación TFIDF. En la Tabla 5.5 se presentan los resultados de los modelos.

Tabla 5.5: Modelos de fusión de características (TFIDF+NN+SVM)

	SU		**SP**	
	CNN	BiLSTM	CNN	BiLSTM
Acc	0.843	0.861	0.499	0.517
Precission	0.404	0.431	0.170	0.178
Recall	0.888	0.881	0.854	0.872
F1 Score	0.555	0.579	0.284	0.296

En cada caso la representación TFIDF es enriquecido con las representaciones obtenidas de las redes preentrenadas, se observa una mejora en la sensibilidad de los clasificadores, es decir en su capacidad de detectar casos positivos

En general observamos resultados semejantes a los obtenidos en los baselines, la métrica F1 mejora ligeramente en todos los experimentos, es notorio que el principal efecto de la fusión de características es sobre el *recall*, es decir que se mejora la sensibilidad del clasificador para detectar casos positivos.

5.2.5. Visualización del aprendizaje arquitectura BiLSTM (SP)

En la Sección 3.3.3 se describe el mecanismo de atención, vemos que el vector de contexto s es una suma ponderada de los estados ocultos. Cada peso en $\boldsymbol{\delta}$ representa la relevancia de cada palabra en el vector de contexto y por tanto para la clasificación. Así que $\boldsymbol{\delta}$ contiene información sobre a qué sección o palabras del texto el clasificador debe dar más importancia. En el contexto de nuestros experimento tomamos el vector de atención de ejemplo y según su valor normalizado asignamos rangos de baja y alta atención ($\alpha > 0.7$). Cada vector de atención fue normalizado dividiendo cada entrada entre el máximo del vector, de forma que las magnitudes de atención entre ellas sean comparables. Los ejemplos que se presentan se obtiene de los resultados obtenidos de la arquitectura Bi-LSTM-Att para el conjunto **SP** por obtener los mejores resultados dentro de la categoría de post, esta categoría fue seleccionada sobre la nivel usuario por que las visualizaciones son más interpretables. En la Figura 5.2, se presenta con una escala de colores de azul a rojo la intensidad de la atención presentada a cada palabra de uno de los ejemplos del conjunto entrenamiento que fueron correctamente clasificados, como provenientes, de una persona con depresión.

De esto interpretamos en el ejemplo que la frase inicial *'over small issues'* fue poco relevante para la clasificación mientras que las palabras *'my'* , *'feel'*, *'anymore'* captaron la mayor atención del clasificador. El clasificador aprendió que la tarea estaba relacionada con el concepto de depresión, ya que puede observarse en la figura 5.3a que si el texto contiene la palabra *'depression'*, toda la atención se centra sobre ella ignorando prácticamente al resto del texto. En la figura 5.3b se presenta el mismo texto, solo que ahora la palabra *'depression'* fue removida de él; el clasificador aún reconoce al ejemplo como parte del grupo de interés, pero vemos que el clasificador da atención a palabras que originalmente no fueron consideradas relevantes para la

clasificación.

Ahora comparemos el efecto del etiquetado de emociones sobre la atención en las palabras, los resultados indican que el clasificador no usa las emociones como características relevantes al emitir la clasificación, pero al parecer tienen un efecto moderador en asignar relevancia a las palabras en el texto, ponderando términos de alta relevancia con de relevancia menor. En la Figura 5.4 se observa primeramente el texto sin etiquetado de emociones y con las etiquetas en negritas. Puede notarse que palabras con atención moderada disminuye su relevancia al presentarse las etiquetas de emoción.

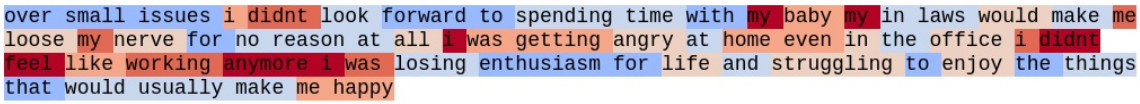
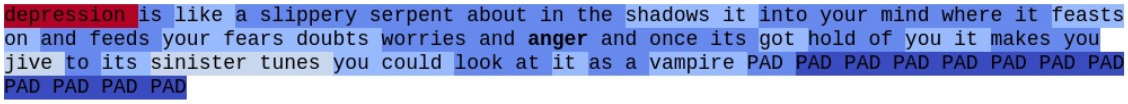
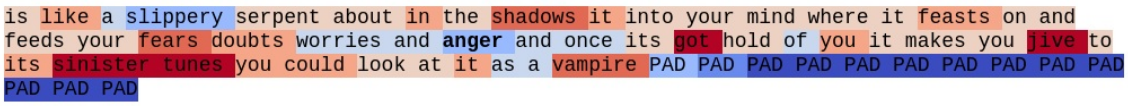


Figura 5.2: Visualización de la atención. Para esta visualización coloreamos el fondo de cada palabra con respecto a la magnitud del peso que se le da en el vector de atención. La escala de color de forma ascendente va de azul a rojo



(a) Distribución de la atención con la presencia de la palabra depresión. Notamos como el sistema concentra la mayor parte de la atención sobre la palabra 'Depressión' clasificar el texto dentro el grupo de interés



(b) Distribución de la atención sin la presencia de la palabra depresión. Removemos la palabra 'depression' del oración, el sistema ahora 'se fija' en el resto de palabras quen en presencia de la palabra 'depression' fueron ignorados

Figura 5.3: Visualización de relevancia de las palabras

for a long while been having issues with mental health i remember asking my mum one day years ago if hearing and seeing things was normal and her response still sticks with me too young and know what real mental health problems are PAD PAD PAD PAD PAD PAD PAD PAD PAD PAD PAD PAD PAD PAD PAD PAD PAD

(a) Sin etiquetado de emociones

for a long **anticipation** while been having issues with mental health i remember asking my mum **fear** one day years ago if hearing **fear** and seeing things was normal and her response still sticks with me too young **anticipation joy surprise** and know what real mental health problems are PAD PAD PAD PAD PAD PAD PAD PAD PAD PAD PAD PAD

(b) Con etiquetado de emociones

Figura 5.4: Efecto del etiquetado de emociones en la atención. Observamos que existe diferencia en el nivel de atención de palabras que parecieran no ser relevantes, este disminuye cuando el texto es enriquecido con el etiquetado de emociones

Figura 5.5: Nube de palabras relevantes para la clasificación de la clase con depresión. Resalta 'depression', 'thank', 'please', 'die' como términos relevantes. Estas palabras son percibidas como relaccionadas con la depresión

Del conjunto de posts correctamente clasificados, se toman las palabras con los mayores valores en el vector de atención para generar el vocabulario de términos relevantes para el clasificador. En la Figura 5.5 se muestra los términos relevantes de este vocabulario. Notamos que 'depression' es la palabra relevante del conjunto, también palabras como 'great', 'thank', 'help','die'. Usando las representaciones vectoriales glove de este vocabulario (33335 palabras), realizamos un análisis de tópicos usando

kmeans con 15 clústers, esta cantidad fue asignada de forma arbitraria debido al tamaño del vocabulario, puesto que buscamos tener grupos de palabras con alta relación semántica. Identificamos 5 grupos que armonizan en el contexto del problema. En la Tabla 5.6 mostramos los grupos con algunas palabras relevantes de los mismos.

1. **Ánimo,cariño y agradecimiento:** Palabras que expresan agradecimiento tales como *thank, thanks, appreciate, bless, congrats*. Tambien palabras que muestran afecto y dar ánimo como *hug,cheers,merry, nice, adorable, cutie, amen, preach, proud*.

2. **Salud,enfermedades, tratamiento:** Términos relativos a trastornos mentales (*depression, anxiety,syndrome, insomia, disorder, panic, schizophrenia*), médicos (*obesity, diabetes, diagnosis, prescription, mental,health*), tratamientos (*meds, xanas,lsd,marijuana, acupunture*).

3. **Relaciones personales:** Encontramos gran cantidad de pronombres, sobre todo en primera persona(I,my,me, myself) tambien palabras que dan la idea de relaciones (*we,ours,yours,she, friends, lover,those,together*)

4. **Enojo,frustración:** Palabras que transmiten momentos de intensidad emocional (*hell,wtf,weird, damn, freakin, cry*) y momentos de tristeza (*chill,lies ,hurts,screwes,sucks*)

5. **Divulgación, expresar ideas:** Hacen referencia a estudios (*reporting, test, essay, stats, pages, recomendation, research*), artículos, ensayos, libros revistas. Para indicar que son pláticas donde las personas hablan sobre conocimiento de cierto tema basado en su investigación

Los tópicos obtenidos son semejantes a los que obtienen (De Choudhury y col.)[12] en su trabajo sobre predicción de depresión en redes sociales, se coincide en que las personas con depresión ven en las redes sociales como una espacio para compartir sus sentimientos, sus relaciones interpersonales, para recibir apoyo emocional, discutir aspectos relacionados con su terapia y tratamiento y para divulgar información sobre la enfermedad.

Tabla 5.6: Palabras agrupadas por tópicos. De los tópicos encontrados podemos inferir que las personas con depresión suelen hablar de sus sentimientos en redes sociales, compartir sus experiencias con la enfermedad y discutir sobre el tratamiento que llevan

Temática	Palabras en el clúster
Animo, cariño y agradecimiento	thanks, congrats, awesome, yay, amazing, aww, wow, gorgeous, adorable, cheers, flawless, congratulations, cute, welcome, okay, please, miss, omg, nice, beautiful, sorry, love, hugs, hello, perfect, cutie, lucky, bye, baby, happy, amen, merry, bless, guys, luv, bff, cutest, lovely, enjoy, hug, follow, friend, appreciate, smile, sweet, mummy, proud, preach, goddamnit
Salud, enfermedades, tratamientos	depression, obesity, diabetes, treatment, cured, experiencing, drugs, headache, therapy, marijuana, treatments, bulimia, heroin, health, mental, therapist, physical, diagnosed, syndrome, panic, lsd, anxiety, disorder, diseases, addiction, pills, diagnosis, xanax, meds, schizophrenia, preventable, symptoms, prescription, consumption, cures, insomnia, autism,
Relaciones personales	my, well, done, again, friends, i,also, night, really, you, hopefully, new, better, still, best, sure, the, anyone, go, one, play, used, won, loved, people, take, wait, that, up, more, win, we, playing, ever, see, just, was, waiting, much, easy, only, someone, she, all, those, god, he, be, with, had, know, your, tonight, been, good, together, any, which, yours, already,
Enojo, frustración	nope, seriously, cool, lol, literally, wtf, sleeping, definitely, dead, sigh, feel, chill, absolutely, totally, fucking, lies, died, hurts, chillin, though, weird, fuck, damn, joke, hilarious, badly,problems, feels, funny, shit, dumbass, screwed, jealous, bored, sucks, stupid, knew, pretty, freakin, cry, shower, changed, drunk, happened, problem, sad, awkward, hell, perfectly, fine, gone
Divulgación, expresar ideas	interesting, reporting, recent, critique, review, test, confirmed, chemistry, stats, source, insightful, wikipedia, questions, signs, brilliant, interview, regarding, context, book, read, reference, clues, study, confirm, suggestions, comparison, post, reading, article, essay, studies, alternative, research, knowledge, proof, newspaper, recommendations, paper, recommend, books, chapter, updated, website

Capítulo 6

Conclusiones y trabajo futuro

En este trabajo de tesis, exploramos cómo se compartan los modelos de aprendizaje profundo en la tarea de detección de depresión en redes sociales, específicamente provenientes de Reddit. Comprobamos tal como (Park y col.)[47] mencionan en su trabajo, la depresión afecta la forma en la que las personas se comunican, y por tanto siendo las redes sociales un espacio importante en la interacciones sociales, es natural que sean una fuente de información rica para este tipo de estudios.

Uno de los objetivos planteados en este trabajo, es poder comparar el desempeño de diferentes métodos de aprendizaje automático entre ellos redes neuronales profundas en esta tarea. Nuestro valor de referencia es una clasificador basado en maquina de soporte vectorial (SVM) con una representación del texto TFIDF. Nuestra métrica de interés el *F1Score*. En este punto vemos que el desempeño de las redes resulta ser semejante al obtenido como *baseline* e incluso superado para las arquitecturas de redes recurrentes. Para la versión de datos a nivel usuario, el *F1 Score* base fue 0.509 mientras que para la CNN y la BiLSTM tuvimos 0.479 y 0.535 respectivamente. Para la versión de datos a nivel post el *F1 Score* base fue 0.212 mientras que para la CNN y la BiLSTM tuvimos 0.210 y 0.291 respectivamente. Vemos que las arquitecturas recurrentes logran superar el resultado base. Aunque obtenemos resultados que compiten con el valor de referencia, debemos resaltar factores que afectan sin duda en el desempeño de nuestros clasificadores. Entre los factores consideramos al desbalanceo de los ejemplos por clase, en ambas versiones de los datos la relación del grupo control

contra el grupo de interés era alrededor de 1 a 8, sabemos que esto afecta al desempeño de clasificador ya que tenderá a clasificar todos como la clase mayoritaria. También detectamos algunas dificultades inherentes al conjunto de datos utilizado, entre estos el que la cantidad de ejemplos por usuario sea muy dispersa, es decir usuarios con entradas escasas y otros con abundantes de ellas, relacionado con este punto la cantidad de palabras por entrada representó un reto adicional por el costo computacional asociado a procesar la totalidad de información. Recordemos que Reddit es un sitio donde las personas publican en foros que tienen diferentes temáticas, así que al extraer las publicaciones de cada usuario es natural que la temática de cada publicación este centrada a la del foro en el que fue publicada, es por eso que detectamos ejemplos que se centraban en temáticas ajenas al problema como foros de espectáculos, tecnología y videojuegos que aportaban poco al proceso de aprendizaje.

Uno de los retos propuestos en este trabajo, era lograr interpretabilidad del proceso de aprendizaje, en este punto logramos identificar palabras que el sistema considera importante para clasificar a una persona con depresión. En este sentido, los resultados obtenidos concuerdan con lo conocido sobre la enfermedad, es decir que las personas con depresión tiene un discurso en redes sociales donde agradecen la atención de otros hacia ellos, buscan dar ánimo a otros que se encuentran en condiciones parecidas a ellos, pueden tener arrebatos de ira o de tristeza, suele compartir información sobre el tema con los demás, suelen hablar de si mismos y sus relaciones interpersonales (Park y col., Moreno y col.)[47, 43].

Tomamos la filosofía de hacer la cantidad mínima de modificaciones al conjunto de datos par que sean las redes quienes 'decidan' lo que es relevante en la tarea. En este punto y considerando los resultados obtenidos, a futuro podría obtenerse mejoras usando modelos adiciones para la selección de características o fusionando modelos de aprendizaje automático con modelos de aprendizaje profundo en busca de sinergia, igual considerar el uso de metadatos para enriquecer la información contextual de los datos.

También experimentar con modelos de mayor potencia que en la actualidad se encuentran en el estado del arte, tales como los transformers que se basan en el

mecanismo de atención usado en este trabajo pero sin hacer uso de redes recurrentes.

Bibliografía

[1] Liliya Akhtyamova, Mikhail Alexandrov y John Cardiff. "Adverse drug extraction in twitter data using convolutional neural network". En: *2017 28th International Workshop on Database and Expert Systems Applications (DEXA)*. IEEE. 2017, págs. 88-92.

[2] Ghelmar Astoveza y col. "Suicidal behavior detection on twitter using neural network". En: *TENCON 2018-2018 IEEE Region 10 Conference*. IEEE. 2018, págs. 0657-0662.

[3] Baharum Baharudin, Lam Hong Lee y Khairullah Khan. "A Review of Machine Learning Algorithms for Text-Documents Classification". En: *Journal of Advances in Information Technology* (2010). ISSN: 1798-2340. DOI: 10.4304/jait.1.1.4-20.

[4] Dzmitry Bahdanau, Kyung Hyun Cho y Yoshua Bengio. "Neural machine translation by jointly learning to align and translate". En: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* (2015), págs. 1-15. arXiv: 1409.0473.

[5] Hend Basly y col. "CNN-SVM Learning Approach Based Human Activity Recognition". En: *International Conference on Image and Signal Processing*. Springer. 2020, págs. 271-281.

[6] Yoshua Bengio, Réjean Ducharme y Pascal Vincent. "A neural probabilistic language model". En: *Advances in Neural Information Processing Systems* 3 (2001), págs. 1137-1155. ISSN: 10495258.

[7] Danushka Bollegala, Yutaka Matsuo y Mitsuru Ishizuka. "Measuring semantic similarity between words using web search engines". En: *16th International World Wide Web Conference, WWW2007*. 2007. ISBN: 1595936548. DOI: 10.1145/1242572.1242675.

[8] Junyi Chen, Shankai Yan y Ka-Chun Wong. "Verbal aggression detection on Twitter comments: Convolutional neural network for short-text sentiment analysis". En: *Neural Computing and Applications* (2018), págs. 1-10.

[9] Corinna Cortes y Vladimir Vapnik. "Support-Vector Networks". En: *Machine Learning* (1995). ISSN: 15730565. DOI: 10.1023/A:1022627411411.

[10] Eleonora D'Andrea y col. "Real-time detection of traffic from twitter stream analysis". En: *IEEE transactions on intelligent transportation systems* 16.4 (2015), págs. 2269-2283.

[11] Munmun De Choudhury, Scott Counts y Eric Horvitz. "Predicting postpartum changes in emotion and behavior via social media". En: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2013, págs. 3267-3276.

[12] Munmun De Choudhury y col. "Predicting depression via social media". En: *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 7. 1. 2013.

[13] Laura Patricia Del Bosque y Sara Elena Garza. "Prediction of aggressive comments in social media: an exploratory study". En: *IEEE Latin America Transactions* 14.7 (2016), págs. 3474-3480.

[14] Mahidhar Dwarampudi y NV Reddy. "Effects of padding on LSTMs and CNNs". En: *arXiv preprint arXiv:1903.07288* (2019).

[15] Paul S Earle, Daniel C Bowden y Michelle Guy. "Twitter earthquake detection: earthquake monitoring in a social world". En: *Annals of Geophysics* 54.6 (2012).

[16] Giorgio Fumera, Ignazio Pillai y Fabio Roli. "Spam filtering based on the analysis of text information embedded into images". En: *Journal of Machine Learning Research* (2006). ISSN: 15324435.

[17] Kavita Ganesan. *Word2Vec*. 2015. URL: https://kavita-ganesan.com/comparison-between-cbow-skipgram-subword/#.X5tWi4hKiCg.

[18] Ian Goodfellow y col. *Deep learning*. Vol. 1. MIT press Cambridge, 2016.

[19] Google. *Google Colaboratory*. URL: https://colab.research.google.com/notebooks/welcome.ipynb?hl=es.

[20] Andreas Hess y Nicholas Kushmerick. "Automatically attaching semantic metadata to Web Services". En: *Proceedings of IIWeb*. 2003.

[21] Christian Karmen, Robert C Hsiung y Thomas Wetter. "Screening Internet forum participants for depression symptoms by assembling and enhancing multiple NLP methods". En: *Computer methods and programs in biomedicine* 120.1 (2015), págs. 27-36.

[22] Kantinee Katchapakirin y col. "Facebook social media for depression detection in the Thai community". En: *2018 15th International Joint Conference on Computer Science and Software Engineering (JCSSE)*. IEEE. 2018, págs. 1-6.

[23] Raghavendra Katikalapudi y col. "Associating internet usage with depressive behavior among college students". En: *IEEE Technology and Society Magazine* 31.4 (2012), págs. 73-80.

[24] Renu Khandelwal. *Word Embeddings for NLP*. 2019. URL: https://towardsdatascience.com/word-embeddings-for-nlp-5b72991e01d4.

[25] Yoon Kim. "Convolutional neural networks for sentence classification". En: *arXiv preprint arXiv:1408.5882* (2014).

[26] Diederik P. Kingma y Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].

[27] Teuvo Kohonen. "An introduction to neural computing". En: *Neural networks* 1.1 (1988), págs. 3-16.

[28] Ji Young Lee y Franck Dernoncourt. "Sequential short-text classification with recurrent and convolutional neural networks". En: *arXiv preprint arXiv:1603.03827* (2016).

[29] Kristina Lerman, Anon Plangprasopchok y Craig A. Knoblock. "Automatically labeling the inputs and outputs of web services". En: *Proceedings of the National Conference on Artificial Intelligence*. 2006. ISBN: 1577352815.

[30] Joseph Lilleberg, Yun Zhu y Yanqing Zhang. "Support vector machines and word2vec for text classification with semantic features". En: *2015 IEEE 14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)*. IEEE. 2015, págs. 136-140.

[31] Huijie Lin y col. "Detecting stress based on social interactions in social networks". En: *IEEE Transactions on Knowledge and Data Engineering* 29.9 (2017), págs. 1820-1833.

[32] Ying Liu y col. "Handling of imbalanced data in text classification: Category-based term weights". En: *Natural Language Processing and Text Mining*. 2007. ISBN: 184628175X. DOI: 10.1007/978-1-84628-754-1_10.

[33] David E Losada y Fabio Crestani. "A test collection for research on depression and language use". En: *International Conference of the Cross-Language Evaluation Forum for European Languages*. Springer. 2016, págs. 28-39.

[34] David E. Losada, Fabio Crestani y Javier Parapar. "CLEF 2017 eRisk overview: Early Risk prediction on the internet: Experimental foundations". En: *CEUR Workshop Proceedings* 1866 (2017). ISSN: 16130073.

[35] David E Losada, Fabio Crestani y Javier Parapar. "Overview of eRisk 2018: Early Risk Prediction on the Internet (extended lab overview)". En: *Proceedings of the 9th International Conference of the CLEF Association, CLEF*. 2018.

[36] David E. Losada, Fabio Crestani y Javier Parapar. "Overview of eRisk 2019 Early Risk Prediction on the Internet". En: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11696 LNCS (2019), págs. 340-357. ISSN: 16113349. DOI: 10.1007/978-3-030-28577-7_27.

[37] Jing Ma, Wei Gao y Kam-Fai Wong. "Rumor detection on twitter with tree-structured recursive neural networks". En: Association for Computational Linguistics. 2018.

[38] Long Ma y Yanqing Zhang. "Using Word2Vec to process big text data". En: *2015 IEEE International Conference on Big Data (Big Data)*. IEEE. 2015, págs. 2895-2897.

[39] Christopher D Manning, Hinrich Schütze y Prabhakar Raghavan. *Introduction to information retrieval*. Cambridge university press, 2008.

[40] Tomas Mikolov y col. "word2vec". En: *URL https://code. google. com/p/word2vec* 22 (2013).

[41] Saif Mohammad y Peter Turney. "NRC Word-Emotion Association Lexicon (aka EmoLex)". En: *National Research Council Canada (NRC).[Online] Available at: http://saifmohammad. com/WebPages/NRC-Emotion-Lexicon. htm (Accessed: 04 November 2017)* (2011).

[42] Miguel Ángel Molina Cabello y col. "Segmentación y detección de objetos en imágenes y vídeo mediante inteligencia computacional". En: (2018).

[43] Megan A Moreno y col. "Feeling bad on Facebook: Depression disclosures by college students on a social networking site". En: *Depression and anxiety* 28.6 (2011), págs. 447-455.

[44] Christopher Olah. *Understanding LSTM Networks*. 2020. URL: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[45] Aytuğ Onan. "Topic-enriched word embeddings for sarcasm identification". En: *Computer Science On-line Conference*. Springer. 2019, págs. 293-304.

[46] Organización Mundial de la salud. *Mental disorders*. https://www.who.int/es/news-room/fact-sheets/detail/mental-disorders, Consultado el 2020-02-12. 2020.

[47] Minsu Park, Chiyoung Cha y Meeyoung Cha. "Depressive moods of users portrayed in Twitter". En: *Proceedings of the ACM SIGKDD Workshop on healthcare informatics (HI-KDD)*. Vol. 2012. 2012, págs. 1-8.

[48] Vik Paruchuri. *Natural Language Processing*. 2013. URL: https://github.com/VikParuchuri/vikparuchuri-affirm/blob/master/natural-language-processing-tutorial.md.

[49] Michael J Paul y Mark Dredze. "You are what you tweet: Analyzing twitter for public health". En: *Fifth International AAAI Conference on Weblogs and Social Media*. 2011.

[50] Eugene S. Paykel. "Basic concepts of depression". En: *Dialogues in Clinical Neuroscience* 10.3 (2008), págs. 279-289. ISSN: 12948322.

[51] Jeffrey Pennington, Richard Socher y Christopher D. Manning. "GloVe: Global vectors for word representation". En: *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference* (2014), págs. 1532-1543. DOI: 10.3115/v1/d14-1162.

[52] Georgios K Pitsilis, Heri Ramampiaro y Helge Langseth. "Effective hate-speech detection in Twitter data using recurrent neural networks". En: *Applied Intelligence* 48.12 (2018), págs. 4730-4742.

[53] Pablo Daniel Pusiol. "Redes convolucionales en comprensión de escenas". B.S. thesis. 2014.

[54] Seyed Mahdi Rezaeinia y col. "Sentiment analysis based on improved pre-trained word embeddings". En: *Expert Systems with Applications* 117 (2019), págs. 139-147.

[55] Adalberto Rincón Núñez y col. "Descripción de escenas por medio de aprendizaje profundo". B.S. thesis. Universidad Autónoma de Occidente, 2018.

[56] Frank Rosenblatt. *The perceptron, a perceiving and recognizing automaton Project Para*. Cornell Aeronautical Laboratory, 1957.

[57] Jonathan Rottenberg. "Mood and emotion in major depression". En: *Current Directions in Psychological Science* 14.3 (2005), págs. 167-170.

[58] David E Rumelhart, Geoffrey E Hinton y Ronald J Williams. "Learning representations by back-propagating errors". En: *nature* 323.6088 (1986), págs. 533-536.

[59] Adam Sadilek, Henry Kautz y Vincent Silenzio. "Modeling spread of disease from social interactions". En: *Sixth International AAAI Conference on Weblogs and Social Media*. 2012.

[60] G. Salton, A. Wong y C. S. Yang. "A Vector Space Model for Automatic Indexing". En: *Communications of the ACM* (1975). ISSN: 15577317. DOI: 10.1145/361219.361220.

[61] Fabrizio Sebastiani. *Machine Learning in Automated Text Categorization*. 2002. DOI: 10.1145/505282.505283. arXiv: 0110053 [cs].

[62] Statista. *Daily social media usage worldwide*. https://www.statista.com/statistics/433871/daily-social-media-usage-worldwide/, Consultado el 2020-02-12. 2018.

[63] Ewan Klein Steven Bird y Edward Loper. *Analyzing Text with the Natural Language Toolkit*. URL: https://www.nltk.org/book/ch06.html.

[64] Michael M Tadesse y col. "Detection of depression-related posts in reddit social media forum". En: *IEEE Access* 7 (2019), págs. 44883-44893.

[65] Carmen Torres López y Leticia Arco García. "Representación textual en espacios vectoriales semánticos". En: *Revista Cubana de Ciencias Informáticas* 10.2 (2016), págs. 148-180.

[66] D. Yu Turdakov. "Word sense disambiguation methods". En: *Programming and Computer Software* (2010). ISSN: 03617688. DOI: 10.1134/S0361768810060010.

[67] Yu-Tseng Wang, Hen-Hsen Huang y Hsin-Hsi Chen. "A Neural Network Approach to Early Risk Detection of Depression and Anorexia on Social Media Text." En: *CLEF (Working Notes)*. 2018.

[68] Lilian Weng. *Attention? Attention!* 2018. URL: https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html.

[69] Liu Wensen y col. "Short text classification based on Wikipedia and Word2vec". En: *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*. IEEE. 2016, págs. 1195-1200.

[70] Jun Xie y col. "Self-Attention-Based BiLSTM Model for Short Text Fine-Grained Sentiment Classification". En: *IEEE Access* 7 (2019), págs. 180558-180570. ISSN: 21693536. DOI: 10.1109/ACCESS.2019.2957510.

[71] Shufeng Xiong y col. "Towards Twitter sentiment classification by multi-level sentiment-enriched word embeddings". En: *Neurocomputing* 275 (2018), págs. 2459-2466.

[72] Haotian Xu y col. "Text classification with topic-based word embedding and convolutional neural networks". En: *Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*. 2016, págs. 88-97.

[73] Dawei Yin y col. "Detection of harassment on web 2.0". En: *Proceedings of the Content Analysis in the WEB* 2 (2009), págs. 1-7.

[74] Peng Zhou y col. "Attention-based bidirectional long short-term memory networks for relation classification". En: *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: Short papers)*. 2016, págs. 207-212.