



INAOE

Aumento de datos para tareas relacionadas al perfilado de autor

por

Victor Jimenez Villar

Tesis sometida como requisito parcial para obtener el grado de
Maestro en Ciencias en el Área de Ciencias Computacionales en el
Instituto Nacional de Astrofísica, Óptica y Electrónica

Supervisada por:

Dr. Luis Villaseñor Pineda, INAOE

©INAOE 2020

El autor otorga al INAOE el permiso de reproducir y distribuir copias
en su totalidad o en partes de esta tesis



Aumento de datos para tareas relacionadas al perfilado de autor

Tesis de Maestría

POR:

Víctor Jiménez Villar

ASESOR:

Dr. Luis Villaseñor Pineda

Instituto Nacional de Astrofísica Óptica y Electrónica
Coordinación de Ciencias Computacionales

Agradecimientos

Esta investigación fue realizada gracias al apoyo otorgado por el Consejo Nacional de Ciencia y Tecnología (CONACYT), a través de la beca No. 718947.

Dedicatoria

A mi familia y amigos, por motivarme cada día superar mis límites.

Resumen

Para resolver las tareas de perfilado de autor, la mayoría de los trabajos existentes se han enfocado en utilizar algoritmos de aprendizaje computacional en combinación con diferentes técnicas para extraer características. La obtención de dichas características requiere un análisis riguroso y en muchos casos es necesaria la intervención de expertos en el tema. Sin embargo existen técnicas de aprendizaje computacional más complejas como las redes neuronales en donde la extracción de características se realiza de forma automática mediante una serie de abstracciones.

La principal motivación para el uso de redes neuronales en perfilado de autor, es debido al increíble éxito del aprendizaje profundo en tareas complejas del procesamiento del lenguaje natural. De acuerdo al estado del arte en la última conferencia del PAN@CLEF los equipos con mejores resultados utilizaron técnicas tradicionales de aprendizaje. Así también en las tareas del ERISK el mejor sistema se construyó extrayendo características en combinación con un ensamble de bolsas de palabras y diferentes clasificadores. Lo que se ha podido observar en los diferentes reportes de estas conferencias es, que los modelos de aprendizaje basados en redes neuronales no han tenido el éxito esperado.

Uno de los principales problemas dentro es la cantidad de datos etiquetados con que se cuenta y se hace más notable cuando se utilizan modelos de aprendizaje profundo. En el caso de perfilado de autor la obtención de estos datos etiquetados manualmente consumen mucho tiempo, son costosos, además se podría comprometer a problemas legales debido al uso de datos personales.

Dada esta problemática este trabajo presenta un estudio sobre el efecto de agregar documentos nuevos, generados artificialmente, mediante aumento de datos a nivel estructural, al conjunto de entrenamiento original y el efecto que tiene en los algoritmos de redes neuronales aplicados en tareas relacionadas al perfilado de autor. Para ello, en esta tesis se propone un esquema general para el aumento de datos con diferentes

estrategias de selección y reemplazo de palabras, principalmente enfocándose a las relaciones de similitud de las palabras.

Gracias a los experimentos realizados fue posible concluir que el aumento de datos propuesto puede mejorar la predicción en tareas relacionadas al perfilado de autor, en comparación con no realizar aumento de datos y algoritmos existen para el aumento de datos en la clasificación textos.

Abstract

The main motivation for the use of neural networks in author profiling is due to the incredible success of deep learning in complex tasks of natural language processing. According to the state of the art in the last PAN @ CLEF conference, the teams with the best results used traditional learning techniques. So also in the ERISK tasks the best system was built by extracting characteristics in combination with an assemble of “bags of words.” and different classifiers. What has been observed in the different reports of these conferences is that the learning models based on neural networks had not the expected success.

One of the main problems within is the amount of tagged data that is available and it becomes more noticeable when using deep learning models. In the case of author profiling, obtaining this manually tagged data is time-consuming, expensive, and could also lead to legal problems due to the use of personal data.

Given this problem, this work presents a study on the effect of adding new, artificially generated documents, by increasing data at a structural level, to the original training set and the effect it has on the neural network algorithms applied in tasks related to the profiling of Author. To do this, this thesis proposes a general scheme for increasing data with different strategies for selecting and replacing words, mainly focusing on the similarity relationships of words.

Thanks to the experiments carried out, it was possible to conclude that the proposed increase in data can improve prediction in tasks related to author profiling, compared to not performing data augmentation, and current algorithms for increasing data in text classification.

Tabla de Contenido

Agradecimientos	I
Dedicatoria	III
Resumen	V
Abstract	VII
Lista de Figuras	XIII
Lista de Tablas	XV
1. Introducción	1
1.1. Planteamiento del problema	2
1.2. Objetivo general	4
1.3. Objetivos específicos	4
1.4. Organización de la tesis	4
2. Marco Teórico	7
2.1. Clasificación de textos	7
2.2. Extracción de características	8
2.2.1. Pre-procesamiento	8
2.2.2. N-Gramas	10
2.2.3. Bolsa de palabras BoW	10
2.2.4. Representaciones distribucionales	11
2.3. Selección de características	12
2.4. Algoritmos de clasificación	13

2.4.1. Máquinas de Soporte Vectorial (SVM)	13
2.4.2. Redes neuronales profundas	15
2.5. Medidas de Evaluación	19
3. Trabajo Relacionado	21
3.1. Perfilado de autor	21
3.1.1. Detección de trastornos mentales en redes sociales	23
3.2. Aumento de datos	26
3.2.1. Aumento de datos supervisado	27
3.2.2. Aumento de datos semi-supervisado	27
3.2.3. Aumento de datos en clasificación de textos	28
3.2.4. Discusión del trabajo relacionado	30
4. Método propuesto	33
4.1. Selección de palabras a reemplazar	34
4.1.1. Etiquetado de partes de la oración	34
4.1.2. Exclusión de palabras importantes	35
4.2. Reemplazo de palabras seleccionadas	36
4.2.1. Similitud relacional	37
5. Configuración experimental y resultados	41
5.1. Configuración experimental	41
5.1.1. Conjunto de datos	42
5.1.2. Pre-procesamiento	44
5.1.3. Configuración de los métodos propuestos	46
5.1.4. Configuración de los modelos de aprendizaje	49
5.2. Resultados	54
5.3. Análisis y discusión de los resultados	56
5.3.1. Comparación con el estado del arte en detección de depresión y anorexia	61
5.3.2. Análisis del aumento de datos	62
6. Conclusiones y Trabajo Futuro	67
6.1. Conclusiones	67
6.2. Trabajo futuro	68

Apéndices	69
Bibliografía	69

Lista de Figuras

2.1. Una celda de LSTM desdoblada sobre el tiempo. Obtenida de Wikimedia Commons y modificada bajo la licencia Creative Commons 4.0	18
4.1. Función de masa para la distribución geométrica con diferentes valores de probabilidad.	35
5.1. Diagrama general de la configuración experimental	42
5.2. Distribución del numero de palabras en los historiales de usuarios estudiados	44
5.3. Arquitectura del modelo Bi-LSTM	52
5.4. Arquitectura del modelo CNN con múltiples tamaños de convolución	53
5.5. Relación entre el aumento del conjunto de datos <i>Depresión 2018</i> y la ganancia en F1.	58
5.6. Relación entre el aumento del conjunto de datos <i>Depresión 2019</i> y la ganancia en F1.	59
5.7. Relación entre el aumento del conjunto de datos y la ganancia en F1 para el conjunto <i>Anorexia</i>	60
5.8. Comparación con el estado del arte en detección de depresión y anorexia	61
5.9. Relación entre el aumento de datos y el vocabulario nuevo agregado. .	63
5.10. Palabras con mayor puntuación χ^2	65

Lista de Tablas

2.1. Matriz de confusión.	19
3.1. Comparación del método propuesto con el estado del arte en aumento de datos para clasificación de textos.	32
4.1. Ejemplo de etiquetado de partes de la oración.	35
5.1. Número de usuarios en los conjuntos de datos y número de secuencias con 64 palabras después del pre-procesamiento, sin aplicar el filtro. Los números resaltados en negritas representan el numero de historiales comparado con el numero de secuencias.	43
5.2. Número de usuarios en los conjuntos de datos y número de secuencias con 64 palabras después del pre-procesamiento, realizando el filtro. Los números resaltados en negritas representan el numero de historiales comparado con el numero de secuencias	44
5.3. Etiquetas utilizadas en el proceso de aumento para los métodos de similitud relacional.	48
5.4. Ejemplos del aumento de datos, las palabras resaltadas en negritas son las que resultaron afectadas después de la transformación.	49
5.5. Ejemplos del aumento de datos para el método basado en relaciones contrarias, las palabras resaltadas en negritas son las que resultaron afectadas después de la transformación.	49
5.6. Parámetros utilizados para el entrenamiento de los modelos basados en redes neuronales.	54
5.7. Resultados en términos de la métrica F1, la variable n indica la magnitud del aumento en el conjunto original.	56

Capítulo 1

Introducción

Imagina que se te ha dado un texto de un autor anónimo, y deseas saber tanto como sea posible del autor (género, ocupación, personalidad etc.), sólo analizando el texto dado. Es sorprendente pero el texto refleja parte de la personalidad del autor. Así que, observando el texto, al determinar su estilo y contenido, es posible inferir información sobre el autor. A esta tarea se le conoce como *perfilado de autor* y está fundada en estudios dentro de la comunidad sociolingüística que han demostrado que las palabras utilizadas en la vida diaria pueden revelar importantes aspectos sociales y psicológicos. Gracias a los avances en computación, el análisis de textos permite a los investigadores obtener características de lo que las personas dicen y también de las particularidades en sus estilos lingüísticos (Pennebaker, Mehl, y Niederhoffer, 2002).

El interés en esta tarea ha crecido gracias al constante flujo de información compartida a través de redes sociales (por ejemplo Twitter ¹, Facebook², Reddit ³) y sus aplicaciones varían desde mercadotecnia hasta seguridad nacional. Existen numerosas razones del por que nos interesa conocer datos relevantes de los usuarios de redes sociales. Por ejemplo, a las empresas les interesaría conocer a qué tipo de usuarios les gusta o no su producto o servicio, con la intención de dirigir una mejor campaña de publicidad (Ikeda et al., 2013). Además en un contexto de seguridad informática, a la policía cibernética le gustaría conocer el perfil de las personas que envían mensajes amenazantes o de acoso sexual (Bogdanova, Rosso, y Solorio, 2012).

Claro está que la tarea no es simple y debido al lenguaje informal de redes sociales y poco estandarizado hace que esta tarea sea aún más desafiante, por ejemplo: errores gramaticales, abreviaturas, anglicismos, emoticonos o incluso texto generado por

¹www.twitter.com

²www.facebook.com

³www.reddit.com

cuentas automáticas. Una de las conferencias más destacadas en perfilado de autor ha sido el PAN@CLEF⁴ (una serie de eventos científicos y tareas compartidas en el análisis forense digital y estilométrico); desde el año 2013 al actual se han estudiado diversos enfoques del perfilado de autor desde una perspectiva multi-idioma (inglés y español principalmente) entre las cuales destacan: identificación de edad y género (Rangel et al., 2013), identificación de personalidad, variación de lenguaje y dimensión de género (Stammatatos et al., 2015). Un tema de particular importancia es el perfilado de características de comportamiento (Kumar et al., 2018) y condiciones médicas (De Choudhury et al., 2013); estas tareas se han desarrollado en subcampos y con sus propias conferencias. Por ejemplo, la conferencia eRISK (Losada, Crestani, y Parapar, 2018), está orientada al perfilado del autor en búsqueda de evidencia de trastornos como la depresión o la anorexia.

1.1. Planteamiento del problema

Para resolver las tareas de perfilado de autor, la mayoría de los trabajos existentes se han enfocado en utilizar algoritmos de aprendizaje computacional en combinación con diferentes técnicas para extraer características: conteo de palabras (Laserna, Seih, y Pennebaker, 2014), identificación de frases personales (Ortega-Mendoza et al., 2018a), análisis de emociones (Aragón et al., 2019) entre otras técnicas. La obtención de dichas características requiere un análisis riguroso y en muchos casos es necesaria la intervención de expertos en el tema. Sin embargo existen técnicas de aprendizaje computacional más complejas como las redes neuronales en donde la extracción de características se realiza de forma automática mediante una serie de abstracciones.

La principal motivación para el uso de redes neuronales en perfilado de autor, es debido al increíble éxito del aprendizaje profundo en tareas complejas para el entendimiento del lenguaje : parafraseo, traducción automática, analogía, implicación textual, similitud semántica, etc. En el conjunto de datos GLUE (Wang et al., 2018) los modelos de aprendizaje profundo han superado la puntuación humana, Christopher D. Manning menciona que desde el año 2015 se produjo un tsunami del aprendizaje profundo en el área de procesamiento de lenguaje natural, debido a la gran cantidad de artículos en conferencias de NLP utilizando aprendizaje profundo (Manning, 2015).

⁴www.pan.webis.de

De acuerdo al estado del arte en la última conferencia del PAN@CLEF los equipos con mejores resultados utilizaron técnicas tradicionales de aprendizaje como lo son máquinas de soporte vectorial SVM en combinación con n-gramas de caracteres. Así también en las tareas del eRISK el mejor sistema se construyó extrayendo características en combinación con un ensamble de bolsas de palabras (BOW por sus siglas en inglés) y diferentes clasificadores. Lo que se ha podido observar en los diferentes reportes de estas conferencias es, que los modelos de aprendizaje basados en redes neuronales no han tenido el éxito esperado.

Uno de los principales problemas dentro del campo de aprendizaje automático es que el éxito de éste depende de la cantidad de datos etiquetados con que se cuente y se hace más notable cuando se utilizan modelos de aprendizaje profundo, el etiquetado manual de datos consume mucho tiempo y es costoso, además se podría incurrir en problemas legales debido al uso de datos personales, como es el caso en las tareas de perfilado de autor. Los estudios actuales tratan con un número pequeño de autores conocidos, en donde el etiquetado manual puede ser aplicado, pero considerando las dimensiones de los datos en redes sociales se convierte en una tarea costosa y difícil.

Uno de los problemas conocidos en la clasificación de textos es el sobre ajuste de los modelos de aprendizaje, el cual se genera en la etapa de entrenamiento debido a que el modelo de memoriza los pocos documentos de entrenamiento. Por lo tanto ante esta situación es deseable tener una amplia diversidad en los textos, es decir tener frases que signifiquen lo mismo pero escritas de forma diferente; para esto se han propuesto diferentes técnicas como el aumento de datos o agregar ruido aleatorio a los ejemplos originales.

Observando las limitantes anteriores este trabajo presenta un estudio incrementando el conjunto de datos de entrenamiento, observando el efecto al agregar documentos nuevos. Para ello, se crean nuevos documentos respetando su estructura, y se agregan al conjunto de entrenamiento original. El estudio analiza el efecto que tiene este incremento en los algoritmos de redes neuronales tradicionales en tareas relacionadas al perfilado de autor.

Algunas de las principales preguntas a contestar en esta investigación son:

- 1.- ¿Cómo conservar el estilo y contenido para el aumento de datos en perfilado de autor?
- 2.- ¿En qué tipo de arquitecturas basadas en redes neuronales tiene mayor impacto

el aumento de datos?

- 3.- ¿Se puede mejorar el perfilado de trastornos mentales mediante el aumento de datos?

1.2. Objetivo general

Proponer un método de aumento de datos, considerando estilo y contenido del texto, para mejorar la predicción de los modelos de aprendizaje profundo en las tareas de perfilado de autor.

1.3. Objetivos específicos

- 1.- Diseñar diferentes estrategias de aumento de datos bajo condiciones supervisadas, que permita conservar el estilo y contenido del documento original y a la vez aumentar el vocabulario.
- 2.- Demostrar el efecto del aumento de datos tanto en modelos de redes neuronales como en modelos lineales.
- 3.- Evaluar y analizar los métodos propuestos para abordar el perfilado de condiciones médicas en redes sociales.

1.4. Organización de la tesis

Esta tesis está organizada de la siguiente forma:

- **Capítulo 2: Marco teórico.** Presenta una rápida introducción a la clasificación de textos con aprendizaje automático, además de mencionar las principales métricas de evaluación utilizadas en este trabajo. Los conceptos descritos son fundamentales para comprender la solución propuesta.
- **Capítulo 3: Trabajo relacionado.** Describe el estado del arte en perfilado de autor y aumento de datos para clasificación de textos, su principal objetivo es conocer como se ha abordado el problema y analizar las ventajas y desventajas de los métodos existentes.

- **Capítulo 4: Métodos propuestos:** En este capítulo se describen a detalle los métodos propuestos, su justificación y el enfoque supervisado aplicado.
- **Capítulo 5: Configuración experimental y resultados:** En este capítulo se describen los conjuntos de datos estudiados y la configuración para los distintos clasificadores empleados, así como los métodos propuestos y los métodos de referencia o de comparación. Se realiza una comparación de los resultados obtenidos con el estado del arte para la detección de depresión y anorexia.
- **Capítulo 6: Análisis y discusión:** Se discute el rendimiento del método de acuerdo a los diferentes parámetros, configuraciones y las dificultades encontradas.
- **Capítulo 7: Conclusiones y trabajo futuro:** Por último se exponen las principales contribuciones de este trabajo y formas en que se puede mejorar.

Capítulo 2

Marco Teórico

En este capítulo se describen conceptos relacionados con la tarea de perfilado de autor mediante algoritmos de aprendizaje automático. Se describen las principales representaciones de un texto dado, las características generales de los clasificadores empleados, así como las medidas de evaluación empleadas para medir los resultados de los diferentes modelos. Además se presenta una introducción de las tareas de procesamiento de lenguaje natural utilizadas en el método propuesto: etiquetado de partes de la oración, paráfrasis y resolución de analogías.

2.1. Clasificación de textos

En años recientes, ha habido un crecimiento exponencial en el número de textos disponibles en internet, a tal grado que es imposible procesarlos manualmente, de ahí su indispensable procesamiento por medios automáticos. Los problemas de clasificación automática de textos han sido ampliamente estudiados en las últimas décadas, especialmente con los recientes avances en procesamiento de lenguaje natural, muchos investigadores están interesados en desarrollar aplicaciones que mejoren los métodos de clasificación de textos. La tarea motivo de este trabajo se circunscribe a las tareas de perfilado de autor, en donde se desea conocer la categoría (clase o tipo de autores) a la que pertenece un documento dado (historial del usuario).

Definición

La clasificación de textos puede ser definida como la tarea de categorizar un grupo de documentos en una o más clases predefinidas de acuerdo a sus temas (Kadhim, 2019). Retomando la definición de (Kadhim, 2019) se parte con un grupo específico

de documentos $D = \{d_1, \dots, d_n\}$ con clases predefinidas $C = \{c_1, \dots, c_m\}$ y un nuevo documento q el cual es generalmente indicado como una consulta, con el objetivo de predecir la clase del documento consultado la cual puede ser una o más clases en C .

De acuerdo a (Kowsari et al., 2019), la clasificación de textos puede describirse en cuatro pasos: extracción de características, reducción de dimensionalidad o selección de características, selección del clasificador y evaluación. A continuación se resumen algunos de los puntos más importantes para este trabajo, tomados del análisis de (Kowsari et al., 2019).

2.2. Extracción de características

El pre-procesamiento y la extracción de características son pasos muy importantes en la clasificación de textos, en las siguientes secciones se presentan algunas de las técnicas más empleadas y se mencionan dos métodos de representación de características: la bolsa de palabras y las representaciones distribucionales.

2.2.1. Pre-procesamiento

Dependiendo de la tarea de clasificación, algunos elementos pueden desecharse para enfocar nuestra atención en los elementos más informativos. Algunos de estos elementos son: palabras de paro, errores gramaticales, signos de puntuación, etc. Además de esto, el texto extraído de redes sociales contiene enlaces de internet, menciones de usuario, etiquetas (conocidos como hashtags), emoticonos y un vocabulario muy informal (p.e. abreviaturas no convencionales). A continuación se explica brevemente algunas técnicas empleadas para el limpiado y pre-procesamiento de textos.

1.- Tokenización: Es un método de pre-procesamiento en el cual se divide una cadena de caracteres en palabras, frases, símbolos y otros elementos dentro del texto llamados *tokens* (Kowsari et al., 2019). Se pueden utilizar diferentes algoritmos para poder realizarlo lo más simple es separar el texto mediante un espacio o caracter común, por ejemplo:

Texto original: *“Los días de verano son calurosos”*.

Los tokens del texto anterior son los siguientes: {“Los”, “días”, “de”, “verano”, “son”, “calurosos”}

- 2.- **Palabras de paro:** Son palabras con mayor frecuencia en los documentos, y por lo tanto poco útiles para la discriminación entre documentos de diferentes clases. Ejemplos de ellas son: {“*a*” “*the*”, “*they*”, “*he*” , “*she*”, ...} (para el idioma Inglés). En algunas tareas de clasificación de textos las palabras de paro no son de importancia y lo más común es removerlas de los documentos o textos. Nothman (Nothman, Qin, y Yurchak, 2018) presenta un análisis de las palabras de paro.
- 3.- **Capitalización:** Los textos contienen diversas formas de capitalización de palabras para formar una oración. Dado que los documentos consisten en muchas oraciones, una capitalización diversa puede ser muy problemática en la clasificación de textos largos. La técnica más común para tratar la capitalización inconsistente es reducir cada palabra a minúsculas (Kowsari et al., 2019).
- 4.- **Reducción de ruido:** La mayoría de los textos contienen caracteres innecesarios para la clasificación de documentos, como signos de puntuación o caracteres especiales. En tareas como detección de autoría pueden ser útiles pero en general agregan ruido a los modelos de clasificación de textos.
- 5.- **Stemming:** Es el proceso de convertir palabras a su forma base o raíz (e.g., morfema base del significado) (Kamath, Liu, y Whitaker, 2019). Uno de los algoritmos más populares es el algoritmo de Porter (Porter, 2001).
- 6.- **Lematización:** Es el proceso de convertir palabras a su forma base o raíz, a diferencia del proceso de stemming es que en este proceso el significado y el contexto puede ser conservado. La desventaja es que este método requiere un diccionario y tablas de búsqueda (Kamath, Liu, y Whitaker, 2019).
- 7.- **Otras técnicas:** Adicionalmente a las técnicas descritas también es posible pre-procesar los textos para intentar normalizarlos con la intención de facilitar al clasificador la identificación de patrones. Algunas de ellas son: la corrección de errores ortográficos, enmascaramiento de textos, etiquetado de partes de la oración, etc.

2.2.2. N-Gramas

Es una técnica para extraer características para representar un texto, los n-gramas son un conjunto de palabras o caracteres que respetan el orden de aparición en el texto; el número n indica la longitud de la secuencia a considerar, lo más común es utilizar valores de n pequeños (uni-gramas, bi-gramas, tri-gramas) (Kowsari et al., 2019).

Ejemplo de bi-gramas:

Texto Original: “*Con el tiempo todo pasa*”

Bi-gramas: {“con el”, “el tiempo”, “tiempo todo”, “todo pasa”}

2.2.3. Bolsa de palabras BoW

El modelo de bolsa de palabras o BoW (por sus siglas en inglés “Bag of Words”) es una representación simplificada de un texto. Normalmente, se utiliza un criterio o pesado específico, como lo puede ser la frecuencia de cada palabra, para representar cada texto. Es decir, en el modelo BoW, el conjunto de documentos es representado mediante una matriz de pesos, siendo las columnas palabras únicas del conjunto de datos y las filas representan un documento. Este modelo es muy simple, donde no es posible capturar el orden secuencial de las palabras -como en una oración o un documento-, con lo que las relaciones semánticas entre las palabras se pierden. Sin embargo, en este modelo las palabras capturan el contenido de un documento y esta representación puede ser utilizada para determinar el tema principal de dicho documento (Kowsari et al., 2019).

Pesado de palabras

La forma más básica de pesado de características es mediante el pesado TF (*term frequency* por sus siglas en inglés), el cual consiste en contar el número de ocurrencias de cada palabra en el conjunto de datos. Los métodos basados en TF generalmente consisten en representar la frecuencia de palabras como un peso escalado o normalizado, aunque es de fácil implementación y muy intuitivo este método está limitado por el hecho de que las palabras más comunes pueden dominar la representación.

TF-IDF Term Frequency-Inverse Document Frequency

Esta técnica de pesado fue propuesta por (Jones, 1972), con el objetivo de mitigar el efecto de las palabras más comunes en el corpus. IDF asigna menos peso a aquellas palabras que por estar presentes en la mayoría de los documentos de la colección no son útiles para identificar patrones discriminativos. La representación matemática del peso de un término en un documento por TF-IDF está dada en la ecuación 2.2.1

$$W(d, t) = TF(d, t) * \log(N/df(t)) \quad (2.2.1)$$

En donde N es el número total de documentos en la colección y $df(t)$ es el número de documentos que contienen el término t , el primer término $TF(d, t)$ es la frecuencia del término t en el documento d . Aunque TF-IDF trata de solucionar el problema de términos comunes en el documento, sigue sufriendo de otras limitaciones. Un problema común es que TF-IDF no se puede utilizar para medir la similitud entre palabras en el documento, dado que cada palabra es independiente representada por un índice.

2.2.4. Representaciones distribucionales

Su principal objetivo es capturar el significado semántico de las palabras, en donde cada palabra del vocabulario es representada mediante un vector n-dimensional de números reales. Recientemente (Mikolov et al., 2013) presentó el modelo Word2Vec, para generar vectores de palabras, el cual tiene dos algoritmos: el modelo CBOW y el modelo Skip-gram respectivamente. CBOW predice la palabra central del contexto que la rodea, mientras que Skip-gram hace lo contrario y predice la distribución (probabilidad) de las palabras de contexto de una palabra central. Word2Vec proporciona una herramienta muy poderosa para descubrir relaciones entre los textos de un corpus así como la similitud entre palabras.

Es un modelo similar Word2Vec, fue propuesto por (Pennington, Socher, y Manning, 2014) y su objetivo principal de capturar contextos globales combinando factorización de matrices y ocurrencias locales. Este modelo demostró ser más rápido en entrenamiento y mejora en tareas como analogía (en comparación con Word2Vec). Los autores generaron diferentes modelos pre-entrenados para su libre acceso, estos modelos fueron entrenados sobre grandes conjuntos de datos como CRAWL¹ y Wiki-

¹www.commoncrawl.org

pedia².

Otro modelo distribucional es el llamado FastText. Este modelo fue desarrollado por el laboratorio de inteligencia artificial de Facebook (Mikolov et al., 2017). A diferencia de los modelos anteriores este modelo considera la morfología de las palabras, cada vector es enriquecido con una bolsa de vectores de caracteres de n-gramas que es derivada de una matriz de co-ocurrencia. La principal ventaja de este modelo es su habilidad de obtener vectores para palabras fuera del vocabulario. Los vectores pre-entrenados están disponibles en la página oficial de FastText³ y la última liberación fue un modelo entrenado en 157 idiomas (Grave et al., 2018).

Para un mayor detalle de la teoría y el cálculo de este tipo de representaciones consultar el capítulo 5 del libro de (Kamath, Liu, y Whitaker, 2019).

2.3. Selección de características

Un problema común en la clasificación de textos es el manejo de grandes espacios vectoriales (cientos de miles dependiendo de la extracción utilizada) y como consecuencia se necesita grandes cantidades de memoria y tiempo de computación para poder procesar los algoritmos de aprendizaje. Una solución efectiva consiste en seleccionar las características que mejor discriminan a las clases.

Existen diversos métodos para seleccionar características entre los más utilizados se encuentran:

- 1.- Umbral de Frecuencia: Se mide para cada palabra w del vocabulario, el número de documentos en que w aparece. Aquellas palabras poco usadas, es decir, con pocos contextos de uso se eliminan. Es de esperar, que las palabras más frecuentes también aparezcan en documentos no vistos (Yang and Pedersen, 1997).
- 2.- Ganancia de información: Se establecen como relevantes todas aquellas palabras con una GI mayor a cero. Esta técnica tiene un sesgo muy importante respecto al conjunto de entrenamiento, sobretudo si éste es pequeño (Yang and Pedersen, 1997).

²www.wikipedia.org

³www.fasttext.cc

- 3.- Chi cuadrada:** Es un test estadístico que mide la independencia entre un término t y una clase c . Para cada término, una puntuación alta indica que la hipótesis nula de independencia debe ser rechazada y la ocurrencia del término y la clase son dependientes (Yang and Pedersen, 1997).

En (Yang and Pedersen, 1997) se encuentra la definición matemática de cada uno de los métodos y (Forman, 2003) presenta un análisis más extenso junto con otros métodos.

2.4. Algoritmos de clasificación

Se han utilizado diversos algoritmos de aprendizaje automático para la clasificación de textos, dentro de los más populares y frecuentemente utilizados como línea base son las máquinas de soporte vectorial o SVM por sus siglas en inglés. Por otro lado, respecto a las arquitecturas de aprendizaje profundo que se han empezado a emplear, se distinguen dos arquitecturas básicas: las redes recurrentes y las redes convoluciones. En esta sección se explican las generalidades de estos algoritmos. En (Minaee et al., 2020) se puede encontrar una revisión más detallada del estado del arte en la clasificación de textos con aprendizaje profundo.

2.4.1. Máquinas de Soporte Vectorial (SVM)

Este algoritmo de clasificación fue propuesto por (Vapnik and Chervonenkis, 1964), desde entonces el algoritmo ha pasado por una serie de mejoras. (Boser, Guyon, y Vapnik, 1992) adaptó el algoritmo para resolver problemas no lineales y la formulación moderna fue desarrollada por (Cortes and Vapnik, 1995).

Retomando las definiciones de (Boser, Guyon, y Vapnik, 1992), SVM encuentra una función de decisión para vectores x de características de dimensión n pertenecientes a alguna clase A o B. La entrada al algoritmo de entrenamiento es un conjunto de p ejemplos x_i con etiquetas y_i :

$$(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_p, y_p) \quad (2.4.1)$$

$$\text{donde } \begin{cases} y_k = 1 & \text{si } x_k \in A \\ y_k = -1 & \text{si } x_k \in B \end{cases}$$

Para los ejemplos de entrenamiento el algoritmo encuentra una función de decisión $D(x)$ durante una fase de aprendizaje. Después del entrenamiento, la clasificación de patrones desconocidos es predicha de acuerdo a la siguiente regla:

$$\begin{aligned} x \in A & \text{ si } D(x) > 0 \\ x \in B & \text{ si no} \end{aligned} \tag{2.4.2}$$

Las funciones de decisión deben ser lineales en sus parámetros pero no están restringidas a dependencias lineales de x . Estas funciones pueden ser expresadas idénticamente a un *perceptron* (Block, Knight Jr, y Rosenblatt, 1962):

$$D(x) = \sum_{i=1}^N w_i \phi_i(x) + b \tag{2.4.3}$$

En la ecuación 2.4.3 ϕ_i son funciones predefinidas de x y w_i y b son los parámetros ajustables a aprender.

En la formulación de (Boser, Guyon, y Vapnik, 1992; Cortes and Vapnik, 1995) podemos encontrar como aproximar este tipo de funciones construyendo hiperplanos separados que maximicen un margen, mediante algoritmos de optimización numérica.

Las ventajas⁴ de SVM son:

- Son efectivas en espacios altamente dimensionales, como es el caso en la clasificación de textos.
- Son efectivas en casos en donde el número de dimensiones es mayor al número de ejemplos.
- Usa un subconjunto de puntos de entrenamiento en la función de decisión llamados vectores de soporte, así que es también eficiente en el uso de la memoria.
- Se pueden especificar diferentes funciones de núcleo para la función de decisión.

⁴Listadas en <https://scikit-learn.org/stable/modules/svm.html>

Las desventajas incluyen:

- Si el número de características es mucho mayor al número de ejemplos, elegir una función de núcleo y término de regularización es crucial para evitar el sobre ajuste.
- Las SVMs no otorgan directamente estimaciones de probabilidad, estas son calculadas utilizando validación cruzada.

2.4.2. Redes neuronales profundas

En la definición de (LeCun, Bengio, y Hinton, 2015), el aprendizaje profundo permite a los modelos computacionales que están compuestos de múltiples capas de procesamiento aprender representaciones de los datos con múltiples niveles de abstracción. Estos métodos han mejorado dramáticamente el estado del arte en el varias tareas en el entendimiento de lenguaje natural, particularmente, la clasificación de textos, el análisis de sentimientos, dar respuesta a preguntas y traducción automática. El aprendizaje profundo descubre estructuras en grandes conjuntos de datos utilizando el algoritmo de retro-propagación para determinar sus parámetros internos que son utilizados para calcular la representación en cada capa de la arquitectura.

A continuación se describen los conceptos principales en aprendizaje profundo tomados principalmente de (LeCun, Bengio, y Hinton, 2015), para una definición más extensa consultar (Goodfellow, Bengio, y Courville, 2016; Kamath, Liu, y Whitaker, 2019).

Retropropagación y entrenamiento de arquitecturas multicapa

Una arquitectura multicapa es una pila de modelos simples y muchos de estos calculan un mapeo entrada-salida no lineal. El procedimiento de retro-propagación que calcula el gradiente de una función objetivo con respecto a los pesos de múltiples capas es una aplicación práctica de la regla de la cadena para derivadas.

Las arquitecturas multicapa aprenden a mapear una entrada de tamaño fijo (por ejemplo, la representación vectorial de un documento) a una salida de tamaño fijo (por ejemplo, las categorías de documentos). Para ir de una capa a la siguiente, un conjunto de unidades (nodos o neuronas) calculan una suma pesada de las entradas de la capa anterior y el resultado pasado es a través de una función no lineal. A la

fecha, la función ReLU (2.4.4) es una de las más populares ya que se ha demostrado empíricamente que permite aprender más rápido en redes neuronales con muchas capas (Glorot, Bordes, y Bengio, 2011). Las neuronas que no forman parte de la capa de entrada ni la de salida se conocen como unidades ocultas.

Las **capas ocultas** pueden ser vistas como una distorsión de la entrada en una forma no lineal para que las categorías sean linealmente separables por la última capa.

$$f(z) = \max(z, 0) \quad (2.4.4)$$

La **capa de entrada** puede ser construida vía pesado TF-IDF, vectores de palabras o alguna otra característica o representación. En la clasificación de documentos usualmente la capa de entrada recibe un documento en representación vectorial (2.4.5).

$$d_j = (w_{1,j}, w_{2,j}, \dots, w_{i,j}, \dots, w_{l_j,j}) \quad (2.4.5)$$

En donde l_j es el tamaño del documento j , y $w_{i,j}$ es la representación vectorial de la palabra i en el documento j .

En la **capa de salida** el número de neuronas es igual al número de clases para clasificaciones con más de dos clases y una para clasificación binaria, es la encargada de realizar la predicción final en base a las representaciones capturadas por la red neuronal.

La **inicialización** de los pesos y sesgo (bias) por lo general es con ceros o números generados con base en algún criterio, un algoritmo común es el conocido como inicialización Glorot o Xavier (Glorot and Bengio, 2010), el cual es basado en números aleatorios extraídos de una distribución de probabilidad uniforme.

Cuando se entrena una red neuronal, la propagación hacia adelante es realizada en lotes o **batch** (un número de instancias de entrenamiento determinado por el hiper-parámetro llamado tamaño de batch), después se calcula el error para cada neurona mediante retro-propagación en el mismo lote.

Una **época** es una sola iteración sobre todo el conjunto de datos, el número de épocas es un hiper-parámetro que determina cuantas veces el algoritmo de aprendizaje itera sobre todo el conjunto de entrenamiento.

Redes Neuronales Recurrentes (RNN)

Los modelos basados en RNNs tienen el objetivo principal de capturar la dependencia entre palabras y la estructura del texto para clasificación de textos Goodfellow, Bengio, y Courville (2016).

Las redes recurrentes procesan una secuencia u_t como entrada, un elemento a la vez, manteniendo en sus neuronas ocultas un “vector de estado” x_t que implícitamente contiene información acerca de todos los elementos anteriores a la secuencia procesada LeCun, Bengio, y Hinton (2015). La formulación general de este concepto es presentado en la ecuación 2.4.6, en donde x_t es el vector de estado en tiempo t y u_t se refiere a la entrada en el paso t .

$$x_t = F(x_{t-1}, u_t, \theta) \quad (2.4.6)$$

Las redes recurrentes son sistemas dinámicos muy poderosos, pero su entrenamiento es muy problemático dado que su gradiente al ser retro-propagado crece o disminuye en cada paso, así que sobre cada paso típicamente explotan o desaparecen (LeCun, Bengio, y Hinton, 2015), para tratar de solucionar este problema se propuso un tipo especial de red recurrente conocida como LSTM.

LSTM

Propuestas originalmente por (Hochreiter and Schmidhuber, 1997) para tratar los problemas de explotación y desvanecimiento de gradientes en redes recurrentes, las redes de memoria a corto y largo plazo (LSTM por sus siglas en inglés, Long Short Term Memory) son un tipo especial de red recurrente que conserva las dependencias largas de maneras más efectivas en comparación con una red recurrente básica. Las redes LSTM usan múltiples capas para regular el monto de información que será permitida en cada estado de nodo. La figura 2.1 muestra la estructura interna de una celda de una LSTM. En donde x_t , o y h_t representan la entrada, salida y el estado oculto en tiempo t . c_{t-1} es la información llevada del estado $t - 1$ al estado t , el cual será combinado con la entrada x_t y el estado oculto h_{t-1} para formar el estado oculto h_t y el arrastre de c_t el cual será enviado al siguiente paso.

dimensión de los vectores de palabras por d y el tamaño de la secuencia de entrada como s , entonces la dimensión de la matriz de la secuencia es sxd , entonces se puede tratar la matriz de la secuencia como una imagen y realizar convoluciones sobre esta vía filtros. Es razonable utilizar filtros con un ancho igual al la dimensión de los vectores de palabras (i.e., d). Así que solo varia la altura del filtro, i.e., el número de filas adyacentes identificado como **tamaño de filtro** o tamaño de núcleo Zhang and Wallace (2015).

Los filtros forman capas de convolución, estas capas de convolución son llamadas mapas de características y pueden ser apiladas para proporcionar múltiples filtros de la entrada. Para reducir la complejidad computacional, las CNNs utilizan una operación llamada **pooling** la cual reduce el tamaño de la salida de una capa a la siguiente en la red. Existen diferentes técnicas de pooling para reducir la salida y conservar características importantes. El pooling más empleado es el método de **max pooling**, el cual selecciona el elemento máximo en la ventana de pooling. Para pasar la salida de las capas de pooling, los mapas son *aplanados* en una columna. La capa final en una CNN es típicamente una capa totalmente conectada Kowsari et al. (2019).

2.5. Medidas de Evaluación

Existen diferentes métricas para evaluar los modelos de aprendizaje computacional, dependiendo de lo que se quiera medir algunas de las más utilizadas son: exactitud, precisión, recuerdo y F1. Para ilustrar cada un de estas métricas, se parte de la clasificación binaria como ejemplo. La tabla 2.1, conocida como matriz de confusión, resume los resultados de un clasificador. A continuación se describen los principales conceptos⁵.

Tabla 2.1: Matriz de confusión.

	Predicción: 1	Predicción: 0
Actual: 1	VP	FN
Actual: 0	FP	VN

La matriz de confusión no es una medida como tal, pero las métricas de evaluación

⁵Tomados de <https://medium.com/analytics-vidhya/complete-guide-to-machine-learning-evaluation-metrics-615c2864d916>

están basados en los números dentro de esta.

Términos asociados a la matriz de confusión:

- 1.- **Verdaderos Positivos (VP):** Es el número total de predicciones en que la clase actual es 1 (positiva) y la predicción también es 1 (positiva).
- 2.- **Verdaderos Negativos (VN):** Es el número total de predicciones en que la clase actual es 0 (negativa) y la predicción también es 0 (negativa).
- 3.- **Falsos Positivos (FP):** El número de predicciones en que la clase verdadera es 0 y la predicción es 1:
- 4.- **Falsos Negativos (FN):** Es el número de predicciones en que la clase verdadera es 1 y la predicción es 0.

El escenario ideal sería que el modelo obtenga 0 falsos positivos y falsos negativos, pero ese no es el caso en la vida real y el objetivo en ocasiones es tratar de minimizar los falsos positivos o los falsos negativos.

Exactitud: La proporción del número de ejemplos en el conjunto de evaluación que son correctamente clasificados por el modelo. La exactitud es utilizada cuando las clases tienen la misma importancia para la clasificación.

$$Exactitud = \frac{VP + VN}{VP + VN + FN + FP} \quad (2.5.1)$$

Precisión: Es la proporción del número de instancias positivas predichas correctamente.

$$Precision = \frac{VP}{VP + FP} \quad (2.5.2)$$

Recuerdo: Representa la proporción de las instancias positivas que lograron ser recuperadas.

$$Precision = \frac{VP}{VP + FN} \quad (2.5.3)$$

Medida F1: Es el promedio armónico de precisión y recuerdo. Proporciona una puntuación más realista al considerar a la precisión y el recuerdo.

$$F_1 = \frac{2 * precision * recuerdo}{precision + recuerdo} \quad (2.5.4)$$

Trabajo Relacionado

Descubrir los rasgos de un autor anónimo es de interés para la comunidad científica en procesamiento de lenguaje natural. Existen numerosas razones, una de ellas es aprovechar el constante flujo de información en redes sociales para entender mejor el lenguaje coloquial de uso diario, hacer que nuestras máquinas puedan identificar emociones, estados de ánimo, el género y edad del usuario, etc. Muchos esfuerzos y avances se han realizado en la última década.

En este capítulo se presenta, por un lado, el trabajo previo en área de perfilado de autor, en específico, los foros de evaluación PAN@CLEF y eRISK. Por otro lado, se presenta una revisión general del aumento de datos en tareas de clasificación de textos. Cabe aclarar que no existen estudios específicamente orientados al aumento de datos en la tarea de perfilado de autor.

3.1. Perfilado de autor

Los trabajos en perfilado de autor se han enfocado a identificar diversos rasgos de los autores: género, edad, nivel educativo, ocupación, rasgos de personalidad, tendencia política. Incluso han ido más allá al tratar de determinar características de comportamiento y condiciones médicas (trastornos como la anorexia o la depresión clínica).

Los primeros trabajos, motivados por la sociolingüística, utilizaron documentos formales: libros, ensayos y/o noticias; variando el tamaño del corpus estudiado de docenas a cientos de documentos. Uno de los primeros trabajos en perfilado de autor, usando medios automáticos, fue presentado por (Pennebaker, Mehl, y Niederhoffer, 2002). Los investigadores presentan evidencia que liga el uso de las palabras con

aspectos de personalidad, situaciones sociales y psicológicas.

(Argamon et al., 2009) demostró que se puede conocer el género, edad, lengua nativa y personalidad con un buen margen de exactitud, a través de ensayos personales de estudiantes. Las características relevantes encontradas fueron estilísticas, por ejemplo, el uso de pronombres, preposiciones y verbos modales.

En la actualidad, la investigación se ha enfocado determinar el perfil del autor utilizando datos extraídos de redes sociales, blogs y foros en línea. A continuación se presentan algunos de los trabajos más relevantes para esta tesis, específicamente los presentados en las conferencias PAN@CLEF siendo una de las pioneras en su tipo al incluir el perfilado de autor en todas sus ediciones.

PAN@CLEF

El mayor evento anual en perfilado de autor PAN es parte de las competencias organizadas bajo el marco del CLEF (Rangel et al., 2013; Rangel and Rosso, 2019). En este evento se ha estudiado el perfilado de autor desde una perspectiva multi-idioma, siendo el idioma inglés y español los más frecuentes. Las características recurrentes a perfilar ha sido género, edad y personalidad (Rangel et al., 2013; Rangel and Rosso, 2019; Rangel et al., 2016; Stammatatos et al., 2015). La mayoría del trabajo existente se distingue por (i) el pre-procesamiento, (ii) la extracción de características o (iii) el método de clasificación.

La técnica más común de **pre-procesamiento** entre los participantes es remover o enmascarar elementos específicos de las redes sociales (*hashtags*, menciones de usuario, enlaces a páginas web, emoticones) (Daneshvar and Inkpen, 2018; Jimenez-Villar et al., 2019; Pizarro, 2019). Además de convertir las palabras a minúsculas, utilizar lematización o *stemming*; algunos participantes remueven puntuación, palabras de paro y caracteres especiales.

En cuanto a la **extracción de características** los n-gramas de caracteres y palabras son ampliamente usados, en efecto las mejores soluciones propuestas para el perfilado de género en el PAN 2017, 2018 y 2019 (Basile et al., 2017; Daneshvar and Inkpen, 2018; Pizarro, 2019) utilizaron un ensamble de n-gramas de caracteres y n-gramas de palabras. En estos trabajos se ha identificado que una representación mediante n-gramas de caracteres puede ser capaz de capturar fragmentos relacionados a la estructura y estilo del texto. Algunas implementaciones también han propuesto esquemas de pesado inspirados en *tf-idf* poniendo énfasis en el estilo y contenido de

los textos.

Con respecto a los **algoritmos de clasificación** existe una gran cantidad de enfoques, siendo lo más común el algoritmo de máquinas de soporte vectorial (SVM). Un punto importante a destacar, es que a partir del año 2018 se han presentado algunas propuestas utilizando aprendizaje profundo. Sin embargo, hasta la fecha no han podido superar a los algoritmos tradicionales. La primera vez que un enfoque de aprendizaje profundo, concretamente una arquitectura CNN, aparece entre los primeros lugares (lugar once) fue en la conferencia PAN 2019 (Rangel and Rosso, 2019). Es importante notar, que ha diferencia del enfoque tradicional, en el caso de los modelos basados en redes neuronales, la extracción de características está implícita en su arquitectura.

3.1.1. Detección de trastornos mentales en redes sociales

Uno de los primeros estudios en la detección de depresión mediante medios automáticos (Rude, Gortner, y Pennebaker, 2004) encontró que los participantes deprimidos utilizan más palabras negativas y el uso del pronombre “yo” (“I” en inglés) más que los participantes no deprimidos.

Moviéndose a la investigación en redes sociales (De Choudhury et al., 2013) empleo *crowdsourcing* (una forma de colaboración empleando a múltiples personas a través de internet) para obtener un conjunto de usuarios de Twitter quienes reportaron ser diagnosticados con depresión clínica, en este trabajo demostraron el uso potencial de Twitter como una fuente de información para medir signos de depresión mayor en individuos. Encontraron que los individuos con depresión muestran baja actividad social, emociones negativas, auto atención, expresión de preocupaciones médicas y relacionales, además de pensamientos religiosos. Estos atributos fueron considerados para construir un clasificador SVM alcanzando una precisión de 70 %. Estos resultados demostraron la factibilidad de detectar la depresión en redes sociales.

Sin embargo, la creación de colecciones de documentos para abordar este tipo de problemas es costosa y difícil. Bajo estas condiciones y debido al particular interés en el perfilado de características de comportamiento (Kumar et al., 2018) y trastornos mentales (De Choudhury et al., 2013), se han desarrollado conferencias y competencias específicas. Una de ellas es la conferencia eRISK (Losada, Crestani, y Parapar, 2018). El principal objetivo de este foro es la detección temprana de un trastorno a través

de los historiales de comunicación de un usuario en blogs. Independientemente del enfoque de detección temprana también es de interés tratar la detección considerando todo el historial de un usuario como un solo documento.

El eRISK 2018 presentó dos tareas: detección de depresión y detección de anorexia. En ambos casos se trata de un problema de clasificación no temática con datos desbalanceados.

eRisk 2018

En la edición eRisk 2017 (Losada, Crestani, y Parapar, 2017, 2018), los organizadores construyeron un conjunto de datos con publicaciones de usuarios deprimidos y no deprimidos extraídos de la red social Reddit. En eRisk 2018 se complementó el conjunto original con más usuarios y se agregó la tarea de detección de anorexia. Para abordar estas tareas se evaluaron un total de 45 contribuciones de diferentes instituciones, algunas de las propuestas dieron un tratamiento estándar experimentado con diferentes características, LDA, n-gramas de palabras y diferentes esquemas de pesado (Cacheda et al., 2018; Almeida, Briand, y Meurs, 2017; Ortega-Mendoza et al., 2018b). Mientras que muy pocos utilizaron enfoques de aprendizaje profundo: (Trotzek, Koitka, y Friedrich, 2018; Wang, Huang, y Chen, 2018; Liu et al., 2018).

El equipo TUA1 (Liu et al., 2018) además de presentar un modelo construido con una SVM Lineal, pesado *tf-idf* y normalización *l2*, también construyeron un modelo basado en una arquitectura compuesta de una red CNN -que actúa como extractor de características- y una LSTM. En su configuración experimental utilizaron una longitud de entrada de 2000 tokens, 64 filtros para la red CNN de tamaño 5, *MaxPooling* de tamaño 4, un factor de 0.25 para *dropout* y *Relu* como función de activación. Para la fase de entrenamiento eligieron entropía cruzada binaria para la función de pérdida y el optimizador *Adam*. Mediante este modelo neuronal se obtuvieron 0.29 de *F1*, en la tarea de depresión y 0.36 para la detección de anorexia.

Los investigadores que conformaron el equipo TBS (Wang, Huang, y Chen, 2018) abordaron las tareas como un problema de clasificación de oraciones y presentaron un modelo basado en CNN en combinación con un pesado *tf-idf*; obteniendo 0.26 de *F1* para la detección de depresión y 0.67 para anorexia.

El equipo ganador (Trotzek, Koitka, y Friedrich, 2018) FHDO-BCSG presento 5 modelos de clasificación diferentes, para la detección de depresión. El mejor modelo fue un ensamble de bolsas de palabras BOW, con diferentes tipos de pesado y

n-gramas, el algoritmo de clasificación utilizado fue regresión logística utilizando un peso modificado para cada clase para incrementar el costo de los falsos negativos; obteniendo 0.64 de $F1$ para depresión y 0.81 para la detección de anorexia. Además presentaron un modelo basado en una *CNN* utilizando vectores *FastText* de 300 dimensiones, entrenados con documentos extraídos de un corpus de 1.37 billones de comentarios en Reddit, una longitud de entrada de 100 tokens, una capa de convolución, 100 filtros con altura igual a 2 y con un ancho correspondiente al tamaño de los vectores de palabras, max pooling de tamaño 1 y CReLU como función de activación; resultando en un vector de 200 dimensiones por documento que es propagado a través de cuatro capas totalmente conectadas. El entrenamiento fue realizado utilizando el optimizador *Adam* para minimizar la entropía cruzada, mediante un tamaño de batch de 10,000 documentos de 100 palabras y una tasa de aprendizaje de $1e - 4$ durante 30 épocas. Este modelo logró obtener una puntuación $F1$ de 0.54 para la detección de depresión y 0.81 para anorexia. Agregando características extraídas manualmente lograron mejorar la puntuación $F1$ a 0.85 para la detección de anorexia.

En general la tarea de detección de depresión fue la más difícil y de un total de 45 modelos evaluados la puntuación $F1$ promedio fue de 0.42 mientras que para anorexia 0.56 indicando que aún falta mucho por mejorar en estas tareas.

eRisk 2019

El propósito general de esta conferencia fue evaluar las metodologías y técnicas empleadas para la detección temprana de signos de depresión (Losada, Crestani, y Parapar, 2019a). A diferencia de la edición anterior la tarea de detección de depresión tuvo un nuevo enfoque, fue orientada a analizar las publicaciones de un usuario en redes sociales con el objetivo de extraer evidencia útil para estimar el nivel de depresión de un usuario mediante el completado automático de un cuestionario basado en el inventario de Beck Beck et al. (1961).

En esta edición destacó un mayor uso de modelos de aprendizaje profundo para tratar el tema como un problema de clasificación. Los modelos más utilizados fueron basados en redes recurrentes y convolucionales. Algunos participantes se enfocaron en conseguir más datos de entrenamiento, como por ejemplo: tomar los datos etiquetas de ediciones anteriores, incorporar información extensa y recuperar datos recavados de la red social Reddit. Es importante hacer notar que los equipos que utilizaron más evidencia para construir sus modelos de aprendizaje obtuvieron los mejores resultados.

El equipo CAMH (Abed-Esfahani et al., 2019) alcanzó una mayor puntuación en predecir la categoría correcta de nivel de depresión utilizando el modelo de lenguaje pre-entrenado GPT-1 como extractor de características y adicionado características de la herramienta LIWC (Pennebaker, Francis, y Booth, 2001) . Utilizando un enfoque no supervisado, lograron obtener un 45 % en la métrica DCHR (Indica la fracción de casos en donde el cuestionario automático indica una categoría de depresión que es equivalente a la categoría del cuestionario real).

El equipo UNSL (Burdisso, Errecalde, y Montes-Y-Gómez, 2019) obtuvo el mejor desempeño para predecir el cuestionario a nivel pregunta obteniendo una puntuación de 41.43 % en AHR (El promedio de todos los usuarios en el que el cuestionario automático tiene exactamente la misma respuesta que el real) y 40 % en DCHR. Esta propuesta consistió en utilizar los datos de entrenamiento de la tarea eRisk 2018 y un algoritmo de clasificación diseñado específicamente para las tareas de detección temprana.

Aunque la efectividad de los modelos es aún modesta, los experimentos sugieren que la evidencia adicional extraída de redes sociales es útil y las herramientas automáticas o semi-automáticas pueden ser diseñadas para detectar individuos en riesgo (Losada, Crestani, y Parapar, 2019b).

3.2. Aumento de datos

El aprendizaje profundo típicamente requiere grandes cantidades de datos etiquetados para tener éxito. El aumento de datos promete resolver el problema de la necesidad de más datos etiquetados, básicamente consiste en aplicar una serie de transformaciones a un ejemplo original para obtener un nuevo dato a partir de éste.

El término **aumento de datos** se refiere a métodos para construir una optimización iterativa o algoritmos de muestreo mediante la introducción de datos no observados o variables latentes (Van Dyk and Meng, 2001). La idea del aumento de datos nació en problemas de **datos incompletos**, como una forma de completar las celdas faltantes en una tabla de contingencia balanceada (Dempster, Laird, y Rubin, 1977). El aumento de datos automático es mayoritariamente utilizado en tareas relacionadas a visión computacional y ayudan a realizar un entrenamiento más robusto particularmente cuando el tamaño de los datos es pequeño.

Derivado del estudio del estado del arte en aumento de datos, las técnicas de

aumento de datos se pueden clasificar en dos categorías (no restringidas a un solo dominio): aquellas que se basan en aprendizaje supervisado y las que utilizan un enfoque semi-supervisado. Los basados en un enfoque supervisado crean muestras nuevas a partir de datos previamente etiquetados y las que utilizan un enfoque semi-supervisado obtienen ejemplos totalmente nuevos en base un modelo supervisado, supervisado débil o heurísticas conociendo la naturaleza de los datos.

3.2.1. Aumento de datos supervisado

El objetivo es crear un nuevo y realista conjunto de entrenamiento aplicando una transformación a la entrada de un ejemplo. Conservando la etiqueta original del ejemplo. Formalmente, sea $q(\hat{x}|x)$ la transformación de aumento de la cual podemos extraer ejemplos aumentados \hat{x} basado en un ejemplo original x . Para que una transformación de aumento sea válida es requerido que cualquier ejemplo $\hat{x} \sim q(\hat{x}|x)$ extraído de la distribución comparta la misma etiqueta de verdad que x , es decir $y(\hat{x}) = y(x)$. Dada una transformación de aumento válida, simplemente se puede minimizar la probabilidad negativa de los ejemplos aumentados (Xie et al., 2019).

El aumento de datos supervisado puede ser equivalentemente visto como construir un conjunto aumentado etiquetado del conjunto original y entrenar el modelo en el conjunto aumentado. El punto crítico es como diseñar esa transformación, en la literatura podemos encontrar dos grupos de algoritmos para **crear** ejemplos de entrenamiento adicionales: los que operan **a nivel estructural**, los cuales crean transformaciones en un ejemplo (imagen, cadena de caracteres, texto, etc.) (Zhong et al., 2017), y **sobre muestreo sintético** creando ejemplos adicionales a nivel características es decir en un espacio vectorial (Chawla et al., 2002).

3.2.2. Aumento de datos semi-supervisado

Estos métodos tienen como característica general el aprender un modelo inicial para posteriormente etiquetar datos nuevos obtenidos de algún dominio similar y re-entrenar el modelo con estos datos nuevos. Tomando la definición de (Xie et al., 2019), la forma general de estos trabajos puede ser resumida como sigue:

- Dada una entrada x , se calcula la distribución $p_\theta(y|x)$ dado x y una versión con ruido $p_\theta(y|x, \epsilon)$ mediante la introducción de un pequeño ruido ϵ . El ruido puede ser aplicado a x o estados ocultos.

- Minimizar una métrica de divergencia entre las dos distribuciones $D(p_\theta(y|x)||p_\theta(y|x, \epsilon))$.

Este procedimiento fuerza el modelo a ser insensible al ruido ϵ y suave con respecto a los cambios en el espacio de entrada. Desde otra perspectiva, minimizando la pérdida de consistencia gradualmente se propaga la información de la etiqueta de ejemplos etiquetados a ejemplos no etiquetados (Miyato et al., 2019).

3.2.3. Aumento de datos en clasificación de textos

El aumento de datos ha sido ampliamente utilizado en tareas de visión computacional (Cubuk et al., 2019), pero menos en tareas de procesamiento de lenguaje natural, en años recientes ha crecido el interés por proponer diversas técnicas para el aumento de datos en la clasificación de textos. A continuación se mencionan algunos de los métodos más relevantes para este trabajo.

Basados en métodos semi-supervisados

Datos con ruido: (Hedderich and Klakow, 2018) propusieron una capa de ruido que es agregada a una arquitectura de red neuronal. Lo que permite modelar el ruido y entrenar una combinación de datos limpios y con ruido, para simular escenarios de pocos recursos el entrenamiento fue realizado con diferentes tamaños de datos limpios, variando desde un 1 % del conjunto original hasta un 10 % (equivalentes de 407 ejemplos y 20,362 respectivamente). Comprobando que en un contexto de bajos recursos reduciendo el conjunto original hasta un 1 %, en la tarea de reconocimiento de entidades nombradas (NER), la clasificación puede mejorar en términos de $F1$ en promedio hasta 10 puntos en $F1$ mediante el uso adicional de datos con ruido y manejando el ruido. Variando el tamaño del conjunto original a un 10 % la ganancia obtenida no se observa por lo que se llega a la conclusión de que un 10 % de datos limpios puede ser suficiente para entrenar el modelo y el ruido adicional puede perjudicar al modelo.

Reinforced Co-Training: (Wu, Li, y Wang, 2018), este método utiliza el algoritmo Q-learning para aprender una política de selección de datos y entonces explotar esta política para co-entrenar clasificadores automáticamente. Realizaron experimentos en la detección de *Clickbait*; este término se refiere a aquellos encabezados con el objetivo de atraer la atención del lector, pero los documentos usualmente tienen menos relevancia con los encabezados correspondientes. El etiquetado de este tipo de

datos consume mucho tiempo y labor. En esta tarea lograron mejorar 3 puntos en términos de la métrica $F1$ en comparación con el modelo base entrenado en forma supervisada.

(Han, Gao, y Ciravegna, 2019) propusieron una técnica de aumento de datos la cual consiste en incorporar ejemplos nuevos al conjunto de entrenamiento mediante un etiquetado basado en la búsqueda de similitudes relacionales en millones de tweets no etiquetados. Realizaron experimentos para la detección de rumores en redes sociales, logrando incrementar en promedio la métrica $F1$ entre 9 y 12 puntos en comparación a no realizar aumento de datos.

UDA (Xie et al., 2019): Es una propuesta híbrida la cual consiste en utilizar métodos existentes de aumento de datos, reemplazo de sinónimos y traducción inversa, para aumentar datos etiquetados y no etiquetados. Mediante el entrenamiento fino del modelo no supervisado BERT lograron aproximar el error de clasificación en 4 conjuntos de datos para la clasificación de opiniones con un margen de un punto porcentual, en comparación con el modelo entrenado en el conjunto completo de datos etiquetados. Con esto se logró comprobar que aún existe una brecha por rebasar cuando se comparan los métodos supervisados con los semi-supervisados.

Por lo general los esquemas para realizar aumento de datos de forma semi-supervisa han requerido de modelos complejos para poder implementarse. Si bien los resultados son prometedores y comparables al estado del arte, no han logrado superar el estado del arte basado en modelos supervisados.

Basados en aprendizaje supervisado

(Zhang, Zhao, y LeCun, 2015) presentaron una exploración empírica de redes convolucionales a nivel carácter. Construyeron conjuntos de datos aumentados para la clasificación de opiniones, mediante el reemplazo de palabras por sus sinónimos utilizando un tesauro. Llegando a reducir el error de clasificación en 1% menos en comparación con el estado del arte, agregando aumento de datos en cuatro de ocho conjuntos de datos.

Aumento de datos contextual (Kobayashi, 2018): Asumen que el sentido de las oraciones no cambia incluso si las palabras en las oraciones son reemplazadas por otras palabras con relaciones paradigmáticas. Este método, estocásticamente reemplaza palabras con otras palabras que son predichas por un modelo de lenguaje bi-direccional. Además proponen un modelo de lenguaje condicionado a la etiqueta que permite

al modelo aumentar oraciones considerando la información de la etiqueta. Mediante experimentos en 6 conjuntos de datos de clasificación de textos logran mejorar la exactitud en 1 % en comparación a no realizar aumento de datos y menor a 1 % en comparación con el remplazo de sinónimos.

EDA (Wei and Zou, 2019): se presenta como una alternativa simple y escalable en comparación con métodos de aumentos de datos basados en redes neuronales, EDA consiste de una combinación de cuatro operaciones a nivel palabra: reemplazo de sinónimos, inserción aleatoria, intercambio aleatorio y eliminación aleatoria. En cinco tareas de clasificación, muestran que es posible mejorar el rendimiento en redes convolucionales y recurrentes, alcanzado entre un 1 y 2 % en comparación de modelos sin aumento de datos.

Paráfrasis neuronal (Kumar et al., 2019): Este trabajo propone un método para obtener paráfrasis neuronales mediante el modelo seq2seq, a diferencia de otros modelos para generar paráfrasis este método busca un balance entre la diversidad y la fidelidad de las oraciones generadas; para esto proponen optimizar un función que combine estos dos factores. Los autores evaluaron su propuesta para la clasificación de intención utilizando una red LSTM y regresión logística, obteniendo una mejora de 3 % en exactitud sobre el método base que es no realizar aumento de datos y 2 % sobre reemplazo de sinónimos.

Traducción de temas (Zhang, Lertvittayakumjorn, y Guo, 2019): Este método traduce todas las palabras reemplazables de una oración a otras clases objetivo. Esta búsqueda de relaciones de similitud se realiza utilizando aritmética de vectores. Realizaron diversos experimentos para la clasificación de documentos mediante *zero-shot text classification*, esta técnica de clasificación consiste en ser capaz de predecir categorías no vistas en la fase de entrenamiento. Mediante un esquema controlado de poco recursos logran obtener ganancias de 1 a 8 % en términos de exactitud, en comparación a no realizar aumento de datos.

3.2.4. Discusión del trabajo relacionado

Al revisar la literatura de los métodos de aumento de datos basados en un enfoque supervisado, podemos observar que son un tanto complejos y en muchos casos, bajo un esquema de experimentación controlando la cantidad de datos etiquetados disponibles, no logran superar a los modelos supervisados.

Todos los trabajos hasta ahora encontrados en la literatura de aumento de datos mediante un enfoque supervisado, están enfocados a la clasificación de textos cortos o clasificación temática, pero ni una enfocada a tareas de perfilado de autor o demostrado ser efectivos en conjuntos desbalanceados. En algunos casos como EDA el reemplazo es totalmente aleatorio o la estructura del documento se corrompe al incorporar operaciones de eliminación sobre las palabras, en otros como el reemplazo de sinónimos no siempre se asegura que la palabra a reemplazar pertenezca a la misma categoría que la palabra original. Los trabajos que respetan la estructura y diversidad de la oración original están basados en modelos de redes neuronales pero es difícil hacerlos escalables. En la tabla 3.1 se presentan las principales características de los diferentes enfoques supervisados relevantes para este trabajo en comparación con el método propuesto. La propuesta de Kumar et al. (2019) puede considerarse que se respeta el estilo contenido de los textos mediante la realización de una paráfrasis neuronal, pero este enfoque tiene la desventaja de utilizar un conjunto de datos externos para aprender a realizar la paráfrasis y considerando que se cuente con este recurso, el tiempo tomado para realizar un paráfrasis neuronal o predecir palabras mediante un modelo de lenguaje hace que el método no sea escalable.

En el caso de perfilado de autor, es necesario que los nuevos ejemplos aumentados respeten tanto el estilo (i.e., la estructura original) como el contenido del texto, por lo que en este trabajo se proponen métodos de aumento de datos que consideren el estilo y contenido del documento original; considerando estilo como la forma o modo de expresar el contenido, siendo el contenido el tema o mensaje a transmitir.

Los resultados hasta ahora alcanzados muestran un beneficio del uso del aumento de datos, no obstante, estos beneficios aún modesto. Por otro lado, las técnicas simples de aumento de datos a nivel palabra han demostrado ser efectivas y escalables, y obtienen resultados comparables a técnicas complejas como la paráfrasis neuronal o modelos de lenguaje.

Tabla 3.1: Comparación del método propuesto con el estado del arte en aumento de datos para clasificación de textos.

Metodo	Clasificación	Datos desbalanceados	Recurso externo	Parafrasis neuronal o modelo de lenguaje	Estilo	Contenido
Zhan2015	Minería de opiniones	No	Tesaurus	No	No	Si
Kobayashi2018	Temática	No	Glove	Si	No	Si
EDA	Minería de opiniones	No	Tesaurus	No	No	Si
Kumar2019	Dialogo	No	Datos alineados	Si	Si	Si
Zhang2019	Temática	No	Glove	No	Si	No
Propuesta	Perfilado	Si	Glove	No	Si	Si

Método propuesto

En el capítulo 3 se describieron los principales retos del perfilado de autor y las principales características de los métodos existentes para el aumento de datos en la clasificación de textos. Observando las características de los métodos de aumento de datos existentes, en este capítulo se describen los métodos propuestos para realizar el aumento de datos para tareas relacionadas al perfilado de autor.

Como se mostró en el capítulo anterior, existen numerosas técnicas de aumento de datos sin embargo no todas son generalizables o escalables, además de que en el caso de perfilado de autor se desea conservar tanto el estilo como el contenido de un texto. De ahí que la mejor forma de realizar el aumento de datos en esta tarea es mediante el parafraseo humano, pero debido a su costo, no siempre es posible. Sin embargo nos podemos apoyar de técnicas automáticas para aproximarse al parafraseo (Androutsopoulos and Malakasiotis, 2010).

El aumento de datos propuesto busca aproximarse al parafraseo de las frases originales y con ellas incrementar el conjunto original. Los métodos desarrollados en este trabajo tienen dos pasos generales que se describen a continuación:

- 1.- **Selección de palabras a reemplazar:** El primer paso consiste en identificar el subconjunto de palabras a reemplazar. Dos criterios son relevantes en esta etapa: (i) la importancia de la palabra en la estructura de la frase, es decir, no se tiene el mismo efecto si se reemplaza una palabra de contenido que una palabra funcional; (ii) la cantidad de palabras a reemplazar, dado que se desea conservar la misma interpretación de la frase original el número de reemplazos deberá controlarse.
- 2.- **Reemplazo de palabras seleccionadas:** Una vez determinado el subconjunto de palabras a reemplazar, se deberán identificar nuevas palabras que no alte-

ren significativamente el sentido de la frase original. Para ello se recurre a un recurso externo, tal como un tesoro, un diccionario, o incluso a modelos distribucionales de vectores de palabras. Una vez identificadas las palabras sustitutas se reconstruye la secuencia manteniendo el orden original para formar la nueva secuencia aumentada.

4.1. Selección de palabras a reemplazar

El primer punto bajo este proceso es el cálculo del número de palabras a reemplazar. Para ello, se retomó el criterio presentado en (Zhang, Zhao, y LeCun, 2015). El número de palabras a reemplazar r se selecciona realizando un muestreo aleatorio de una distribución de probabilidad geométrica.

Una función de probabilidad geométrica es la función de probabilidad del número X del ensayo de Bernoulli de obtener éxito, soportado en el conjunto de los números naturales. Una distribución geométrica da la probabilidad de que la primera ocurrencia de éxito requiere de k ensayos independientes, cada uno con una probabilidad de éxito p . Si la probabilidad de éxito de cada ensayo es p , entonces la probabilidad de que en el k -ésimo ensayo, de k ensayos, sea el primer éxito es representada en la ecuación 4.1.1.

$$Pr(X = k) = (1 - p)^{k-1}p \quad (4.1.1)$$

Por lo tanto la aleatoriedad del número r es controlada mediante la modificación del parámetro p como se representa en figura 4.1. Como puede verse en dicha figura, mientras menor sea el valor de p se modificará un mayor número de palabras, agregando una mayor diversidad y consecuentemente incrementando la probabilidad de alterar el significado original.

4.1.1. Etiquetado de partes de la oración

El proceso de selección debe cuidar la modificación de ciertas partes de la oración para, por un lado, evitar perder la interpretación original, y por otro intentar conservar el estilo de la frase original. Por tal motivo, cada frase es etiquetada asignando a cada palabra su etiqueta POS (*part of speech*) correspondiente. La tabla 4.1 muestra un ejemplo del tipo de etiquetado aplicado. Gracias al etiquetado las palabras a

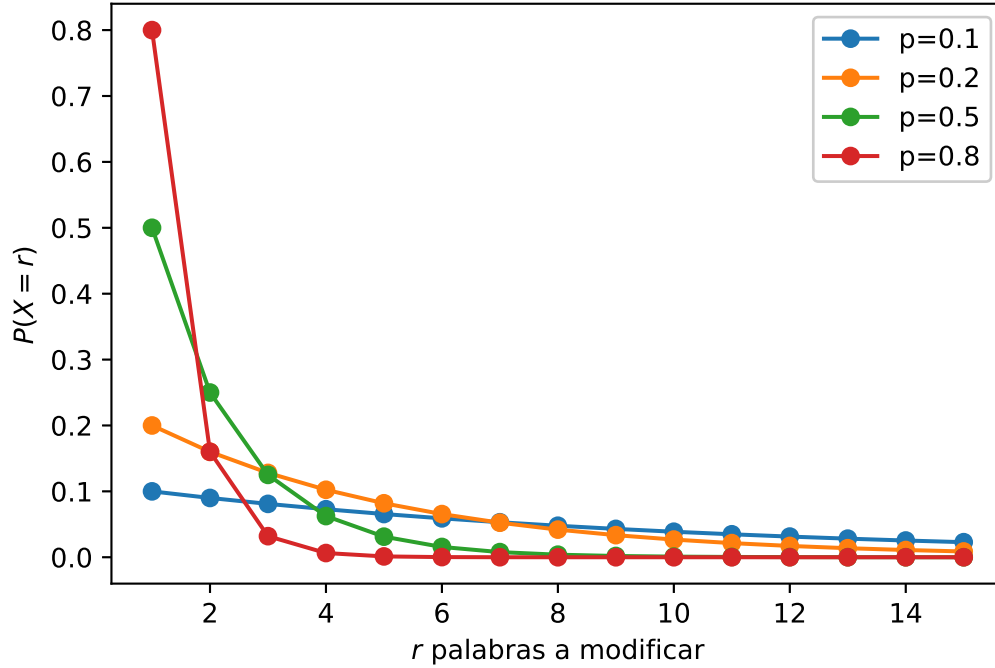


Figura 4.1: Función de masa para la distribución geométrica con diferentes valores de probabilidad.

reemplazar solo son aquellas que están más fuertemente asociadas al contenido de la frase, es decir, solo se seleccionan aquellas palabras con funciones gramaticales de: sustantivo, adjetivo, verbo y/o adverbio.

Tabla 4.1: Ejemplo de etiquetado de partes de la oración.

Secuencia	<i>I</i>	<i>am</i>	<i>running</i>	<i>out</i>	<i>of</i>	<i>ideas</i>
Etiqueta	PRP	VBP	VBG	IN	IN	NNS
Equivalencia	Pronombre	Verbo	Verbo	Preposición	Preposición	Sustantivo

4.1.2. Exclusión de palabras importantes

Además de las palabras funcionales también es deseable mantener palabras que aportan información para la tarea de clasificación que se desea realizar, por lo tanto

la primera propuesta consiste en evitar seleccionar, entre las palabras a reemplazar, aquellas palabras dependientes a la clase del documento.

Para este proceso recurrimos a la técnica de selección de características conocida como prueba de independencia χ^2 (Chi cuadrada). En estadística, la prueba χ^2 es aplicada para comprobar la independencia de dos eventos, donde los eventos A y B son definidos a ser independientes si $P(AB) = P(A)P(B)$ o, equivalente, $P(A|B) = P(A)$ y $P(B|A) = P(B)$. En la selección de características para la clasificación de textos, los dos eventos son: *ocurrencia del término* y *ocurrencia de la clase*. Posteriormente se ordenan los términos de mayor a menor respecto a la ecuación 4.1.2.

$$\chi^2(D, t, c) = \sum_{e_t \in \{0,1\}} \sum_{e_c \in \{0,1\}} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}} \quad (4.1.2)$$

El término e_t indica la ausencia o presencia del término t en el documento, similarmente el término e_c indica si el documento se encuentra en la clase c . N es la frecuencia observada en D y E es la frecuencia esperada. χ^2 mide por cuanto los conteos esperados E y los conteos observados N se desvían de cada uno. Un valor alto de χ^2 indica que la hipótesis de independencia, la cual implica que los conteos esperados y observados son similares, es incorrecta. Si los dos eventos son dependientes, entonces la ocurrencia del término hace la ocurrencia de la clase más probable (o menos probable), entonces el término debería ser seleccionado como relevante.

A través de este método se identifican todas aquellas palabras dependientes de la clase, y por ende, de relevancia para la tarea de clasificación. Dada la importancia de estas palabras se evitará reemplazarlas, excluyéndolas del proceso de selección de palabras a reemplazar.

4.2. Reemplazo de palabras seleccionadas

Una vez identificadas las palabras a reemplazar, el siguiente paso es, mediante la consulta de alguna fuente de conocimientos externa, buscar palabras candidatas similares a la palabra que se desea reemplazar. (Zhang, Zhao, y LeCun, 2015) proponen consultar un tesoro con el objetivo de obtener los sinónimos de una palabra, sin embargo el vocabulario contenido en el tesoro puede ser muy limitado o demasiado formal para el contexto del texto a aumentar.

Una alternativa es buscar palabras similares a través de representaciones distri-

bucionales de las palabras. Se ha demostrado que estas representaciones capturan similitudes relacionales, las cuales pueden ser recuperadas por medio de aritmética de vectores (Levy and Goldberg, 2014). El presente trabajo explora dos enfoques utilizando este recurso. La siguiente sección explica ambos enfoques.

4.2.1. Similitud relacional

La idea principal es reemplazar una palabra en una secuencia por una palabra similar o altamente relacionada ya que ambas se utilizan en contextos similares. Para realizar esto se recupera el vector de la palabra a reemplazar de un modelo pre-entrenado de vectores de palabras y se calcula la distancia respecto a cada vector en el modelo pre-entrenado. En este caso, hemos usado la medida de similitud coseno, véase la ecuación 4.2.1.

$$\cos(A, B) = \frac{A * B}{||A|| ||B||} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \sqrt{\sum_{i=1}^n (B_i)^2}} \quad (4.2.1)$$

donde A y B son los vectores n -dimensionales de las palabras a ser comparadas.

Específicamente para encontrar las palabras candidatas W , dada una palabra w se buscan las k palabras más similares a w de acuerdo a la ecuación 4.2.2.

$$\operatorname{argmax}_{v \in V} (\cos(v, w)) \quad (4.2.2)$$

En donde v es una palabra del vocabulario de vectores pre-entrenados excluyendo la palabra w . Una alta similitud coseno (cercana a 1) significa que los vectores comparten una dirección muy similar.

Reemplazo por palabras equivalentes

Al calcular la similitud coseno entre dos palabras dados sus vectores, es posible encontrar que palabras como *feliz* y *triste* suelen tener una similitud cercana. Esto sucede por que ambas palabras ocurren en contextos de uso similares. Sin embargo, no deseamos realizar este tipo de sustituciones, ya que sustituir una palabra por su antónimo (en lugar de un sinónimo) podría causar que la etiqueta original del documento se pierda.

Para resolver este problema en lugar de utilizar la ecuación 4.2.2, buscamos relaciones de similitud incluyendo pares de palabras. Por ejemplo, los vectores codificados

de palabras pueden capturar la relación de género en los pares de palabras “hombre:rey”, “mujer:reina”. Mediante el uso de aritmética de vectores se puede expresar dicha relación a través de una analogía entre pares de palabras. Así la palabra “reina” puede ser identificada mediante la analogía “hombre es a rey como mujer es a ?”. De esta forma de manera general podemos expresar una analogía de la forma “ w_c es a w como w^* es a v ”, tal como se indica en las ecuaciones 4.2.3 y 4.2.4.

$$reina \approx rey - hombre + mujer \quad (4.2.3)$$

$$v \approx w - w_c + w^* \quad (4.2.4)$$

El objetivo principal es encontrar palabras candidatas v que comparten la misma relación reflejada en $w - w_c$ pero no necesariamente similar a w .

Para encontrar las palabras candidatas v utilizamos el método 3COSMUL (Levy and Goldberg, 2014), ecuación 4.2.5:

$$\operatorname{argmax}_{v \in V} \frac{\cos(v, \hat{w}_c) \cos(v, w)}{\cos(v, w_c) + \epsilon} \quad (4.2.5)$$

Siendo w la palabra a reemplazar, w_c la palabra representativa de la clase positiva y w^* una de las palabras más similares a w_c , V es el vocabulario de los vectores pre-entrenados, ϵ es un número pequeño para evitar una división por cero (puede ser 0.0001).

Reemplazo por palabras opuestas

En escenarios desbalanceados el aumento de datos puede llevarnos a un sobre muestreo de la clase minoritaria, generalmente la clase de interés. Los métodos presentados anteriormente fueron diseñados para aumentar esta clase minoritaria (a la cual también nos referimos como clase positiva). Sin embargo, una desventaja de realizar esto es que el modelo de aprendizaje se restringe al vocabulario de la clase positiva lo que provoca un sobre ajuste. En un intento por contrarrestar esta situación, se propone un método que incorpora documentos seleccionados de la negativa a la clase positiva. Por supuesto, para ello es necesario realizar una transformación de las instancias negativas.

Para llevar a cabo esta transformación adaptamos el método propuesto por

(Zhang, Lertvittayakumjorn, y Guo, 2019). En dicho método se aborda el problema de *zero-shot text classification* para ello toma un documento de una clase etiquetada y lo *traduce* para considerarlo como instancia de una clase totalmente nueva. Este método no tiene ninguna restricción respecto sobre las palabras a reemplazar, en nuestro caso hemos incluido un criterio de selección para guiar la generación de los nuevos documentos que servirán para aumentar el conjunto de la clase de interés.

La idea básica de la transformación recae en el mismo método visto en la sección anterior. Sin embargo, los pares de palabras usadas para representar la analogía son escogidas para identificar palabras con una relación opuesta. Así por ejemplo, en el caso de la tarea de detección de depresión, asociaremos la palabra “*feliz*” como representativa de la clase *no deprimidos* y la palabra “*triste*” para la clase *deprimidos*. Ahora bien, reemplazaremos palabras de documentos de la clase *no deprimidos* por palabras que presentan la relación opuesta deseada, la cual es guiada por el par de palabras *feliz* vs *triste*.

Configuración experimental y resultados

En este capítulo se presenta la configuración experimental y el detalle de los conjuntos de datos empleados, además de los parámetros elegidos para la construcción de los diferentes clasificadores para la evaluación del método propuesto, al final se presenta un análisis de los resultados y se comparan las diferentes estrategias de aumento de datos. Además para poder reproducir los resultados, este proyecto esta públicamente disponible en github.com/v1ktop.

5.1. Configuración experimental

La configuración experimental sigue un enfoque supervisado. En la cual se cuenta con un conjunto de historiales de usuario, los cuales pueden verse como un sólo documento, a este documento X le corresponde su etiqueta correspondiente $y \in Y$ en una relación uno a uno. En todos los conjuntos de datos usados se trata únicamente de dos clases, es decir, se trata de una clasificación binaria ($|Y| = 2$).

La metodología empleada está compuesta de 4 fases: preprocesamiento, aumento de datos, entrenamiento y evaluación. En el preprocesamiento se realizan las modificaciones necesarias para normalizar los documentos, además de segmentarlos y filtrarlos (tal como se explica en párrafos posteriores); posteriormente se pasa a la etapa de aumento de datos. Una vez con que se han aumentado los datos de entrenamiento se construye un modelo de clasificación, a través de un algoritmo de aprendizaje máquina (en específico es de nuestro interés los métodos de redes profundas). Finalmente, se evalúa el modelo de clasificación sobre un conjunto de datos que no ha sido aumenta-

do ni utilizado en la búsqueda de parámetros durante el entrenamiento, únicamente es preprocesado de la misma forma que los datos de entrenamiento. La figura 5.1 muestra las diferentes fases descritas.

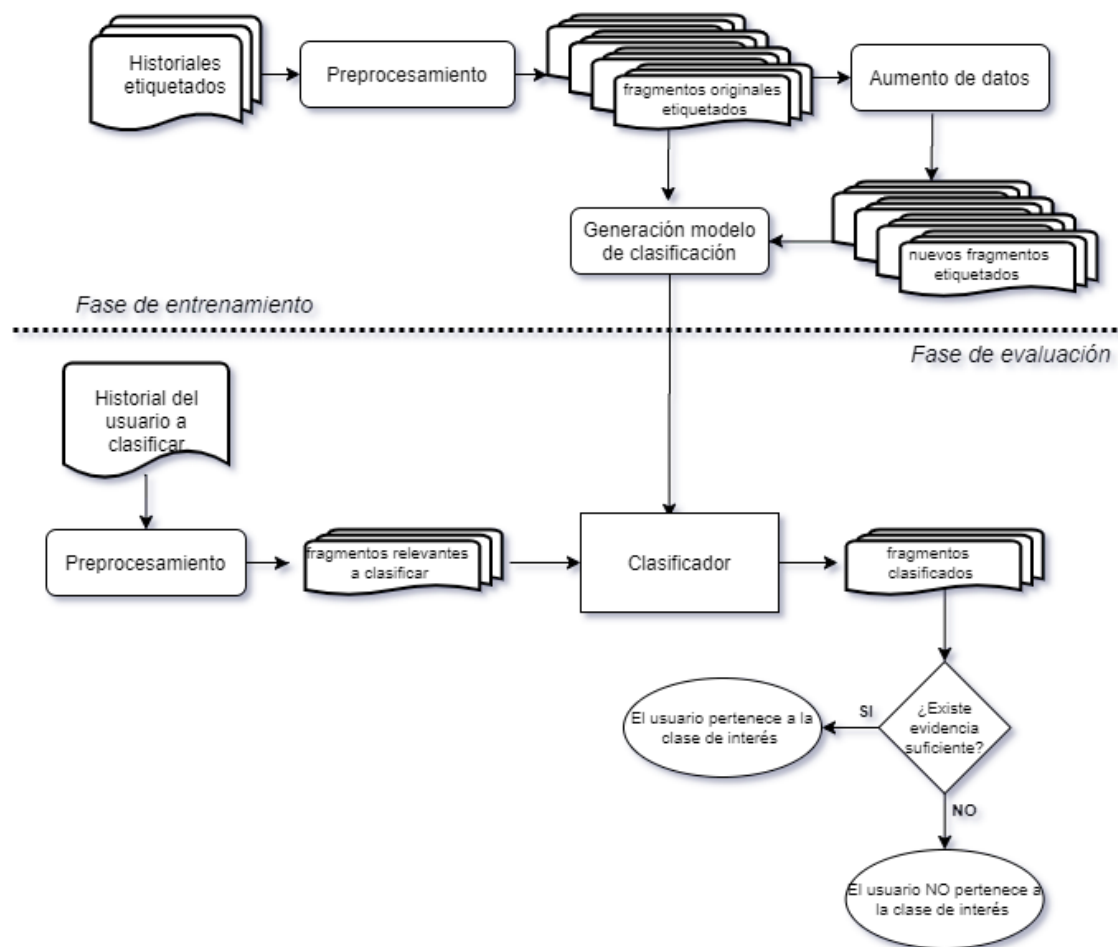


Figura 5.1: Diagrama general de la configuración experimental

5.1.1. Conjunto de datos

Depresión 2018 y Anorexia: Con el propósito de estudiar la detección temprana de depresión y anorexia, los autores (Losada, Crestani, y Parapar, 2018) recopilaron publicaciones de diversos usuarios de la red social Reddit. Para cada usuario la colección contiene una secuencia de publicaciones en orden cronológico. Este conjunto de datos se caracteriza por tener una gran cantidad de texto pero con muy pocos usua-

rios, como se puede observar en la figura 5.2. Hay dos categorías para cada usuario en cada tarea. El número de usuarios total en cada conjunto se presenta en la tabla 5.1. Dado que se trabaja con conjuntos de datos muy desbalanceados el aumento de datos solo se aplica sobre la clase de interés o clase positiva.

Depresión 2019: Presentado en la tareas eRisk 2019 Losada, Crestani, y Parapar (2019a), a diferencia de la edición 2018 en esta ocasión el objetivo es predecir los niveles de depresión de un usuario (mínima, media, moderada, severa). Con el objetivo de que los resultados sean comparables en este trabajo se redujo el problema a una clasificación binaria como se trato con el conjunto del 2018; para esto los usuarios con depresión media a severa se tomaron como ejemplos de la clase positiva.

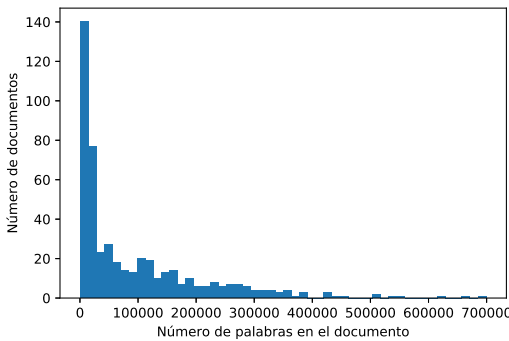
Para el entrenamiento solo se consideraron 16 usuarios para la clase positiva como se muestra en la tabla 5.1, para obtener usuarios de la clase negativa se tomaron los etiquetados como negativos en el conjunto de entrenamiento del eRisk 2018. Finalmente el conjunto de test, solo se dividió en dos clases quedando 60 positivos y 10 negativos (deprimidos y no deprimidos respectivamente).

Tabla 5.1: Número de usuarios en los conjuntos de datos y número de secuencias con 64 palabras después del pre-procesamiento, sin aplicar el filtro. Los números resaltados en negritas representan el numero de historiales comparado con el numero de secuencias.

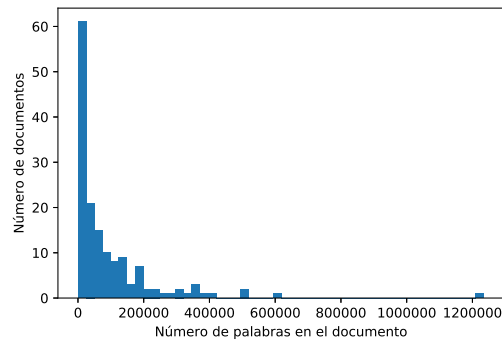
Usuarios	Entrenamiento	Evaluación	Vocabulario
<i>Conjunto 1: Depresión 2018</i>			
deprimido	135 - 31,396	79 - 25,967	
no-deprimido	752 - 227,189	741 - 272,703	
Total	<i>887 - 138,232</i>	<i>820 - 298,670</i>	<i>202,151</i>
<i>Conjunto 2: Depresión 2019</i>			
deprimido	16 - 5,731	60 - 18,534	
no-deprimido	752 - 227,189	10 - 5,011	
Total	<i>768 - 232,920</i>	<i>70 - 23,545</i>	<i>195,047</i>
<i>Conjunto 3: Anorexia</i>			
con anorexia	61 - 23,335	73 - 16,751	
sin-anorexia	411 - 36,484	742 - 254,640	
Total	<i>472 - 130,574</i>	<i>815 - 271,391</i>	<i>67,724</i>

Tabla 5.2: Número de usuarios en los conjuntos de datos y número de secuencias con 64 palabras después del pre-procesamiento, realizando el filtro. Los números resaltados en negritas representan el numero de historiales comparado con el numero de secuencias

Usuarios	Entrenamiento	Evaluación	Vocabulario
<i>Conjunto 1: Depresión 2018</i>			
deprimido	135 - 24,483	79 - 25,967	
no-deprimido	744 - 98,783	741 - 272,703	
Total	879 - 123,266	820 - 298,670	104,800
<i>Conjunto 2: Depresión 2019</i>			
deprimido	16 - 5,731	60 - 18,534	
no-deprimido	746 - 125,823	10 - 5,011	
Total	762 - 131,554	70 - 23,545	118,668
<i>Conjunto 2: Anorexia</i>			
con anorexia	61 -23,335	73 -16,751	
sin-anorexia	411 -107,239	742 -254,640	
Total	472 -130,574	815 -271,391	131,264



(a) Depresión 2018



(b) Anorexia

Figura 5.2: Distribución del numero de palabras en los historiales de usuarios estudiados

5.1.2. Pre-procesamiento

Dado que los documentos extraídos de redes sociales no siguen un lenguaje formal y además de texto existen direcciones de páginas web que los usuarios comparten, emoticonos y caracteres especiales, entre otros; es necesario que antes del aumento de datos exista un preprocesamiento de los textos como una forma de reducir el ruido de los documentos originales.

Los pasos del procesamiento seguido son los siguientes:

- 1.- Normalización: Se identifican las páginas web en el texto y se reemplazan mediante la etiqueta `http_`.
- 2.- Tokenización: Utilizando la herramienta NLTK se remueve de cada texto signos de puntuación y caracteres especiales.
- 3.- Segmentación: Los documentos originales son segmentados en pequeños fragmentos. Es decir, cada historial de usuario se fragmenta en secuencias de 64 palabras (véase la siguiente sección).
- 4.- Filtrado: Solo se conservan segmentos identificados como importantes para la clasificación (véase la siguiente sección).

Segmentación y filtrado

Con el proposito de que el aumento de datos pueda ser proporcional independientemente de la longitud del documento original. Cada documento se dividió en segmentos de 64 palabras¹. Posteriormente se filtró el conjunto de entrenamiento para conservar solo los segmentos importantes para realizar la clasificación. Es decir, se identificaron aquellos fragmentos con la mayor cantidad de palabras discriminantes. Para ello se identificaron las palabras más discriminantes dentro del vocabulario del conjunto de entrenamiento mediante la técnica de selección de características χ^2 . Posteriormente se conservaron aquellos fragmentos que contengan un determinado número de palabras con alta puntuación χ^2 .

Específicamente solo se seleccionaron términos estadísticamente significativos al nivel 0.001, equivalente a una puntuación $\chi^2 > 10.83$ con un grado de libertad. En la tabla 5.2 se muestran los números de usuarios y secuencias obtenidas después de aplicar este filtro; para el conjunto de depresión el criterio de selección fue que la secuencia contuviera al menos 20 palabras de 1071 palabras con alta puntuación, y para el conjunto de anorexia 15 palabras de 1032. Como puede observarse en ambos casos se trata de umbrales altos. Esto se debe principalmente a que en las palabras con alta puntuación están presentes palabras vacías, palabras que tradicionalmente se eliminan para tareas de clasificación temática. No obstante, en nuestro caso, se trata de una tarea donde el estilo es importante (p.e. uso de pronombres personales).

¹Este parámetro se determinó de manera empírica.

5.1.3. Configuración de los métodos propuestos

Para comprobar la efectividad del método propuesto se experimenta con 7 configuraciones diferentes: 2 líneas base, 2 métodos del estado del arte y 3 métodos propuestos. Además de esto se introduce un parámetro n para observar el grado pertinente del aumento de datos, el cual indica el número de documentos nuevos aumentados, tomando valores enteros en el rango $[1, 10]$.

Sin aumento de datos

Este método es la primera línea base y solo consideran los datos originales filtrados para el entrenamiento de los modelos (véase la tabla 5.2).

Sobre muestreo

Esta línea base, consiste en incrementar el número de ejemplos de la clase minoritaria; este método no implica alguna pérdida de información ya que ningún elemento es modificado o descartado. Sin embargo la única desventaja es, que el modelo de aprendizaje generado tiende a sobre ajustarse, debido a que no agrega variabilidad en los datos.

Tesaurus

Este método del estado del arte fue propuesto por (Zhang, Zhao, y LeCun, 2015) y demostró mejoras de un 1 a 2 % en exactitud para la clasificación de opiniones. También fue implementado por (Wei and Zou, 2019) con algunas modificaciones obteniendo una mejora entre un 1 y 2 % en comparación de no hacer aumento de datos, otros trabajos que utilizan este método como referencia y han encontrado evidencia de que agrega una ganancia en los resultados de clasificación son: (Jungiewicz and Smywiński-Pohl, 2019), (Kumar et al., 2019), (Park and Ahn, 2019).

Para decidir cuantas palabras reemplazar dada una secuencia de palabras, se utiliza el parámetro $p = 0.5$ para calcular el valor de r palabras a reemplazar, la selección de dichas palabras es aleatoria, el recurso externo para encontrar sinónimos es un tesaurus (en este caso Wordnet²), y finalmente en la fase de reemplazo, de las palabras candidatas, se selecciona el índice dado un número aleatorio s generado de una

²www.wordnet.princeton.edu/

distribución geométrica con parámetro $q = 0.5$.

El propósito de este método es ser muy conservativo en la modificación del texto original y el número s controla la diversidad del vocabulario que por lo genera. Para decidir que palabras reemplazareemplada).

Sustitución sin restricción y reemplazo mediante similitud coseno

Diversos estudios sugieren utilizar vectores de modelos pre-entrenados como Word2Vec, Glove, entre otros; la idea es recuperar palabras que se utilizan en contextos similares, en lugar de sinónimos.

Para decidir que palabras reemplazar se omiten palabras de paro y aquellas que no sean etiquetadas como sustantivos, adjetivos, verbos y adverbios; con el propósito de agregar mas variabilidad en los ejemplos el número r es calculado con el parámetro $p = 0.2$. En la fase de reemplazo las palabras más similares se seleccionan mediante similitud coseno, utilizándolas de mayor a menor en una selección sin reemplazo.

El modelo de vectores pre-entrenados para representar las palabras de una secuencia fue Glove³ con 300 dimensiones (Pennington, Socher, y Manning, 2014). Este modelo fue pre-entrenado con la base de datos Common Crawl, con 42 millones de tokens y 1.9 millones de palabras.

Con este método se espera obtener mayor diversidad en el vocabulario en comparación a utilizar un tesoro y obtener palabras muy similares que se emplean en el mismo contexto.

Sustitución con restricción χ^2 y reemplazo mediante similitud relacional

A diferencia del método anterior, una vez calculado el número r de palabras a reemplazar, se omiten las palabras con mayor puntuación χ^2 con un nivel de significación estadística de 0.001. Con este método se espera conservar una combinación de estilo y contenido además de agregar variabilidad en los datos.

Reemplazo mediante similitud relacional equivalente

En la fase de selección se fija el valor del parámetro $p = 0.2$ y en la fase de reemplazo se utiliza la similitud relacional positiva; esto es obtener un vocabulario muy similar a la etiqueta de la clase pero no el mismo. Las relaciones buscadas se

³<https://nlp.stanford.edu/projects/glove/>

enlistan en la tabla 5.3 para cada tarea de clasificación. Por ejemplo, para buscar las palabras candidatas a la palabra “boyfriend”, se utiliza la relación “*depressed*” es a “*boyfriend*” como “*anxious*” es a ?.

Reemplazo mediante similitud relacional contraria

Este último método es similar al método anterior, lo único que cambia es la clase objetivo, en este caso se toman los documentos de clase opuesta (la clase negativa). Por ejemplo para buscar las palabras candidatas a la palabra “boyfriend”, se utiliza la relación “*happiness*” es a “*boyfriend*” como “*anxious*” es a ?. La tabla 5.3 resume las etiquetas empleadas para realizar el aumento.

Tabla 5.3: Etiquetas utilizadas en el proceso de aumento para los métodos de similitud relacional.

Conjunto	Clase	Etiqueta	Palabra relacionada
Depresión	1	depressed	anxious
	0	happiness	frustrated
			unhappy
			despondent
			discouraged
Anorexia	1	anorexic	bulimic
	0	healthy	underweight
			obese
			malnourished
			unhealthy

Ejemplos del aumento de datos

En la tabla 5.4 se presentan diversos ejemplos de aumento, el método basado en tesaurus agrega un vocabulario mas formal, en comparación con los basados en similitudes relacionales. El método basado en restricción χ^2 conserva palabras importantes como “feel”, mientras que los otros no toman en consideración esto. Por otra parte el método basado en relaciones equivalentes agrega la palabra “*unfortunate*” como una palabra relacionada a la palabra “*unhappy*”.

La tabla 5.5 presenta ejemplos del aumento basado en relaciones contrarias, las palabras relacionadas a un contexto feliz, son llevadas a un contexto contrario. Por ejemplo el verbo “*talked*” es reemplazado por “*complained*” y “*bothered*”.

Tabla 5.4: Ejemplos del aumento de datos, las palabras resaltadas en negritas son las que resultaron afectadas después de la transformación.

Método	Secuencia
Sin Aumento	a lot of the time i have trouble communicating why i feel so unhappy
Thesauro	a lot of the time i hold trouble communicating why i feel thusly infelicitous
Sin Restricción	a lots of the time i have trouble communicating why i feeling so unhappy
Restricción χ^2	a lot of the time i have difficulty informing why i feel so unhappy
Equivalencia	a much of the place i have troubles informing why i feeling so unfortunate

Tabla 5.5: Ejemplos del aumento de datos para el método basado en relaciones contrarias, las palabras resaltadas en negritas son las que resultaron afectadas después de la transformación.

Método	Secuencia
Sin Aumento	i connected with a girl we sat up and talked all night
(happiness, anxious)	i disconnected with a girl we sat up and talked all night
(happiness, frustrated)	i connected with a girl we sat up and complained all night
(happiness, unhappy)	i connected with a boy we complained up and talked all night
(happiness, despondent)	i dispirited with a girl we sat up and talked all night
(happiness, discouraged)	i connected with a shy we dismayed up and bothered all night

5.1.4. Configuración de los modelos de aprendizaje

Para evaluar el efecto del aumento de datos se utilizaron dos arquitecturas de aprendizaje profundo. Ambas son arquitecturas con resultados relevantes en tareas de clasificación de textos: una red LSTM bidireccional y una red convolucional CNN. Cada arquitectura tiene diferencias, por ejemplo, al considerar el aspecto secuencial inherente de un texto, en el caso de la red recurrente; o cuando se consideran subsecuencias como elementos aislados en el caso de la red convolucional.

A pesar de que el enfoque principal de este trabajo está enfocado al efecto del aumento de datos en redes neuronales profundas, también se realizaron experimentos

en modelos tradicionalmente usados en la clasificación de textos. El objetivo es tener valores de referencia respecto a los métodos propuestos.

Como métodos de clasificación tradicional, se usaron las Máquinas de Soporte Vectorial, considerando el desbalanceo o no al modificar el parámetro de regularización c . Nos referiremos al modelo que no considera el desbalanceo como SVM y cuando se considera lo indicamos como SVM-C.

Modelos lineales

El primer modelo es construido mediante una Máquina de Soporte Vectorial (SVM) con kernel lineal, la entrada es el historial completo de un usuario representado como un vector de características mediante el pesado *tf-idf* y normalizado mediante la norma l_2 , las palabras de paro se mantienen y se utiliza todo el vocabulario extraído como características.

El segundo algoritmo utilizado, SVM-C, es basado en el primer modelo, con la diferencia de que en este caso se modifica el parámetro de regularización C y automáticamente se ajustan los pesos inversamente proporcional a la frecuencia de las clases en los datos de entrada de acuerdo la ecuación 5.1.1

$$C = N/2c_n \quad (5.1.1)$$

En donde N es el número total de ejemplos y c_n el número de ejemplos en la clase c .

Modelos basados en redes neuronales

Con el objetivo principal de establecer las bases sobre en que tipo de arquitecturas es más recomendable realizar aumento de datos. Se implementan dos arquitecturas diferentes: una red Bidireccional LSTM (Bi-LSTM) y una red convolucional (CNN); teniendo en común la capa de entrada y capa de salida.

La **capa de entrada** recibe una secuencia de 64 palabras, cada palabra es representada por un vector de 300 dimensiones obtenido del modelo pre-entrenado FastText⁴, si alguna palabra no está en el vocabulario, su vector es obtenido de la representación de sus n -gramas de caracteres. En el entrenamiento esta capa es estática para reducir el número de parámetros entrenables.

⁴<https://fasttext.cc/docs/en/crawl-vectors.html>

La **capa de salida** es una neurona que recibe como entrada la última capa oculta del modelo, la representación aprendida de los parámetros internos. Mediante la función sigmoide, ecuación 5.1.2, se calcula la probabilidad de que la secuencia de palabras pertenezca a la clase 0 o a la clase 1.

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (5.1.2)$$

Para inicializar los pesos de la capa final correctamente, el *bias* (sesgo) inicial se deriva de la ecuación 5.1.3. Con la inicialización correcta la función de pérdida inicial se debe aproximar a $\ln(2) = 0.69314$.

$$\begin{aligned} p_0 &= \frac{\text{pos}}{\text{pos} + \text{neg}} = \frac{1}{1 + e^{-b_0}} \\ b_0 &= -\log_e\left(\frac{1}{p_0 - 1}\right) \\ b_0 &= \log_e(\text{pos}/\text{neg}) \end{aligned} \quad (5.1.3)$$

Configurando el sesgo inicial correctamente ayuda a la convergencia del modelo desde la primer época.

Derivado de la arquitectura presentada en (Adhikari et al., 2019), en la figura 5.3, se presenta la arquitectura empleada para el modelo Bi-LSTM, la red bidireccional se compone de dos redes LSTM con 256 neuronas cada una, posteriormente se aplica una capa de *Dropout* con una tasa de 0.2, una capa totalmente conectada con 256 unidades, una capa de *Dropout* con una tasa de 0.2 y en la última capa una sola neurona activada mediante la función sigmoide 5.1.2. Los nodos intermedios de las capas ocultas se activan con la función de activación Relu 2.4.4.

En la figura 5.4, se presenta la arquitectura empleada para la red convolucional (CNN), esta arquitectura es basada en el trabajo de (Kim, 2014). Se implementan tres tamaños de filtro [3,4,5], cada uno con 300 filtros. Los filtros realizan convoluciones en una matriz que representa a la secuencia de palabras y generan mapas de características de longitud variable; la operación de *Max Pooling* se realiza sobre cada mapa, es decir, se calcula el número mayor de cada mapa de características. A partir de esto se obtienen diferentes vectores de características de diferentes tamaños y la penúltima capa se forma concatenándolos para formar un vector final de características, la capa final recibe este vector de características para clasificar la secuencia de palabras. Los

nodos intermedios de las capas ocultas se activan con la función de activación Relu 2.4.4.

Entrenamiento

Para encontrar los hiperparámetros de los modelos se realizó una división del conjunto de entrenamiento en 3 particiones diferentes (3 K-Folds) con una proporción de 66 % para entrenar y 33 % para evaluar.

En el caso de los modelos de redes neuronales se entrenan de forma que sean

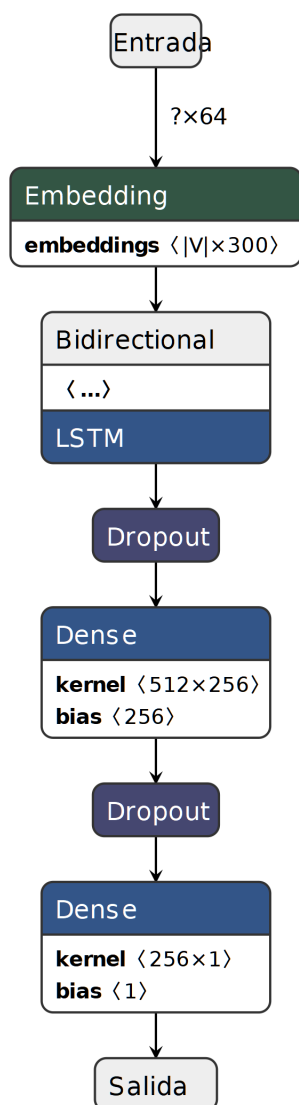


Figura 5.3: Arquitectura del modelo Bi-LSTM

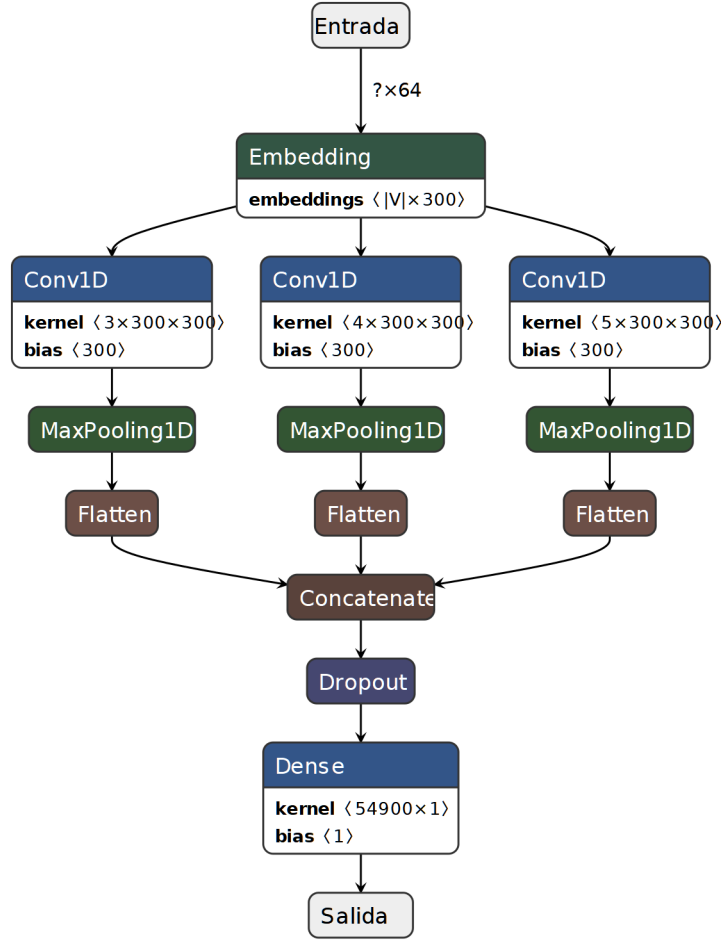


Figura 5.4: Arquitectura del modelo CNN con múltiples tamaños de convolución

sensibles al desbalance (Wang et al., 2016), utilizando un peso adicional para cada clase, calculado mediante la fórmula 5.1.1. Con esto el error es incrementado para ejemplos en la clase de interés y decrementado para la clase menos importante.

Los parámetros elegidos para el entrenamiento se resumen en la tabla 5.6.

Evaluación

Como resultado del entrenamiento se tiene un clasificador. Este clasificador es evaluado a través de un conjunto de datos previamente seleccionado, el cual no ha sido utilizado en la fase de entrenamiento. Cabe recordar que dicho clasificador se ha entrenado para determinar la clase de un fragmento del historial de un usuario. De

Tabla 5.6: Parámetros utilizados para el entrenamiento de los modelos basados en redes neuronales.

Parámetro	Valor
Tasa de aprendizaje	1.00E-03
Tamaño del Batch	1024
Función de pérdida	Entropía cruzada binaria
Maximo número de épocas	20
Criterio de paro	CNN=6; Bi-LSTM=3
Pruebas independientes	3

esta forma la predicción final se realiza observando la clase de todos los fragmentos del usuario en evaluación. Si el número de fragmentos pertenecientes a la clase de interés supera cierto umbral, se considera que se tiene suficiente evidencia para determinar que el usuario pertenece a la clase de interés (véase la figura 5.1).

Implementación

Para el preprocesamiento y el etiquetado de las secuencias de texto se utilizó la librería NLTK (Loper and Bird, 2002), para la normalización y el cálculo de medidas de similitud de los embeddings la librería gensim⁵. Los modelos lineales fueron implementados utilizando la librería scikit-learn⁶ (Pedregosa et al., 2011), los modelos neuronales⁷ (Abadi et al., 2015). Todas en su última versión mediante en el lenguaje de programación Python. Finalmente el 50 % de los modelos fueron entrenados con una computadora personal y el 50 % en Colab⁸ (Una herramienta de acceso gratuito para entrenar redes neuronales en la nube).

5.2. Resultados

En la tabla 5.7 se presentan los resultados de los experimentos mediante el promedio de la métrica $F1$ calculada en base a la clase de interés (la clase positiva). Debido a la aleatoriedad de las redes neuronales los resultados obtenidos en estos modelos se

⁵www.radmehurek.com/gensim

⁶www.scikit-learn.org/stable/

⁷www.tensorflow.org

⁸colab.research.google.com

presentan como un promedio de 3 ejecuciones independientes y la desviación estándar obtenida; la columna nombrada como n indica el valor de aumento correspondiente en el conjunto de datos, así con $n = 1$ indica que el conjunto de entrenamiento original se duplico. En la evaluación se utilizó un umbral igual a 0.5 para los modelos Bi-LSTM, SVM y SVM-C; 0.4 para el modelo CNN.

En esta tabla se comparan los métodos propuestos; reemplazo mediante relaciones equivalentes, relaciones contrarias y restricción mediante la selección de características χ^2 , contra la línea base (sin aumento de datos) y los métodos de referencia: (i) sobre muestreo, (ii) utilizando un tesoro, y (iii) selección sin restricción.

Los mejores valores encontrados para cada conjunto de datos están resaltados en negritas. Así para el conjunto de *Depresión 2018* el mejor valor encontrado fue de $53\% \pm 1$ en $F1$ utilizando el método Sin restricción, para el conjunto de *Depresión 2019* $88\% \pm 2$ mediante los métodos: Tesoro, Restricción χ^2 y Relación Positiva; finalmente para el conjunto de *Anorexia* $82\% \pm 1$ mediante el método Restricción χ^2 . En general los resultados para el conjunto de Depresión 2019 y Anorexia, son mejores en comparación a los obtenidos en el conjunto Depresión 2018, esto se lo podemos atribuir a la forma en que se etiquetaron los datos ya que para el conjunto de Depresión 2018 el etiquetado se realizó de forma automática y se capturo un gran número de falsos positivos.

En primer lugar se compara la línea base (no realizar aumento de datos) contra los diferentes métodos de aumento de datos, en donde se puede observar que la mayoría de los métodos superan esta línea base, a excepción en el modelo basado en SVM-C para el conjunto de depresión 2019 en donde no se consigue mejorar el modelo base.

Con respecto a los algoritmos el aumento de datos el método que sobre sale es el basado en Restricción χ^2 ya que obtiene los mejores resultados para el conjunto de Depresión 2019 mediante la red Bi-LSTM y también para el conjunto de Anorexia utilizando la red CNN. Por otra parte en los algoritmos lineales se observa un gran incremento en el modelo SVM obteniendo mejores resultados, a excepción del conjunto Depresion 2019, en comparación con el algoritmo SVM-C que considera el desbalance de las clases.

Tabla 5.7: Resultados en términos de la métrica F1, la variable n indica la magnitud del aumento en el conjunto original.

Conjunto de datos		Bi-LSTM		CNN		SVM		SVM-C	
	Metodo	F1	n	F1	n	F1	n	F1	n
Depresión 2018	Sin aumento	47±3	-	45±5	-	16	-	50	-
	Over	50±2	3	51±3	6	51	4	51	9
	Tesauro	50±4	5	50±1	5	50	10	48	1
	Sin restriccion	48±6	5	53±1	2	53	10	50	3
	Restricción Chi2	52±3	4	50±3	2	53	10	50	1
	Relacion Positiva	51±5	8	50±1	9	41	8	49	1
	Relacion Contraria	46±3	9	49±1	10	52	8	51	1
Depresión 2019	Sin aumento	74±8	-	81±3	-	0	-	50	-
	Over	81±13	1	85±3	2	13	9	50	1
	Tesauro	80±5	7	88±2	7	10	10	49	1
	Sin restriccion	74±10	1	85±2	2	10	7	47	1
	Restricción Chi2	88±4	1	83±4	1	10	8	47	1
	Relacion Positiva	78±15	1	88±2	6	6	8	46	1
	Relacion Contraria	65±7	9	82±3	9	31	4	15	5
Anorexia	Sin aumento	75±3	-	80±0	-	67	-	72	-
	Over	80±2	5	81±2	1	77	7	75	6
	Tesauro	80±1	8	82±3	2	76	2	75	3
	Sin restriccion	78±2	4	81±1	6	78	6	78	6
	Restricción Chi2	79±2	5	82±1	7	78	5	77	1
	Relacion Positiva	79±1	5	82±1	1	78	6	76	1
	Relacion Contraria	76±4	4	81±2	1	81	8	75	5

5.3. Análisis y discusión de los resultados

Con el objetivo de observar la relación entre el valor F1 alcanzado y la magnitud n en el conjunto de datos, se presentan la figuras 5.5, 5.6 y 5.7. En las gráficas se comparan los métodos: sin aumento de datos, Tesauro, restricción χ^2 , relación equivalente y relación contraria. Las gráficas se presentan en una escala de 0 a 100% para el valor $F1$, el eje x refleja el número de secuencias aumentadas por cada secuencia original en el conjunto de entrenamiento y el eje y la ganancia o pérdida en $F1$ en comparación con la línea base que es no realizar aumento de datos.

En la figura 5.5 (a), el aumento para la red bidireccional, el mejor resultado con una ganancia de 4.51 puntos se obtiene con $n = 4$ con el método *Restricción χ^2* . Se puede observar que una vez alcanzado el balance la ganancia comienza a decrecer,

los métodos restantes no obtienen la misma ganancia, por lo que en esta tarea es importante conservar las características discriminantes cuando se realiza el aumento de datos, sin embargo, si el conjunto se n-plica muchas veces el modelo se sobre ajusta a los datos que se conservan.

En la figura 5.5 (b), aumento de datos para la red CNN, el mejor valor encontrado con una ganancia de 5.45 puntos fue con $n = 6$ con el método *Restricción χ^2* . A diferencia de la red recurrente las ganancias en F1 se encuentran en un rango entre 0 y 5 puntos, los métodos con menor ganancia fueron; el basado en equivalencias y el relaciones contrarias.

En la figura 5.5 (c), aumento de datos para SVM, se presenta una tendencia creciente en relación al parámetro n , el mejor valor obtenido es una ganancia de 38 puntos mediante el método *Restricción χ^2* . En este caso la ganancia se debe más a la afectación de los pesos *tf-idf* que al aumento de datos, aún así el método basado en restricción χ^2 ofrece una mejora desde el primer documento en comparación con los otros métodos. La figura 5.5 (d) representa los resultados en el algoritmo SVM-C, esta figura muestra que el aumento de datos en este caso no es necesario para este tipo de modelos o no se aprovecha como lo haría una red neuronal.

Las gráficas que representan la comparación de los algoritmos propuestos para el conjunto de datos Depresión 2019, se presenta en la figura 5.6. En la subfigura (a) la única ganancia significativa se da en el modelo Bi-LSTM con 13.82 puntos en *F1*, sin embargo después de triplicar conjunto de entrenamiento para la clase positiva ocurre lo contrario. Estas variaciones muy notables se deben a que el conjunto de test es menor y por lo tanto los falsos positivos afectan en gran medida en la evaluación. En la subfigura (b) se presenta la evaluación en el modelo CNN, en este caso el método del estado del arte Tesauro obtiene mejores resultados en comparación con los métodos propuestos. En la subfigura (c) aumentar el conjunto mediante relaciones contrarias obtiene mejores resultados, aunque se le puede atribuir al peso que se le asigna a las palabras, dado que el método de relaciones contrarias aumenta documentos de clase positiva para aumentar la clase positiva. Su contra parte se representa en la subfigura (d) y dado que el algoritmo SVM-C considera el desbalance de los ejemplos no se consiguen mejoras. Como nota final en este conjunto no se consigue el balance de clases debido a las proporciones del conjunto.

La figura 5.7 presenta el aumento de datos para el conjunto de anorexia. Al igual que en las gráficas anteriores se presenta el efecto del aumento de datos en los diferen-

tes algoritmos de clasificación empleados. Similar que en los conjuntos de depresión el método de Restricción χ^2 obtiene mejores ganancias en los diferentes aumentos. Para la red convolucional la ganancia es menor, pero se observa que es el método más consistente ya que los diferentes aumentos no afectan en sentido contrario a la clasificación como sucede con el método de tesauro y equivalencia contraria. Para el modelo lineal SVM el método de equivalencias contrarias obtiene mejores resultados, conforme crece el aumento de datos pero en el modelo SVM-C obtiene el peor rendimiento. Para el modelo SVM-C el método de Restricción χ^2 vuelve a sobre salir con hasta 4.98 puntos de ganancia a la línea base, acumulando más evidencia sobre nuestra hipótesis inicial.

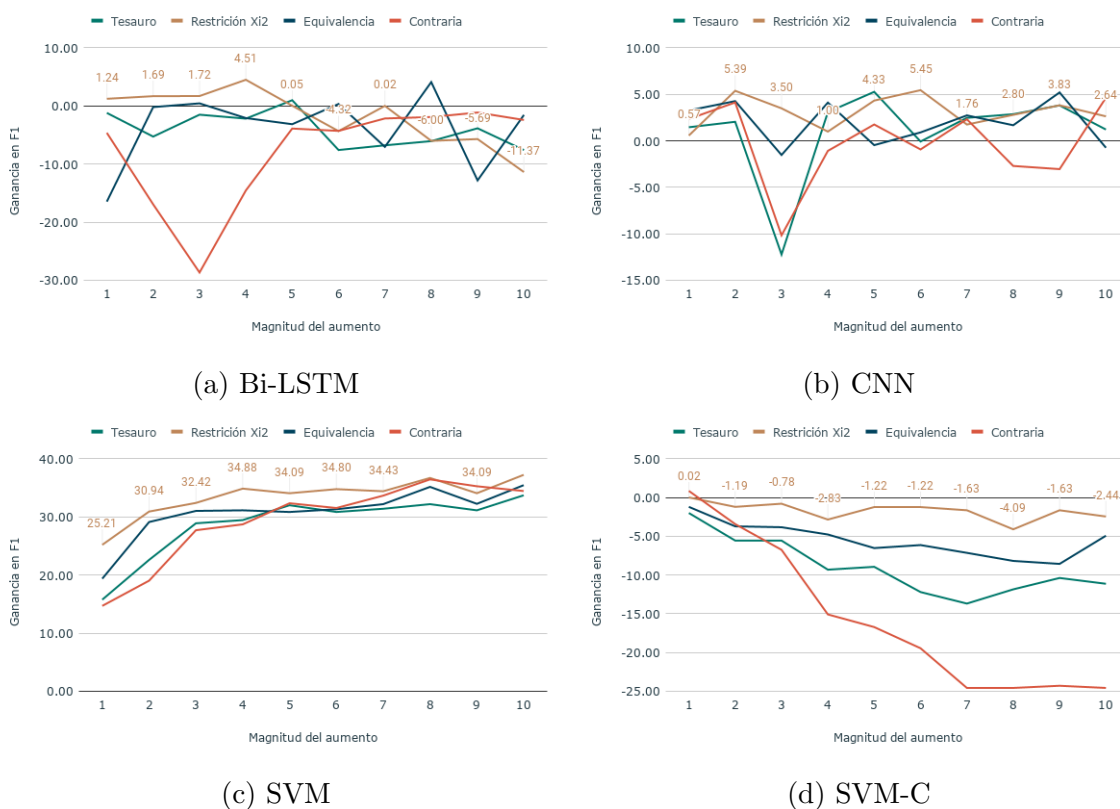
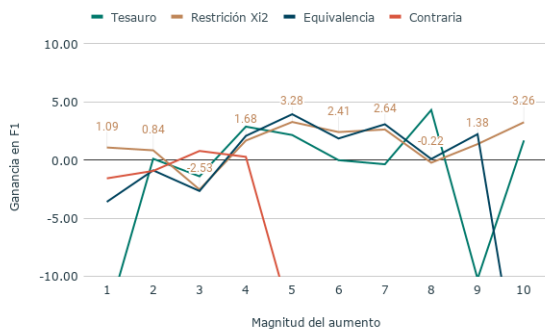


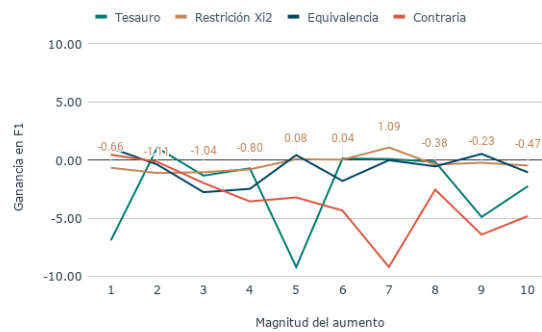
Figura 5.5: Relación entre el aumento del conjunto de datos *Depresión 2018* y la ganancia en F1.



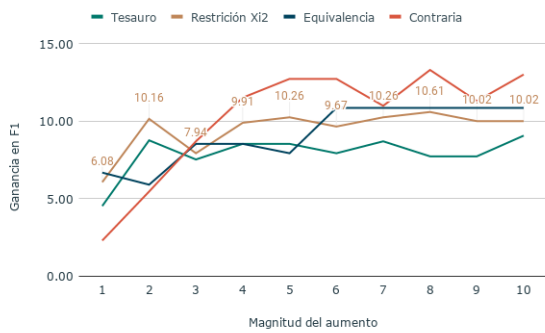
Figura 5.6: Relación entre el aumento del conjunto de datos *Depresión 2019* y la ganancia en F1.



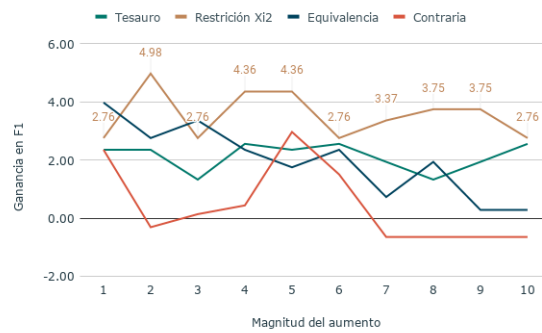
(a) Bi-LSTM



(b) CNN



(c) SVM



(d) SVM-C

Figura 5.7: Relación entre el aumento del conjunto de datos y la ganancia en F1 para el conjunto *Anorexia*.

5.3.1. Comparación con el estado del arte en detección de depresión y anorexia

En la figura 5.8 se comparan los resultados obtenidos mediante aumento de datos utilizando una red Bi-LSTM y el aumento de datos mediante el método Restricción χ^2 , con los modelos evaluados en la conferencia eRISK 2018 (Losada, Crestani, y Parapar, 2018). Para detección de depresión de un total de 45 modelos nuestra propuesta se puede ubicar en el sexto lugar por arriba del tercer cuartil. Para la detección de anorexia, de un total de 35 propuestas nuestro modelo quedaría en el segundo lugar y muy por encima del tercer cuartil. Es importante señalar que para la detección de depresión el mejor modelo presentado en la tarea eRisk2018 se obtuvo mediante la ingeniería de características y para la detección de anorexia se utilizó una red convolucional con vectores distribucionales entrenados en un corpus perteneciente al dominio, por lo que dichas propuestas podrían mejorar mediante el aumento de datos propuesto. Como nota final los resultados para el conjunto de Depresión 2019 no se comparan con los obtenidos en el evento eRisk2019 por que se realizó una evaluación diferente.

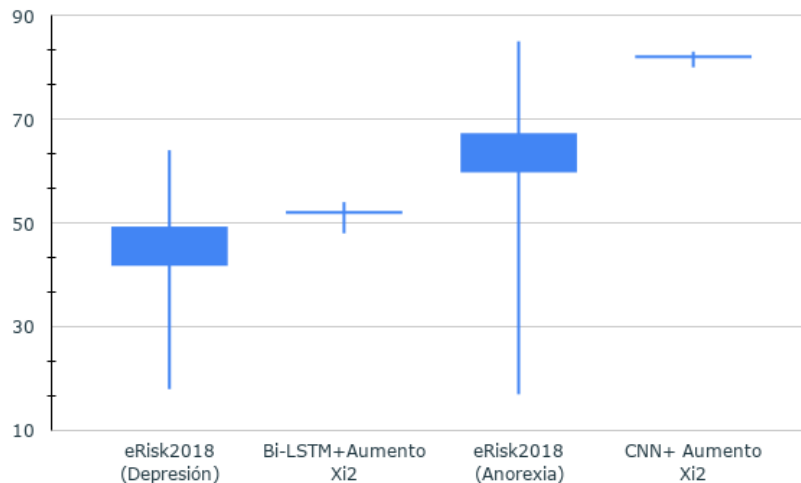


Figura 5.8: Comparación con el estado del arte en detección de depresión y anorexia

5.3.2. Análisis del aumento de datos

Con el objetivo de comprobar como afecta el aumento de datos a la originalidad y diversidad del documento original, se recopilaron estadísticas del aumento en el vocabulario además de presentar las palabras más relevantes utilizadas por el método de Restricción χ^2 y para el filtro de secuencias en el pre-procesamiento.

Aumento del vocabulario

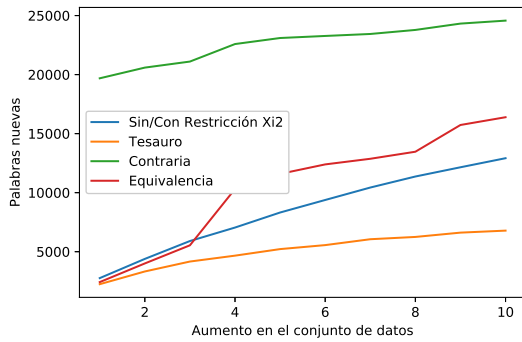
En la figura 5.9 se representa para el eje y el numero de palabras nuevas agregadas en relación con el parámetro n que indica la magnitud del aumento de datos. El objetivo de esta figura es comparar el vocabulario nuevo introducido de acuerdo a cada método de aumento en los diferentes conjuntos de datos.

Como se puede esperar conforme aumenta el número de documentos el vocabulario también lo hace. En las subfiguras (a, c y e) se compara el aumento del vocabulario para la clase positiva, en la cual el método basado en relaciones contrarias incrementa drásticamente el vocabulario desde un documento por cada instancia y el que menos agrega palabras es el basado en tesauro, debido a que el método tesauro utiliza el parámetro p para selección igual a 0.5 y en promedio solo reemplaza 2 palabras por cada segmento a aumentar. Los métodos con y sin restricción agregan el mismo número de palabras debido a que solo difieren en que palabras reemplazar. Por otra parte el basado en relaciones de equivalencia agrega un mayor vocabulario a los dos anterior por lo que se logra el objetivo de insertar un vocabulario diferente al emplear un criterio de similitud diferente.

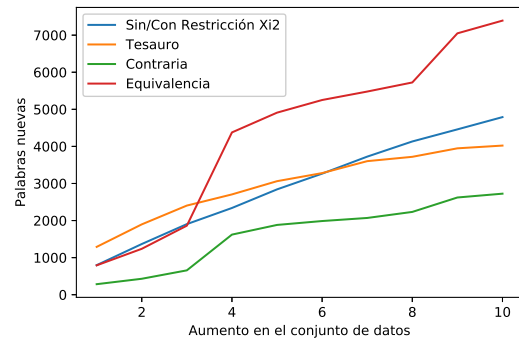
En la subfiguras 5.9 (b, d y f), se compara el aumento del vocabulario considerando ambas clases, debido a que el aumento de datos propuesto se basa sobre la clase de interés (la clase positiva). Resalta el hecho de que aunque el método de equivalencias contrarias introduce un gran vocabulario en la clase positiva para el conjunto de depresión 2018 y anorexia, solo agrega de 500 a 2000 mil palabras nuevas considerando ambas clases, sin embargo para el conjunto de depresión 2019 agrega hasta 7000 palabras nuevas. Este incremento drastico en el vocabulario impacta de forma negativa a los resultados.

En resumen el método que agregó más vocabulario fue el basado en relaciones contrarias, seguido del basado en equivalencias; el método Tesauro es muy conservador en el numero de palabras nuevas agregadas pero se puede observar que las palabras

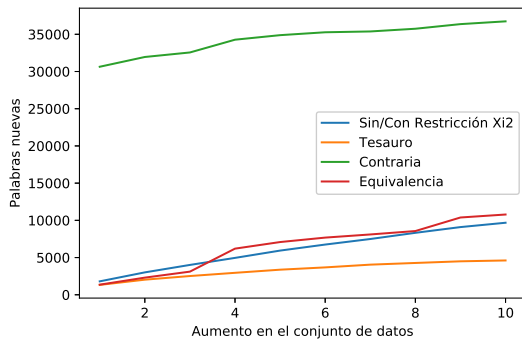
agregadas no aparecen en la clase contraria. Es interesante que el método con restricción agrega la misma modificación que el que no la realiza y puede obtener mejores resultados, por lo que se comprueba la efectividad del método.



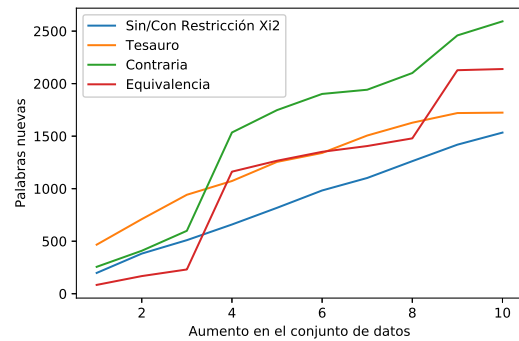
(a) Depresión 2018: Clase positiva



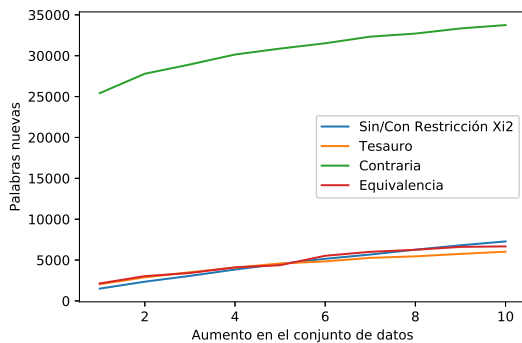
(b) Depresión 2018: Ambas clases



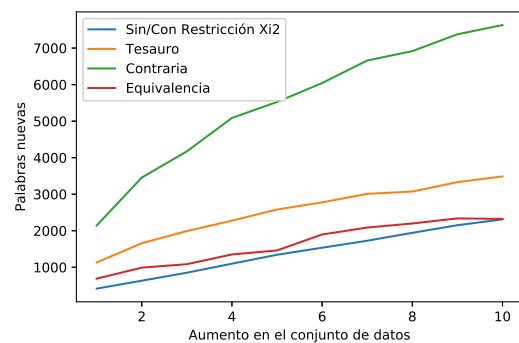
(c) Depresión 2019: Clase positiva



(d) Depresión 2019: Ambas clases



(e) Anorexia: Clase positiva



(f) Anorexia: Ambas clases

Figura 5.9: Relación entre el aumento de datos y el vocabulario nuevo agregado.

Palabras con mayor puntuación χ^2

En la figura 5.10, se representan las palabras con mayor puntuación χ^2 mismas que sirvieron para realizar el pre-procesamiento y también para el método de aumento con restricción. La figura muestra las palabras más importantes en un tamaño de fuente más grande seguidas de las de menor importancia en una fuente más pequeña.

Como se ha demostrado en estudios previos las palabras relacionadas con pronombres personales y posesivos son más utilizadas por personas con signos de depresión o anorexia, además de palabras relacionadas a relaciones personales como: “boyfriend”, “feeling”, “friends”, “dating”. También sobre salen palabras relacionadas a la enfermedad como: “meds”, “medication”, “anorexia”, “depression”; entre otras.



Figura 5.10: Palabras con mayor puntuación χ^2

Conclusiones y Trabajo Futuro

El aumento de datos estructural no es una tarea trivial ya que pertenece al área de generación de lenguaje natural. Se deben considerar muchos factores: la originalidad del texto, la diversidad, la conservación del estilo, la conservación de la etiqueta etc.

Con el objetivo general de “Proponer un método de aumento de datos, considerando estilo y contenido del texto, para mejorar la predicción de los modelos de aprendizaje profundo en las tareas de perfilado de autor”, se presentaron diferentes estrategias de aumento de datos y se evaluaron en dos arquitecturas de aprendizaje profundo.

Gracias a los experimentos realizados y al análisis de los resultados, se llegaron a las siguientes conclusiones.

6.1. Conclusiones

Nuestra primera conclusión es que el aumento de datos ayuda a los métodos basados en redes neuronales, al menos, en las dos arquitecturas evaluadas. Los métodos propuestos mejoran el resultado y dependen fuertemente del conjunto de datos. El método propuesto restringiendo el reemplazo es el que obtiene mejores resultados. Este método conserva las palabras con mayor puntuación χ^2 , sin embargo corre el riesgo de sobre ajuste conforme aumenta el número de documentos nuevos. Si se necesita un gran número de documentos el método basado en relaciones equivalentes es el ideal ya que ofrece un vocabulario más amplio mediante la elección de diferentes semillas (palabras relacionadas a la clase de interés).

Con respecto al efecto del aumento de datos en diferentes arquitecturas de red, el aumento de datos es de mayor beneficio para arquitecturas basadas en redes recurren-

tes, ya que consideran el texto como secuencia con valores dependientes en lugar de considerar características aisladas como es el caso de las redes convolucionales y los modelos lineales. En los modelos lineales se pudo observar un incremento significativo en los resultados de clasificación pero esto es debido a la importancia que se le está dando a la clase positiva y se logró comprobar con un modelo lineal que considera el desbalance; en donde la línea base es muy cercana a los valores que se pueden obtener realizando aumento de datos. Aún considerando este hecho es preferible realizar aumento de datos por los beneficios de regularización que ofrece.

Finalmente se pudo comprobar que es posible mejorar la predicción en el perfilado de depresión y anorexia, logrando una ganancia entre 1 y 5 puntos en términos de la métrica F1 en comparación con no realizar aumento de datos y una ganancia entre 1 y 3 puntos en comparación con otros métodos de aumento. Además fue posible igualar los resultados del estado del arte utilizando modelos neuronales menos complejos.

6.2. Trabajo futuro

Debido a las limitaciones de este proyecto, existen alternativas que no se estudiaron por ejemplo:

- Explorar técnicas supervisadas basadas en parafraseo neuronal, estas técnicas pueden ofrecer una mayor calidad de generación de texto, la principal limitante es el costo computacional.
- Explorar técnicas semi-supervisadas o auto-supervisadas, el aprendizaje auto-supervisado es una rama del aprendizaje computacional que ha demostrado ser una opción para obtener grandes cantidades de datos con etiquetas débiles, sin embargo exige contar con muchos recursos computacionales para poder ser implementado.
- Evaluar el aumento de datos en otras tareas de clasificación similares como: detección de engaño, tendencias suicidas, lenguaje agresivo, entre otras.
- Finalmente se pueden implementar los modelos del estado del arte para la detección de depresión y anorexia, mejorándolos mediante el aumento de datos.

Bibliografía

- Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Goodfellow, I.; Harp, A.; Irving, G.; Isard, M.; Jia, Y.; Jozefowicz, R.; Kaiser, L.; Kudlur, M.; Levenberg, J.; Mané, D.; Monga, R.; Moore, S.; Murray, D.; Olah, C.; Schuster, M.; Shlens, J.; Steiner, B.; Sutskever, I.; Talwar, K.; Tucker, P.; Vanhoucke, V.; Vasudevan, V.; Viégas, F.; Vinyals, O.; Warden, P.; Wattenberg, M.; Wicke, M.; Yu, Y.; y Zheng, X. 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.
- Abed-Esfahani, P.; Howard, D.; Maslej, M.; Patel, S.; Mann, V.; Goegan, S.; y French, L. 2019. Transfer learning for depression: Early detection and severity prediction from social media postings. *CEUR Workshop Proceedings* 2380(September 2019):9–12.
- Adhikari, A.; Ram, A.; Tang, R.; y Lin, J. 2019. Rethinking complex neural network architectures for document classification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4046–4051.
- Almeida, H.; Briand, A.; y Meurs, M.-J. 2017. Detecting early risk of depression from social media user-generated content. In *CLEF (Working Notes)*.
- Androutsopoulos, I., y Malakasiotis, P. 2010. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research* 38:135–187.
- Aragón, M. E.; López-Monroy, A. P.; González-Gurrola, L. C.; y Montes-y Gómez, M. 2019. Detecting Depression in Social Media using Fine-Grained Emotions. (2013):1481–1486.

- Argamon, S.; Koppel, M.; Pennebaker, J. W.; y Schler, J. 2009. Automatically profiling the author of an anonymous text. *Communications of the ACM* 52(2):119.
- Basile, A.; Dwyer, G.; Medvedeva, M.; Rawee, J.; Haagsma, H.; y Nissim, M. 2017. Is there life beyond n-grams? a simple svm-based author profiling system. In *Working Notes of CLEF 2017-Conference and Labs of the Evaluation Forum*.
- Beck, A. T.; Ward, C. H.; Mendelson, M.; Mock, J.; y Erbaugh, J. 1961. An inventory for measuring depression. *Archives of general psychiatry* 4(6):561–571.
- Block, H. D.; Knight Jr, B.; y Rosenblatt, F. 1962. Analysis of a four-layer series-coupled perceptron. ii. *Reviews of Modern Physics* 34(1):135.
- Bogdanova, D.; Rosso, P.; y Solorio, T. 2012. On the impact of sentiment and emotion based features in detecting online sexual predators. In *Proceedings of the 3rd workshop in computational approaches to subjectivity and sentiment analysis*, 110–118. Association for Computational Linguistics.
- Boser, B. E.; Guyon, I. M.; y Vapnik, V. N. 1992. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, 144–152.
- Burdisso, S. G.; Errecalde, M.; y Montes-Y-Gómez, M. 2019. UNSL at Erisk 2019: A Unified Approach for Anorexia, Self-harm and Depression Detection in Social Media. *CEUR Workshop Proceedings* 2380:9–12.
- Cacheda, F.; Iglesias, D. F.; Nóvoa, F. J.; y Carneiro, V. 2018. Analysis and experiments on early detection of depression. volume 2125.
- Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; y Kegelmeyer, W. P. 2002. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research* 16:321–357.
- Conneau, A.; Schwenk, H.; Barrault, L.; y Lecun, Y. 2016. Very deep convolutional networks for text classification. *arXiv preprint arXiv:1606.01781*.
- Cortes, C., y Vapnik, V. 1995. Support-vector networks. *Machine learning* 20(3):273–297.

- Cubuk, E. D.; Zoph, B.; Mane, D.; Vasudevan, V.; y Le, Q. V. 2019. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 113–123.
- Daneshvar, S., y Inkpen, D. 2018. Gender identification in twitter using n-grams and lsa. In *Proceedings of the Ninth International Conference of the CLEF Association (CLEF 2018)*.
- De Choudhury, M.; Gamon, M.; Counts, S.; y Horvitz, E. 2013. Predicting depression via social media. In *Seventh international AAAI conference on weblogs and social media*.
- Dempster, A. P.; Laird, N. M.; y Rubin, D. B. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)* 39(1):1–22.
- Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. *Journal of machine learning research* 3(Mar):1289–1305.
- Glorot, X., y Bengio, Y. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 249–256.
- Glorot, X.; Bordes, A.; y Bengio, Y. 2011. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 315–323.
- Goodfellow, I.; Bengio, Y.; y Courville, A. 2016. *Deep learning*. MIT press.
- Grave, E.; Bojanowski, P.; Gupta, P.; Joulin, A.; y Mikolov, T. 2018. Learning word vectors for 157 languages. *arXiv preprint arXiv:1802.06893*.
- Han, S.; Gao, J.; y Ciravegna, F. 2019. Neural language model based training data augmentation for weakly supervised early rumor detection. In *Proceedings of the 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 105–112.
- Hedderich, M. A., y Klakow, D. 2018. Training a neural network in a low-resource setting on automatically annotated noisy data. *arXiv preprint arXiv:1807.00745*.

- Hochreiter, S., y Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.
- Ikeda, K.; Hattori, G.; Ono, C.; Asoh, H.; y Higashino, T. 2013. Twitter user profiling based on text and community mining for market analysis. *Knowledge-Based Systems* 51:35–47.
- Jimenez-Villar, V.; Sánchez-Junquera, J.; Montes-y Gómez, M.; Villaseñor-Pineda, L.; y Ponzetto, S. P. 2019. Bots and gender profiling using masking techniques. In *Proceedings of the Ninth International Conference of the CLEF Association (CLEF 2019)*.
- Jones, K. S. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*.
- Jungiewicz, M., y Smywiński-Pohl, A. 2019. Towards textual data augmentation for neural networks: synonyms and maximum loss. *Computer Science* 20.
- Kadhim, A. I. 2019. Survey on supervised machine learning techniques. *Artificial Intelligence Review* 52(1):273–292.
- Kamath, U.; Liu, J.; y Whitaker, J. 2019. *Deep learning for nlp and speech recognition*, volume 84. Springer.
- Kim, Y. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.
- Kobayashi, S. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 452–457.
- Kowsari, K.; Jafari Meimandi, K.; Heidarysafa, M.; Mendu, S.; Barnes, L.; y Brown, D. 2019. Text classification algorithms: A survey. *Information* 10(4):150.
- Kumar, R.; Reganti, A. N.; Bhatia, A.; y Maheshwari, T. 2018. Aggression-annotated corpus of hindi-english code-mixed data. *arXiv preprint arXiv:1803.09402*.

- Kumar, A.; Bhattamishra, S.; Bhandari, M.; y Talukdar, P. 2019. Submodular optimization-based diverse paraphrasing and its effectiveness in data augmentation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 3609–3619.
- Laserna, C. M.; Seih, Y. T.; y Pennebaker, J. W. 2014. Um. . Who Like Says You Know: Filler Word Use as a Function of Age, Gender, and Personality. *Journal of Language and Social Psychology* 33(3):328–338.
- LeCun, Y.; Bengio, Y.; y Hinton, G. 2015. Deep learning. *nature* 521(7553):436–444.
- Levy, O., y Goldberg, Y. 2014. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the eighteenth conference on computational natural language learning*, 171–180.
- Liu, N.; Zhou, Z.; Xin, K.; y Ren, F. 2018. Tua1 at erisk 2018. In *CLEF (Working Notes)*.
- Loper, E., y Bird, S. 2002. Nltk: the natural language toolkit. *arXiv preprint cs/0205028*.
- Losada, D. E.; Crestani, F.; y Parapar, J. 2017. erisk 2017: Clef lab on early risk prediction on the internet: experimental foundations. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, 346–360. Springer.
- Losada, D. E.; Crestani, F.; y Parapar, J. 2018. Overview of eRisk 2018: Early Risk Prediction on the Internet (extended lab overview). *CEUR Workshop Proceedings* 2125.
- Losada, D. E.; Crestani, F.; y Parapar, J. 2019a. Overview of eRisk 2019 Early Risk Prediction on the Internet. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 11696 LNCS:340–357.
- Losada, D. E.; Crestani, F.; y Parapar, J. 2019b. Overview of erisk at clef 2019: Early risk prediction on the internet (extended overview). In *CLEF (Working Notes)*.

- Manning, C. D. 2015. Computational Linguistics and Deep Learning. *Computational Linguistics* 41(4).
- Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; y Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.
- Mikolov, T.; Grave, E.; Bojanowski, P.; Puhersch, C.; y Joulin, A. 2017. Advances in pre-training distributed word representations. *arXiv preprint arXiv:1712.09405*.
- Minaee, S.; Kalchbrenner, N.; Cambria, E.; Nikzad, N.; Chenaghlu, M.; y Gao, J. 2020. Deep Learning Based Text Classification: A Comprehensive Review. 1(1):1–42.
- Miyato, T.; Maeda, S.-i.; Koyama, M.; y Ishii, S. 2019. Virtual Adversarial Training : A Regularization Method for Supervised and Semi-Supervised Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41(8):1979–1993.
- Nothman, J.; Qin, H.; y Yurchak, R. 2018. Stop word lists in free open-source software packages. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, 7–12.
- Ortega-Mendoza, R. M.; López-Monroy, A. P.; Franco-Arcega, A.; y Montes-y Gómez, M. 2018a. Emphasizing personal information for Author Profiling: New approaches for term selection and weighting. *Knowledge-Based Systems* 145:169–181.
- Ortega-Mendoza, R. M.; López-Monroy, A. P.; Franco-Arcega, A.; y Montes-y Gómez, M. 2018b. Peimex at erisk2018: Emphasizing personal information for depression and anorexia detection. In *CLEF (Working Notes)*.
- Park, D., y Ahn, C. W. 2019. Self-supervised contextual data augmentation for natural language processing. *Symmetry* 11(11):1393.
- Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; y Duchesnay, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

- Pennebaker, J. W.; Francis, M. E.; y Booth, R. J. 2001. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates* 71(2001):2001.
- Pennebaker, J. W.; Mehl, M. R.; y Niederhoffer, K. G. 2002. Psychological Aspects of Natural Language Use: Our Words, Our Selves. *Annual Review of Psychology* 54(1):547–577.
- Pennington, J.; Socher, R.; y Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.
- Pizarro, J. 2019. Using n-grams to detect bots on twitter: notebook for pan at clef 2019. In *Proceedings of the Ninth International Conference of the CLEF Association (CLEF 2019)*.
- Porter, M. F. 2001. Snowball: A language for stemming algorithms.
- Rangel, F., y Rosso, P. 2019. Overview of the 7th Author Profiling Task at PAN 2019: Bots and Gender Profiling. *CLEF 2019 Labs and Workshops, Notebook Papers. CEUR Workshop Proceedings*.
- Rangel, F.; Rosso, P.; Koppel, M.; Stamatatos, E.; y Inches, G. 2013. Overview of the author profiling task at PAN 2013. *CLEF Conference on Multilingual and Multimodal Information Access Evaluation* 352–365.
- Rangel, F.; Rosso, P.; Verhoeven, B.; Daelemans, W.; Potthast, M.; y Stein, B. 2016. Overview of the 4th author profiling task at PAN 2016: Cross-genre evaluations. *CEUR Workshop Proceedings* 1609:750–784.
- Rude, S.; Gortner, E.-M.; y Pennebaker, J. 2004. Language use of depressed and depression-vulnerable college students. *Cognition & Emotion* 18(8):1121–1133.
- Schuster, M., y Paliwal, K. K. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing* 45(11):2673–2681.
- Stamatatos, E.; Daelemans, W.; Verhoeven, B.; Juola, P.; López-López, A.; Pott-hast, M.; y Stein, B. 2015. Overview of the 3rd Author Profiling Task at PAN 2015. *CLEF 2015 Labs and Workshops, Notebook Papers. CEUR Workshop Proceedings* 1391(31):898–927.

- Trotzek, M.; Koitka, S.; y Friedrich, C. M. 2018. Word embeddings and linguistic metadata at the clef 2018 tasks for early detection of depression and anorexia. In *CLEF (Working Notes)*.
- Van Dyk, D. A., y Meng, X.-L. 2001. The art of data augmentation. *Journal of Computational and Graphical Statistics* 10(1):1–50.
- Vapnik, V., y Chervonenkis, A. Y. 1964. A class of algorithms for pattern recognition learning. *Avtomat. i Telemekh* 25(6):937–945.
- Wang, S.; Liu, W.; Wu, J.; Cao, L.; Meng, Q.; y Kennedy, P. J. 2016. Training deep neural networks on imbalanced data sets. In *2016 international joint conference on neural networks (IJCNN)*, 4368–4374. IEEE.
- Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; y Bowman, S. R. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.
- Wang, Y.-T.; Huang, H.-H.; y Chen, H.-H. 2018. A neural network approach to early risk detection of depression and anorexia on social media text. In *CLEF (Working Notes)*.
- Wei, J., y Zou, K. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 6383–6389.
- Wu, J.; Li, L.; y Wang, W. Y. 2018. Reinforced co-training. *arXiv preprint arXiv:1804.06035*.
- Xie, Q.; Dai, Z.; Hovy, E.; Luong, M.-T.; y Le, Q. V. 2019. Unsupervised data augmentation. *arXiv preprint arXiv:1904.12848*.
- Yang, Y., y Pedersen, J. O. 1997. A comparative study on feature selection in text categorization. In *Icml*, volume 97, 35.
- Zhang, Y., y Wallace, B. 2015. A sensitivity analysis of (and practitioners’ guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.

- Zhang, J.; Lertvittayakumjorn, P.; y Guo, Y. 2019. Integrating semantic knowledge to tackle zero-shot text classification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 1031–1040.
- Zhang, X.; Zhao, J.; y LeCun, Y. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, 649–657.
- Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; y Yang, Y. 2017. Random erasing data augmentation. *arXiv preprint arXiv:1708.04896*.