UNIVERSIDAD
DE ANTIOQUIA

1 8 0 3

Aplicación del procesamiento de lenguaje natural para verificación de identidad.

Felipe Orlando López Pabón.

Universidad de Antioquia
Facultad de Ingeniería
Departamento de Ingeniería Electrónica y Telecomunicaciones
Medellín, Colombia
2019

Aplicación del procesamiento de lenguaje natural para verificación de identidad.

Felipe Orlando López Pabón
Estudiante de pregrado en Ingeniería Electrónica

Trabajo de grado presentado para optar por el título de:
Ingeniero Electrónico

Asesor
PhD. Juan Rafael Orozco Arroyave

Co-Asesor
MSc. Juan Camilo Vasquez Correa

Línea de investigación
Procesamiento digital de señales y análisis de patrones
Grupo de Investigación en Telecomunicaciones Aplicadas
GITA

Universidad de Antioquia
Facultad de Ingeniería
Departamento de Ingeniería Electrónica y
Telecomunicaciones
Medellín, Colombia
2019

Agradecimientos

Principalmente, agradezco a mi mamá Luz Marina Pabón Tellez y a mi papá José Ramiro López López, quienes, con amor incondicional y comprensión, son los encargados de brindarme las bases y todo lo necesario para obtener este gran logro, a parte de las motivaciones y el gran ejemplo que cada uno de ellos me ha dado. A mis hermanos César y Camilo, quienes, con paciencia y dedicación me han acompañado en esta largo camino. A mi novia Adriana y mi hija Antonella, que, desde los últimos años se han convertido en mi motivación para no detenerme en la búsqueda de este sueño. A toda mi familia, porque en algún momento necesité apoyo de ellos, en especial de Jair Mazabuel y Cristina Mendez.

Agradezco a mis amigos y compañeros de carrera Daniel Escobar y Cristian Rios, quienes, en más de una ocasión, me extendieron la mano para ayudarme tanto en ámbitos académicos como en ámbitos personales, y que, también me ayudaron en el desarrollo de este trabajo. A mis compañeros de línea del Grupo de investigación GITA Paula Pérez, Luis Felipe Gómez, Luis Felipe Parra, Surley Berrio, Sebastian Guerrero y Tomas Arias, que, con paciencia, supieron ayudarme en todo lo que necesité.

Por último, quiero agradecer a mi asesor Juan Rafael Orozco y a mi co-asesor Juan Camilo Vasquez, porque gracias a ellos, fue posible el desarrollo de este trabajo, quienes, con dedicación y paciencia me guiaron y me brindaron las bases para crecer como profesional.

Índice

Índice de figuras

Índice de tablas

Resumen

Las empresas utilizan la verificación de identidad para garantizar que los usuarios y los clientes proporcionen información asociada con la identidad de una persona real. En un ámbito académico, de igual forma, es relevante controlar que la información que los estudiantes dan es verídica y que los procesos que se realizan en las instituciones (tales como trabajos, exámenes, exposiciones, entre otras) sean realizados por aquellas personas que realmente están matriculadas, esto con el fin de controlar el fraude. La verificación de identidad mitiga el riesgo de fraude mediante diferentes estrategias, entre las cuales, las más exitosas son las basadas en biometría. En Colombia, según datos recientes publicados por el ministerio de Educación Nacional, la educación virtual muestra una tasa de crecimiento de 98,9 % en el número de matrículas de educación superior, lo cual indica que, en varias instituciones, como por ejemplo, en la Universidad de Antioquia, hay gran cantidad de estudiantes en la modalidad virtual. A pesar de todos los grandes beneficios de la modalidad virtual de educación, esta trae consigo diversos problemas, entre ellos, suplantación de identidad y fraude en actividades evaluativas. Para resolver estos problemas, en este trabajo se propone desarrollar, mediante el procesamiento de lenguaje natural y algoritmos de aprendizaje automático, una metodología que permita verificar a qué grupo de estilo lingüístico de escritura (existirán 3 grupos) pertenecen más de 100 usuarios pertenecientes a la comunidad universitaria, los cuales se registraron en una plataforma virtual y realizaron dos tareas que consisten en argumentar una solución a problemas que está sufriendo el país actualmente y dar una opinión sobre un tema social.

Los resultados muestran, a pesar de la poca cantidad de datos y la calidad de los textos, que es posible encontrar diferencias entre estilos de escritura de los usuarios de acuerdo con su nivel escolar, obteniendo resultados de eficiencia en clasificación biclase (niveles inferiores vs niveles superiores) de hasta 75 % y eficiencia en la clasificación triclase (niveles inferiores vs niveles intermedios vs niveles superiores) de hasta 53.3 %. Otro resultado obtenido muestra, que mediante modelos de mezclas Gaussianas, se logra identificar, de una manera muy certera, los usuarios pertenecientes al grupo de usuarios de niveles de escolaridad intermedios y superiores, y diferenciarlos de usuarios con niveles bajos de escolaridad.

1. Introducción.

1.1. Contexto

Las empresas utilizan la verificación de identidad para garantizar que los usuarios y los clientes proporcionen información asociada con la identidad de una persona real. Las transacciones online representan una cantidad cada vez mayor de negocios, por ello, saber con quién se está haciendo negocios es un requisito comercial fundamental para las empresas. En un ámbito académico, de igual forma, es relevante controlar que la información que los estudiantes dan es verídica y que los procesos que se realizan en las instituciones (tales como trabajos, exámenes, exposiciones, entre otras) sean realizados por aquellas personas que realmente deben hacerlas, esto con el fin de controlar el fraude. La verificación de identidad mitiga el riesgo de fraude mediante diferentes estrategias, entre las cuales las más exitosas son las basadas en biometría, la cual es descrita como el área dentro de la computación científica que permite reconocer a un individuo basándose en sus rasgos físicos o de comportamiento como el rostro, la huella digital, geometría de la mano, iris, pulsación de tecla, firma, voz, etc [1]. En Colombia, según datos recientes publicados por el ministerio de Educación Nacional, la educación virtual muestra una tasa de crecimiento en el número de matrículas de educación superior desde el 2011 (13,6 %) hasta el 2014 (90 %). En 2015 se moderó, pero en 2016 volvió a repuntar hasta llegar a 98,9 % [2]. Un ejemplo claro de educación virtual es la plataforma Ingeni@ de la Facultad de Ingeniería que ofrece la Universidad de Antioquia, la cual busca promover e instalar otras formas de enseñar, aprender y producir conocimiento de manera colaborativa, y que, contribuya a ampliar la oferta educativa que tiene la universidad a nivel de pregrado, posgrado y educación continua [3]. A pesar de todos los grandes beneficios antes mencionados, la modalidad virtual trae consigo diversos problemas, entre ellos suplantación de identidad y fraude en actividades evaluativas. Aunque para acceder a cursos virtuales cada estudiante tiene un nombre de usuario y una contraseña, esto no garantiza que las personas que ingresan al curso sean realmente usuarios, por lo cual el docente no sabe con total certeza si la persona que realiza un trabajo, tarea o asignación realmente es el estudiante matriculado o si es un impostor. En un contexto educativo, la definición de plagio hace referencia a copiar obras de terceros haciéndolas pasar por propias, es decir, es un intento de engaño hacia el profesor, autoridades académicas y hacia la propia institución, pues demuestra (si se logra detectar) que no existió mayor esfuerzo personal por llevar a cabo el trabajo asignado [4]. Con el fin de lograr mejores calificaciones, algunos estudiantes permiten que otras personas realicen los exámenes

o pruebas que se le asignan, obviamente, personas con mayor dominio del tema y con mayor experiencia sobre la asignación establecida, lo cual es muy común, por ejemplo, en los exámenes de inglés en la modalidad virtual de la Facultad de Ingeniería. Para resolver este problema, se ha trabajado en estrategias para regular el fraude, buscando que éstas no adicionen complejidades extras al usuario. A medida que las deficiencias de los sistemas de acceso tradicionales basados en contraseñas se vuelven cada vez más agudas, los investigadores han centrado su atención en la biometría y está empezando a ganar aceptación como un método legítimo para determinar la identidad de los individuos.

En este trabajo se propone desarrollar, mediante el procesamiento de lenguaje natural y algoritmos de aprendizaje automático, una metodología que permita verificar a qué grupo de estilo lingüístico (existirán 3 grupos, explicados más adelante) pertenece un usuario, el cual debe registrarse en una plataforma virtual y realizar algunas tareas que consisten en argumentar una solución a problemas que está sufriendo el país actualmente y dar una opinión sobre un tema social.

1.2. Estado del arte

En los últimos 20 años, el estudio sobre la biometría basada en el análisis de texto para uso de autenticación y verificación ha crecido en diversos temas, como por ejemplo, en [5] identificaron a los usuarios de un chat, a partir de su comportamiento verbal en la plataforma. Los autores extraen características como la frecuencia de palabras y caracteres, el uso de signos de puntuación, errores ortográficos intencionales y no intencionales, el uso de vocabulario, la longitud de las oraciones y el ordenamiento particular de las palabras. Los autores proponen un modelo basado en perfiles (*Profile based approach*, PBA), en el cual se compara el texto de consulta con un modelo del autor candidato, y se determina la probabilidad que el documento haya sido escrito por el autor. Como resultados, se obtuvieron índices de reconocimiento de hasta 98.5 % haciendo uso del modelo PBA en la base de datos COPA, la cual consta de datos demográficos, estadísticos, registros de juegos, interacciones y quejas de 403 jugadores únicos, que jugaron al menos una vez. Mientras que para la base de datos Ekşisözlük, la cual contiene entradas sobre diferentes temas escritos en turco por 252 usuarios registrados, también usando PBA, se logra un índice de reconocimiento de 87.9 % . Lo cual indica que el enfoque PBA es adecuado para la atribución de autor en conjuntos de datos informales de chat. También se indica que para conjuntos cerrados de tamaño moderado (es decir, hasta 1000 autores), y con una cantidad bastante pequeña de texto de consulta, es posible identificar autores a partir de sus comunicaciones de

chat en línea.

En [6], los autores diseñan y examinan un enfoque automático que adopta características de estilo de escritura para estimar la reputación de los usuarios en las redes sociales. Éste evalúa el rendimiento de la clasificación de los métodos más modernos bajo distintas formas de definir buenas y malas clases de reputación de usuario basadas en los datos recopilados. Se calculan cuatro tipos de características de estilo de escritura: léxica (F1), sintáctica (F2), estructural (F3) y específica de contenido (F4). Las características se clasifican usando ocho algoritmos diferentes, basados en C4.5 (Arboles de decisión), Red neuronal (*Neural Network*, NN), Máquina de soporte vectorial (*Support Vector Machine*, SVM) y *Naïve Bayes* (NB). Se obtiene el mejor resultado de clasificación en las reputaciones de los usuarios, utilizando el conjunto de características F1 + F2 + F3 + F4 y la técnica SVM, el cual alcanza una tasa de aciertos de hasta el 94.50 %. Similarmente, en otro trabajo, como el desarrollado en [7], desarrollan un nuevo tipo de características léxicas para su uso en la clasificación de texto estilístico, basada en taxonomías de varias funciones semánticas de ciertas palabras o frases de elección. Logran demostrar la utilidad de tales características para las tareas de clasificación de textos estilísticos haciendo uso de la versión *Sequential minimal optimization* (SMO) del algoritmo de aprendizaje de la SVM, con un núcleo lineal, para determinar la identidad del autor (porcentajes de precisión cercanas a 90 %), la nacionalidad (superior a 90 %), el género de los personajes literarios (porcentaje de precisión aproximadamente de 75 %), el sentimiento de un texto, evaluación positiva/negativa, (92 % de precisión) y el carácter retórico de los artículos de revistas científicas (con un 86.8 %). Además de lo anterior, muestran cómo el uso de características funcionales ayuda a comprender mejor las diferencias estilísticas entre diferentes tipos de textos.

Similarmente, en [8], detectan automáticamente el género 3000 usuarios de Twitter. Desarrollan un sistema que pueda usar el conocimiento para interpretar los estilos lingüísticos utilizados por los géneros, considerando diferentes conjuntos de características, que son relativamente independientes del texto, como las palabras funcionales (*Function words*, FW) que consisten en palabras que tienen poco significado léxico o ambiguo, pero sirven para expresar relaciones gramaticales con otras palabras dentro de una oración, o especificar la actitud o el estado de ánimo del hablante; y los n-gramas de parte del discurso(*n-grams Part of speech* , ngrampos) que hacen referencia a un tipo de modelo de lenguaje probabilístico para predecir el siguiente elemento en una secuencia en forma de un modelo de Markov de orden “n-1”. Probaron varios conjuntos de características diferentes utilizando dos clasificadores diferentes; Naïve Bayes (NB) y algoritmos de máxima entropía. Los mejores resultados muestran una precisión de aproximadamente el 71 %,

mediante el conjunto de caracteristicas ngrampos y el clasificador NB.

1.3. Hipótesis

Las personas mejoran su capacidad de redacción a medida que avanzan en su carrera Universitaria, por lo cual, el estilo lingüístico de una persona en los primeros niveles es diferente al de una persona en niveles intermedios y también al de una persona en últimos niveles o el de una persona con un título universitario.

1.4. Objetivos

1.4.1. Objetivo general

Desarrollar algoritmos que permitan diferenciar los estilos lingüísticos de personas que pertenezcan a la comunidad universitaria y que se registran en una plataforma web mediante técnicas de procesamiento de lenguaje natural (*Natural Language Processing*, NLP).

1.4.2. Objetivos específicos

1. Extraer características lingüísticas relevantes asociadas a textos escritos realizados por los usuarios de la página web.

2. Implementar sistemas de clasificación y construir modelos de usuario que permitan diferenciar personas por su estilo lingüístico.

3. Evaluar la utilidad de las medidas de NLP para diferenciar estilos lingüísticos.

4. Medir el desempeño del sistema de clasificación mediante medidas de acierto como los porcentaje de: eficiencia, precisión, sensibilidad, especificidad, F1-score y matriz de confusión.

1.5. Contribución de este trabajo

El estilo lingüístico de tres grupos de personas es considerado teniendo en cuenta la siguiente distribución: Grupo 1 (G1): personas que están en los primeros niveles de la Universidad, Grupo 2 (G2): aquellas personas que están en niveles intermedios y Grupo 3 (G3) compuesto por personas que están en los último niveles y también por personas que están realizando algún posgrado o que ya son profesionales. Se estima el estilo lingüístico por medio de diferentes características tales como *Bag of Words* (BoW), *Term*

frequency – Inverse document frequency (TF-IDF), Word2vec, *Global Vectors* (GloVe) y características gramaticales. Se implementaron dos sistemas de clasificación con el fin de medir la importancia de las características en el estilo lingüístico del usuario: uno basado en una SVM y otro basado en Bosques Aleatorios (*Random Forest*, RF). También se consideran Modelos de mezclas Gaussianas (*Gaussian mixture model*, GMM), para modelar el estilo lingüístico del usuario y posteriormente medir la eficiencia con la que el algoritmo predice la pertenencia del usuario a uno de los 3 grupos (G1, G2 o G3), teniendo en cuenta una medida llamada distancia de de Bhattacharyya.

2. Marco teórico

2.1. Extracción de características

2.1.1. Bag of Words (BoW)

Bolsa de palabras (*Bag of Words*, BoW) es un método para extraer características de documentos de texto. Crea un vocabulario de todas las palabras únicas que aparecen en todos los documentos del conjunto de entrenamiento, donde no se tiene en cuenta la gramática e incluso el orden de las palabras, pero mantiene la multiplicidad [9]. A un alto nivel, BoW incluye los pasos mostrados en la Figura 1

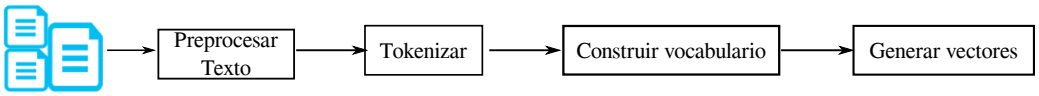

Figura 1. Esquema del método BoW.

Los vectores generados se pueden usar en algoritmos de aprendizaje automático para la clasificación y predicción de documentos, donde la longitud de dichos vectores siempre será igual al tamaño del vocabulario. Cuando se considera un documento grande, donde el vocabulario generado es enorme, pueden resultar vectores con muchos valores iguales a 0. Esto se le conoce como un vector disperso (*sparse vector*), los cuales requieren más memoria y recursos computacionales al modelar y la gran cantidad de posiciones o dimensiones puede hacer que el proceso de modelado sea muy desafiante para los algoritmos tradicionales. El modelo BoW solo considera si una palabra conocida aparece en un documento o no, esto da la idea de que documentos similares tendrán recuentos de palabras similares entre sí. En otras palabras, cuanto más similares sean las palabras en dos documentos, más similares podrán ser los documentos. Entre las limitaciones de BoW se resaltan:

1. Significado semántico: el enfoque BoW básico no considera el significado de la palabra en el documento. Ignora completamente el contexto en el que se usa. La misma palabra se puede utilizar en múltiples lugares según el contexto o palabras cercanas.

2. Tamaño del vector: para un documento grande, el tamaño del vector puede ser enorme, lo que resulta en un costo computacional alto. Es posible que se deba ignorar las palabras según su relevancia para su caso de uso.

En lugar de dividir nuestra oración en una sola palabra (*mono-gram*), podemos dividir en dos palabras (*bi-gram* o *2-gram*). A veces, la representación *bi-gram* parece ser mucho mejor que la representación *mono-gram*. Estos a menudo se pueden representar usando notación *N-gram*.

2.1.2. Term frequency – Inverse document frequency (TF-IDF)

TF-IDF significa frecuencia de término - frecuencia inversa de documento. El significado de TF-IDF es una medida estadística que se utiliza para evaluar la importancia de una palabra para un documento de una colección o un corpus. La importancia aumenta proporcionalmente al número de veces que aparece una palabra en el documento, pero es compensada por la frecuencia de la palabra en el corpus [10].

Frecuencia de término (*Term Frequency*, TF): Es una medida de la frecuencia de la palabra en el documento actual. Dado que cada documento es diferente en longitud, es posible que un término aparezca mucho más veces en documentos largos que en documentos más cortos. El término frecuencia se divide a menudo por la longitud del documento para normalizar. En la Ecuación 1 se muestra la forma de obtener el valor de TF.

$$Tf(t) = \frac{\text{Número de veces que aparece el término } t \text{ en el documento}}{\text{Número total de términos en el documento}} \quad (1)$$

Frecuencia inversa de documento (*Inverse Document Frequency*, IDF): Es una medida de cuán rara es la palabra entre los documentos. Entre más raro es el término, mayor es la puntuación IDF. En la Ecuación 2 se muestra cómo se calcula:

$$IDF(t) = \log\left(\frac{\text{Número total de documentos}}{\text{Número total de documentos con el término } t}\right) \quad (2)$$

Finalmente,

$$\text{TF} - \text{IDF} = TF \cdot IDF \quad (3)$$

De la Ecuación 3, se concluye que sí TF-IDF=0 para una palabra en especifíco en todos los documentos, dicha palabra no es muy informativa, ya que aparece en todos los documentos.

2.1.3. Word2vec

En el procesamiento de lenguaje natural, a menudo se mapean palabras en vectores que contienen valores numéricos para que la máquina pueda entenderlo. La inserción de palabras (*Word Embeddings*) es un tipo de mapeo

que permite que las palabras con un significado similar tengan una representación similar. Una forma tradicional de representar palabras es vectores *one-hot*, que es esencialmente un vector con solo un elemento objetivo que es 1 y los otros 0. La longitud del vector es igual al tamaño del vocabulario único total del corpus. Convencionalmente, estas palabras únicas están codificadas en orden alfabético. Es decir, se espera que los vectores one-hot para las palabras que comienzan con "a" tengan el objetivo "1" en un índice inferior, mientras que los de las palabras que comienzan con "z" tengan el objetivo "1" en un índice más alto. En la Figura 2 podemos observar un ejemplo.

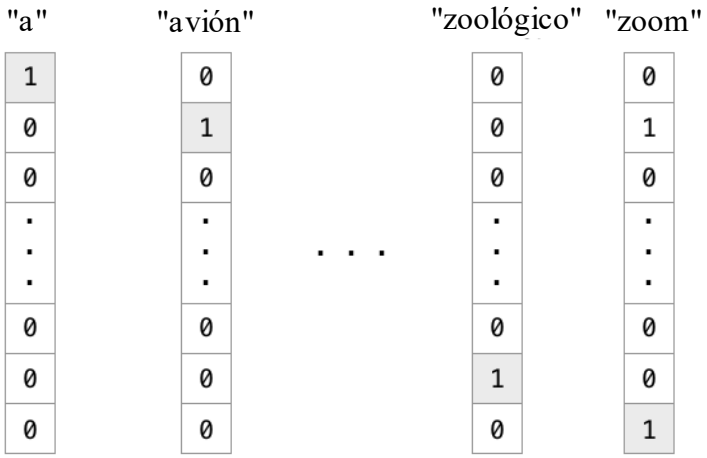

Figura 2. Ejemplo codificación *one-hot*.

Aunque esta representación de palabras es simple y fácil de implementar, hay varios problemas. Primero, no puede inferir ninguna relación entre dos palabras dada su representación única. Por ejemplo, la palabra "querer" y "amar", aunque tienen un significado similar, sus objetivos "1" están lejos el uno del otro. Por otro lado, la dispersión es otro problema, ya que hay numerosos "0" redundantes en los vectores. Esto significa que se está desperdiciando mucho espacio, por lo cual se necesita una mejor representación de las palabras para resolver estos problemas.

Word2Vec es una solución eficiente para estos problemas. Word2Vec usa las palabras cercanas para representar las palabras de destino con una red neuronal poco profunda cuya capa oculta codifica la representación de la palabra [11]. En términos generales, las representaciones vectoriales de una palabra en particular se le conoce como *Word Embeddings*, las cuales se pueden obtener utilizando dos métodos (ambos con redes neuronales): *Skip Gram* y *Continuos Bag Of Words* (CBOW).

Para *Skip Gram*, la entrada es la palabra de destino, mientras que las salidas son las palabras que rodean las palabras de destino. Por ejemplo, en la oración "Yo tengo un perro lindo", la entrada sería "un", mientras que la salida es "yo", "tengo", "lindo" y "perro", asumiendo que el tamaño de la ventana es 5. Todos los datos de entrada y salida son de la misma dimensión y están codificados de la manera *one-hot*. La red contiene 1 capa oculta cuya dimensión es igual al tamaño de incrustación (*embedding*), que es más pequeño que el tamaño del vector de entrada / salida. Al final de la capa de salida, se aplica una función de activación de softmax para que cada elemento del vector de salida describa la probabilidad de que aparezca una palabra específica en el contexto. La Figura 3 muestra la estructura de la red.

Las incrustaciones de palabras objetivo pueden obtenerse extrayendo las capas ocultas después de introducir la representación *one-hot* de esa palabra en la red. Con *Skip Gram*, la dimensión de representación disminuye desde el tamaño del vocabulario (V) hasta la longitud de la capa oculta (N). Además, los vectores son más "significativos", lo que implica que describen mejor la relación entre palabras.

Para el caso de CBOW, el cual es en el que haremos énfasis en este trabajo, es muy similar a skip-gram, excepto que intercambia la entrada y la salida, es decir, se toma el contexto de cada palabra como entrada e intenta predecir la palabra correspondiente al contexto. La estructura de la red neuronal se muestra en la Figura 3, donde se toman las C palabras de contexto. Cuando Wvn se utiliza para calcular las entradas de la capa oculta, se toma un promedio de todas estas C palabras de entrada de contexto.

Cuando el contexto de la palabra solo está representado por una sola palabra, la red neuronal para el modelo CBOW se ve como en la Figura 4. Como se puede ver, hay una capa de entrada, una capa oculta y, finalmente, una capa de salida. La función de activación de la capa oculta es la identidad $a = 1$ (Generalmente, llamada función de activación lineal) y la función de activación para la salida es una Softmax.

La entrada o la palabra de contexto es un vector codificado en *one-hot* de tamaño V. La capa oculta contiene N neuronas y la salida es nuevamente un vector de longitud V con los elementos que son los valores de softmax (función softmax). Los términos de la Figura 4, que son similares a los usados en la Figura 3, son: $W \in \mathbb{V} \times \mathbb{N}$, que hace referencia a la matriz de pesos que mapea la entrada x a la capa oculta y $W' \in \mathbb{N} \times \mathbb{V}$ es la matriz de pesos que mapea las salidas de la capa oculta a la capa de salida final. Las neuronas de la capa oculta copian la suma ponderada de entradas a la siguiente capa.

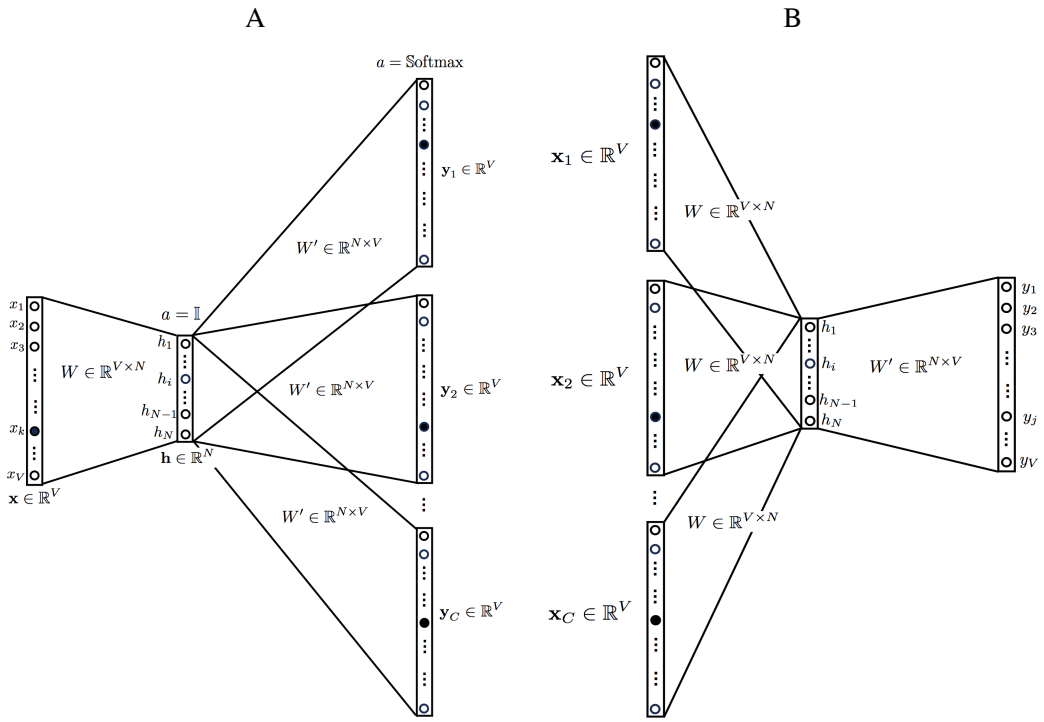

Figura 3. Topología de modelos utilizados en Word2Vec. A) *Skip Gram*, B) CBOW. **x**: Vector en codificación *one-hot* de la palabra, **h**: Capa oculta de N neuronas, N también es el número de dimensiones para representar la palabra. **a**: Función de activación. Figura adaptada de [11].

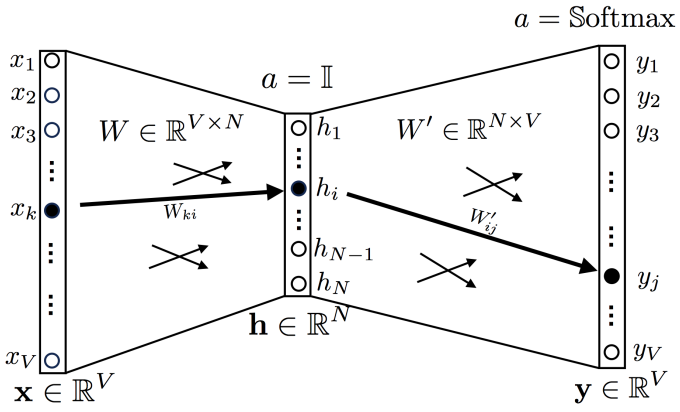

Figura 4. Topología red neuronal modelo CBOW con una sola palabra de contexto. Figura tomada de [11].

2.1.4. Global Vectors for Word Representation (GloVe)

El modelo GloVe, obtiene vectores de palabras al examinar las coocurrencias de ellas dentro de un corpus [12]. Antes de entrenar el modelo, se debe construir una matriz de co-ocurrencia X, donde una celda X_{ij} representa la frecuencia con la que aparece la palabra i en el contexto de la palabra j. Se recorre el corpus solo una vez para construir la matriz X y, a partir de entonces, se utilizan estos datos de co-ocurrencia en lugar del corpus.

Una vez que X está lista, la tarea es generar los vectores en un espacio continuo para cada palabra que observamos en el corpus. Se producirán vectores con una restricción suave para cada par de palabras, palabra i y palabra j de la forma

$$\vec{w_i}^{\intercal}\vec{w_j} + b_i + b_j = \log(X_{ij}) \tag{4}$$

Donde $\boldsymbol{w_i}$ y $\boldsymbol{w_j}$ son vectores de palabras, b_i y b_j son términos de sesgo escalar asociados con las palabras i y j, respectivamente. En palabras intuitivas, se quiere construir vectores de palabras que retengan cierta información útil acerca de cómo coexisten cada par de palabras i y j. Haremos esto minimizando una función objetivo J, que evalúa el error cuadrático medio de la ecuación 4, ponderada con una función f

$$J = \sum_{i=1}^{V}\sum_{j=1}^{V} f(X_{ij})(\vec{w_i}^{T}\vec{w_j} + b_i + b_j - \log(X_{ij}))^2 \tag{5}$$

Se elige f tal que ayude a evitar que los pares de palabras comunes (aquellos con valores X_{ij} grandes) desvíen demasiado nuestro objetivo

$$X_{ij} = \begin{cases} \left(\frac{X_{ij}}{x_{\max}}\right)^{\alpha}, \text{si } X_{ij} < x_{\max} \\ 1, \text{en otro caso.} \end{cases} \tag{6}$$

Donde $x_{\max}$ hace referencia al máximo valor de co-ocurrencia que puede tener una palabra i con una palabra j. Cuando se encuentran pares de palabras comunes (donde $X_{ij} > x_{\max}$) esta función acota su salida a 1, para todos los demás pares de palabras, se devuelve algún peso en el rango (0,1), donde la distribución de pesos en este rango se decide por α, el cual, es un hiperparámetro que controla la sensibilidad de los pesos a los recuentos de co-ocurrencia incrementados.

2.1.5. Características gramaticales

Se tienen en cuenta las siguientes ocho características gramaticales [13]: La legibilidad de un texto con la puntuación de lectura de Flesch (*Flesch*

reading score, FR) (Ecuación 7), el nivel de grado de Flesch-Kincaid (*Flesch-Kincaid grade level*, FG) (Ecuación 8), la densidad de proposiciones (*propositional density*, DP) (Ecuación 9), la densidad de contenido (*content density*, DC) de un texto (Ecuación 10). El conjunto de características se completa con los recuentos de parte de la voz (*Part-Of-Speech counts*): proporción de sustantivos a verbos (*Noun to Verb ratio*, NVR), proporción de sustantivos (*Noun ratio*, NR), proporción de pronombres (*Pronoun ratio*, PR) y proporción de conjunciones subordinadas a coordinadas (*subordinated to coordinated conjunctions ratio*, SCR) (Ecuaciones 11 - 14).

$$FG = 206{,}835 - 1{,}015\frac{\text{número de palabras}}{\text{número de oraciones}} - 84{,}6\frac{\text{número de sílabas}}{\text{número de palabras}} \tag{7}$$

$$FG = 0{,}39\frac{\text{número de palabras}}{\text{número de oraciones}} + 11{,}8\frac{\text{número de sílabas}}{\text{número de palabras}} + 15{,}59 \tag{8}$$

$$DP = \frac{\text{número de(verbos + adjetivos + preposiciones + conjunciones)}}{\text{número de palabras}} \tag{9}$$

$$DC = \frac{\text{número de(verbos + sustantivos + adjetivos + adverbios)}}{\text{número de palabras}} \tag{10}$$

$$NVR = \frac{\text{número de sustantivos}}{\text{número de verbos}} \tag{11}$$

$$NR = \frac{\text{número de sustantivos}}{\text{número de(sustantivos + verbos)}} \tag{12}$$

$$PR = \frac{\text{número de pronombres}}{\text{número de(pronombres + sustantivos)}} \tag{13}$$

$$SCR = \frac{\text{número de conjunciones subordinadas}}{\text{número de conjunciones coordinadas}} \tag{14}$$

2.2. Algoritmos de clasificación

2.2.1. Máquina de soporte vectorial (*Support vector Machine*, SVM)

El objetivo de un SVM es discriminar muestras de datos al encontrar un hiperplano separador que maximice el margen entre clases. La SVM de

margen suave permite errores en el proceso de encontrar el hiperplano óptimo. Esos errores son muestras de datos ubicadas en el lado equivocado de hiperplano pero dentro del margen óptimo.

La función de decisión de una SVM de margen suave se expresa de acuerdo con la Ecuación 15, donde ξ_n es una variable de holgura que penaliza la cantidad de errores permitidos en el proceso de optimización. $y_n \in \{-1, +1\}$ son las etiquetas de clase, $\phi(\mathbf{x}_n)$ es una función kernel para transformar el espacio de características $\mathbf{x}$ en un espacio dimensional superior donde se puede encontrar una solución lineal del problema. El vector de pesos $\mathbf{w}$ y el valor de sesgo b definen el hiperplano separador.

$$y_n \cdot (\mathbf{w}^T \phi(\mathbf{x}_n) + b) \geq 1 - \xi_n, \quad n = 1, 2, 3, \cdots, N \tag{15}$$

El problema de optimización para encontrar el hiperplano se define en la Ecuación 16, donde el hiperparámetro C controla la compensación entre ξ_n y el ancho del margen. Las muestras $\mathbf{x}_n$ que satisfacen la condición de igualdad en la Ecuación 15 se llaman vectores de soporte ($\mathbf{x}_m$).

$$\begin{aligned} \underset{\mathbf{w},b}{\text{minimize}} \quad & \tfrac{1}{2} ||\mathbf{w}||^2 + C \sum_{n=1}^{N} \xi_n \\ \text{subject to} \quad & y \cdot (\mathbf{w}^T \mathbf{x}_n + b) \geq 1 - \xi_n \\ & \xi_n \geq 0 \end{aligned} \tag{16}$$

Se considera la función kernel Gaussiano $\phi(\mathbf{x}_n) = \mathrm{e}^{-\gamma^2 ||\mathbf{x}_n - \mathbf{x}_m||^2}$, donde el hiperparámetro γ es el ancho de banda del kernel. Se pueden encontrar más detalles sobre el proceso para optimizar un SVM con una función de kernel en [14].

2.2.2. Bosques Aleatorios (*Random Forest*, RF)

Este es uno de los métodos de ensambe más comunes, que se basa en la combinación de múltiples algoritmos de árboles de decisión para tomar la decisión final. El algoritmo RF crea múltiples árboles tipo CART, cada uno entrenado con una muestra (después de aplicarle bootstrapping) de los datos de entrenamiento originales, y busca solo en un subconjunto seleccionado aleatoriamente de las variables de entrada para determinar una división (para cada nodo). Para la clasificación, cada árbol en el bosque aleatorio emite un voto unitario para la clase más popular en la entrada. La salida del clasificador está determinada por un voto mayoritario de los árboles [15]. Los hiperparámetros para optimizar el clasificador son la cantidad de árboles (N) y la profundidad máxima de los árboles (D).

La Figura 5 muestra el esquema del clasificador RF.

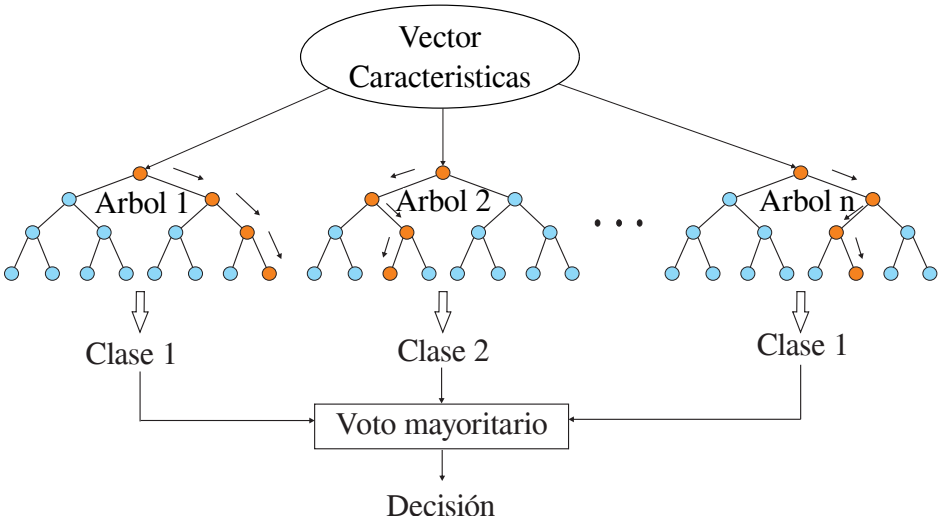

Figura 5. Esquema de un clasificador RF.

2.2.3. Modelos de mezclas Gaussianas (*Gaussian Mixture Model*, GMM)

Un variable escalar aleatoria y continua, x, tiene una distribución normal o Gaussiana si su función de densidad de probabilidad (FDP) es:

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] = \mathcal{N}(x;\mu,\sigma) \tag{17}$$

Donde μ y σ son la media y la desviación estándar de la variable aleatoria x respectivamente. Esta definición se puede extender a la distribución normal de múltiples variables. Si $\boldsymbol{x} = (x_1, x_2, ..., x_D)^\intercal$ es un vector aleatorio normal, también llamado variable aleatoria Gaussiana multivariada o vectorial, este cumple una distribución Gaussiana si:

$$p(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{D}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right] = \mathcal{N}(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) \tag{18}$$

Donde $\boldsymbol{\mu}$ es el vector de medias y $\boldsymbol{\Sigma}$ es la matriz de covarianzas de $\boldsymbol{x}$.

Las distribuciones Gaussianas son comunmente usadas en el campo de la ingeniería, no solo por sus características computacionales altamente deseables, sino también, por su capacidad de describir de una manera adecuada los datos obtenidos de fenómenos naturales del mundo real, gracias a la ley de los grandes números [16].

Una distribución general y más completa a la anterior, es la distribución de mezclas Gaussianas. Un vector aleatorio sigue un distribución de mezclas Gaussianas si su FDP puede ser descrita así:

$$\begin{aligned} p(\boldsymbol{x}) &= \sum_{m=1}^{M} \frac{c_m}{(2\pi)^{\frac{D}{2}} \left|\boldsymbol{\Sigma_m}\right|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}\left(\boldsymbol{x}-\boldsymbol{\mu_m}\right)^T \boldsymbol{\Sigma_m}^{-1}\left(\boldsymbol{x}-\boldsymbol{\mu_m}\right)\right] \\ &= \sum_{m=1}^{M} c_m N(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) \end{aligned} \tag{19}$$

Donde c_m es el peso asociado a la m-ésima componente Gaussiana y cumple $\sum_{m=1}^{M} c_m = 1$. Se hace uso de los GMM para lograr modelar el estilo lingüístico del usuario y posteriormente medir la distancia o la similitud del GMM del usuario con el GMM del grupo al que pertenece (G1, G2 o G3). Esto, debido a que los GMM permiten modelar un conjunto de datos, a partir de la suma ponderada de un numero finito de FDPs Gaussianas, donde cada distribución busca modelar una sub-población, para que en conjunto la mezcla de Gaussianas modele toda población en general.

2.3. Reducción de dimensionalidad

2.3.1. Análisis Lineal Discriminante (*Linear Discriminant Analysis*, LDA)

El método LDA es utilizado comúnmente como un método de reducción de dimensión para ayudar a reducir el costo computacional y además evitar sobre ajustes de modelo al minimizar el error en la optimización de parametros [17]. Donde, para cada muestra, se consideran las etiquetas de clase y el objetivo principal es encontrar las direcciones (discriminantes lineales) que maximizan la separabilidad entre clases. Para realizar LDA, se siguen los siguientes pasos:

1. Calcular las medias de los vectores D dimensionales para las distintas clases incluidas en el conjunto de datos.

2. Calcular las matrices de covarianza para cada clase y, entre las dos clases.

3. Se calculan los vectores propios (*eigenvectores*) $\in \{e_1, e_2, \cdots, e_D\}$ con sus correspondientes valores propios (*eigenvalores*) $\in \{\lambda_1, \lambda_2, \cdots, \lambda_D\}$.

4. Se organizan los eigenvalores de forma descendente y se eligen los k eigenvectores con sus correspondientes eigenvalores más altos para formar una matriz $W \in \mathbb{D} \times \mathbb{K}$

Esta nueva matriz se utiliza para transformar las muestras en el nuevo subespacio. Para esto se recurre a la expresión matemática $Y = X \cdot W$.

Cuando se aplica el método LDA para reducción de características, cada vector de características tiene su respectiva etiqueta de clase. Es decir, en la matriz de características cada muestra tiene una etiqueta que indica la clase a la que pertenece. Las etiquetas se utilizan para calcular la matriz de covarianza de cada clase para encontrar las componentes con máxima variación en los datos, podemos ver un ejemplo de una reducción de dimensión de 2 a 1 en la Figura 6.

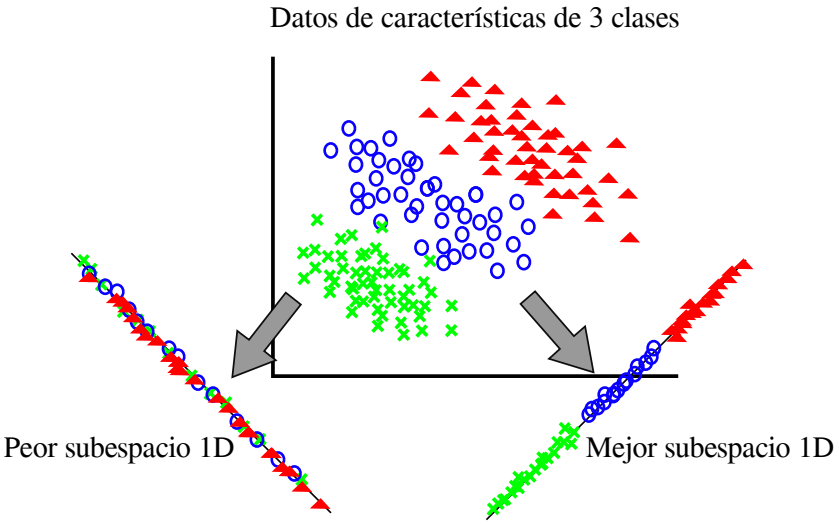

Figura 6. Análisis Lineal Discriminante.

2.4. Medidas de desempeño

Matriz de confusión: En general, cuando se realiza un proceso que involucre reconocimiento de patrones, se utiliza la llamada matriz de contingencia o de confusión, con la cuál se evalúa el desempeño del sistema dependiendo del número de aciertos y fallos en la etapa de clasificación de nuevos datos [18]. Una matriz de confusión para sistemas de clasificación biclase se muestra en la Tabla 1.

De acuerdo con esta matriz y tomando como referencia una de las clases en el conjunto de entrenamiento, se definen los siguientes términos:

Tabla 1. Matriz de confusión.

	Clase verdadera	
Clase estimada	Clase 0	Clase 1
Clase 0	TP	FP
Clase 1	FN	TN

- ✓ Verdadero positivo (*True positive*, TP): hace referencia a el número (o porcentaje) de patrones de clase 0 que el sistema clasifica correctamente como pertenecientes a la clase 0.
- ✓ Falso negativo (*False negative*,FN): corresponde a el número (o porcentaje) de patrones de clase 0 que el sistema clasifica incorrectamente como pertenecientes a la clase 1.
- ✓ Falso positivo (*False positive*, FP): es el número (o porcentaje) de patrones de clase 1 que el sistema clasifica incorrectamente como pertenecientes a la clase 0.
- ✓ Verdadero negativo (*True negative*, TN): es el número (o porcentaje) de patrones de clase 1 que el sistema clasifica correctamente como pertenecientes a la clase 1.

Eficiencia: Tasa de acierto o eficiencia (*Accuracy*, ACC): Esta medida es la proporción de patrones correctamente clasificados por el sistema:

$$ACC = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FN} + \mathrm{FP} + \mathrm{TN}} \tag{20}$$

Sensibilidad (SEN): La sensibilidad indica la capacidad del sistema para detectar los patrones de clase de referencia. Por ejemplo, el porcentaje de personas enfermas que están correctamente identificadas como personas portadoras de la condición.

$$SEN = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{21}$$

Especificidad (ESP): La especificidad indica la capacidad del sistema para rechazar los patrones que no pertenecen a la clase de referencia. Por ejemplo, el porcentaje de personas sanas que están identificadas correctamente como personas que no tienen la condición.

$$ESP = \frac{\mathrm{TN}}{\mathrm{FP} + \mathrm{TN}} \tag{22}$$

Precisión (PRE): Hace referencia a la proporción de resultados positivos que son verdaderos resultados positivos.

$$PRE = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{23}$$

F1-Score (F1): Es el medio armónico de precisión y sensibilidad.

$$F1 = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{24}$$

Coeficiente Kappa de Cohen (*Cohen's kappa coefficient*, κ):
El Coeficiente Kappa de Cohen, Ecuación 25, es una estadística robusta útil para pruebas de confiabilidad entre evaluadores (grado de acuerdo entre los evaluadores). Similar a los coeficientes de correlación, puede variar desde -1 a +1, donde 0 representa la cantidad de acuerdo que se puede esperar de una oportunidad aleatoria, y 1 representa un acuerdo perfecto entre los evaluadores. Al igual que con todas las estadísticas de correlación, kappa es un valor estandarizado y, por lo tanto, se interpreta de la misma manera en múltiples estudios [19].

$$\kappa = \frac{p_o - p_\epsilon}{1 - p_\epsilon} \tag{25}$$

Donde p_o es la probabilidad empírica de acuerdo en una etiqueta asignada a cualquier muestra, y p_ϵ es el acuerdo esperado cuando ambos evaluadores asignan etiquetas al azar. Cohen sugirió que el resultado de κ se interpretara de la siguiente manera: los valores ≤ 0 indican que no hay acuerdo, valores entre 0.01 y 0.20 como ningún acuerdo a leve, 0.21–0.40 como justo, 0.41–0.60 como moderado, 0.61–0.80 como sustancial y 0.81–1.00 como un acuerdo casi perfecto.

2.5. Distancia de Bhattacharyya

La distancia de Bhattacharyya, D_b, mide la similitud de dos distribuciones de probabilidad y está estrechamente relacionado con el coeficiente de Bhattacharyya, el cual se usa para determinar la proximidad relativa de las dos muestras consideradas. Este tipo de distancia se utiliza para medir la separabilidad de las clases en la clasificación y se considera más confiable que otro tipo de distancias, ya que, cuando dos clases tienen medios similares pero diferentes desviaciones estándar, la distancia Bhattacharyya crece dependiendo de la diferencia entre las desviaciones estándar.

Por ejemplo, si se tienen las siguientes dos distribuciones de probabilidad $f(x)$ y $g(x)$, la D_b seria de la forma:

$$D_b(f(x), g(x)) = -ln(B_C(f(x), g(x))), \tag{26}$$

donde B_C se conoce como el coeficiente de Bhattacharyya y se define para distribuciones de probabilidad continuas como

$$B_C(f(x), g(x)) = \int \sqrt{f(x)g(x)}. \tag{27}$$

3. Bases de datos

Para este trabajo, fue necesario crear una plataforma web que cumpliera la función de recolectar los textos de los usuarios que ingresaran y realizaran las tareas. Dicha plataforma de captura se desarrolló con la finalidad principal de construir la base de datos, la cual se compone de dos etapas principales: la fase de registro y la fase de captura, ver Figura 7. En la fase de registro, se le pide al usuario sus datos personales, como el nombre, la cédula, edad, genero, programa al que pertenece y nivel de escolaridad. Antes de permitir el registro, el usuario debe aceptar los términos y condiciones que permiten el uso de sus datos solamente con fines académicos. Posteriormente, en la fase de captura, el usuario debe realizar una serie de tareas en las cuales debe hacer uso del teclado. Dichas tareas se explicarán con mas detalle a continuación.

Figura 7. Página web para la recolección de los textos de los usuarios. En la izquierda se ve la etapa de registro y a la derecha se ve una de las dos tareas, la otra tarea es similar.

Para desarrollar y evaluar el sistema de clasificación, fue necesario recolectar datos de diferentes personas pertenecientes a la comunidad universitaria, mediante la plataforma de captura explicada anteriormente para poder generar los 3 Grupos de estílos lingüísticos: Grupo 1 (G1), Grupo 2 (G2) y Grupo 3 (G3). En el Grupo 1 se encuentran las personas que se encuentran en los niveles iniciales, es decir: 1,2 y 3 de sus estudios de pregrado, en el Grupo 2, las personas de niveles intermedios: 4,5,6,7 y 8, mientras que en el Grupo 3 se encuentran las personas en niveles superiores: 9,10, al igual que estudiantes de posgrado (Maestría y Doctorado) y personas profesionales.

Se solicitó a los usuarios, entrar a la página y diligenciar la información de registro, para posteriormente pasar a realizar las tareas. Ambas tareas

consisten en responder una pregunta con un mínimo 200 caracteres. En la Tabla 2 se explican con más detalle las tareas realizadas:

Tabla 2. Descripción de las tarea realizadas para la construcción de la base de datos.

Tarea	Descripción
1	Desde su área profesional, argumentar una posible solución frente a la contaminación de fuentes hídricas que está sufriendo el país actualmente.
2	Desde su punto de vista, cuente cómo le pareció la actuación de la selección Colombia en el mundial de Fútbol Rusia 2018.

Al final, se logra obtener un total de 141 usuarios, donde 111 de ellos realizaron la Tarea 1 y 30 realizaron la Tarea 2. De los 111 usuarios que realizaron la Tarea 1, 42 pertenecen al Grupo 1, 36 al Grupo 2 y 33 al Grupo 3, mientras que los 30 usuarios que realizaron la Tarea 2, están equitativamente distribuidos, es decir, 10 usuarios por grupo.

La información de todos los participantes para este estudio se muestra en la Tabla 3, la cual contiene información el numero de hombres, de mujeres, el promedio de edad, el número de estudiantes de pregrado, de maestría y doctorado y el número de profesionales.

Tabla 3. Información acerca de todos los participantes de este estudio. μ: promedio, σ: desviación estándar.

	Total		**Grupo 1 (G1)**		**Grupo 2 (G2)**		**Grupo 3 (G3)**	
	Hombres	Mujeres	Hombres	Mujeres	Hombres	Mujeres	Hombres	Mujeres
Número de personas	96	45	38	14	29	17	29	14
Edad	23.7 ± 5.5	24.5 ± 7.5	20.5 ± 3.2	20.2 ± 1.3	24.0 ± 4.9	23.5 ± 4.2	27.8 ± 5.9	29.9 ± 10.9
Estudiantes de Pregrado	85	37	38	14	29	17	18	6
Profesionales	3	5	-	-	-	-	3	5
Magisters	4	-	-	-	-	-	4	-
Doctores	4	3	-	-	-	-	4	3

4. Metodología

La metodología seguida en este trabajo se muestra en la Figura 8. En términos generales, el procedimiento comienza con un preprocesamiento de los textos adquiridos a través de la página web, que consiste en eliminar información irrelevante y que no aporta en la distinción entre usuarios (se remueven las letras mayúsculas, la puntuación, las *stopwords*, se realiza lematización y por último se remueven los acentos). Luego, procedemos a la extracción de características (que considera las características descritas en la sección 2.1), seguidas de: 1. Clasificación utilizando SVM o RF, o 2. Generación de modelos de usuarios mediante GMM y finalmente, el rendimiento de los algoritmos utilizados se mide utilizando diferentes métricas (eficiencia, sensibilidad, especificidad, precisión, F1-score, entre otras).

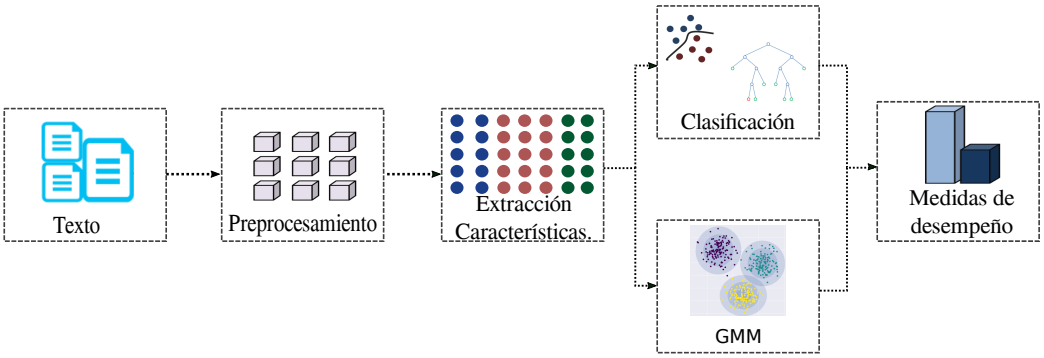

Figura 8. Diagrama de bloques de la metodología implementada en este estudio.

4.1. Experimentos

Se realizan 3 experimentos, (1) clasificación biclase (G1 vs G3) y clasificación triclase (G1 vs G2 vs G3) utilizando el clasificador SVM, (2) clasificación biclase (G1 vs G3) y clasificación triclase (G1 vs G2 vs G3) utilizando el clasificador RF y (3) generación de 3 modelos de grupos de usuario usando los 111 usuarios que realizaron la Tarea 1 y posteriormente generar 30 modelos, uno para cada usuario que realizó la Tarea 2.

Los experimentos (1) y (2) se realizan con el fin de lograr separar los usuarios por su nivel de escolaridad mediante algoritmos que buscan la mayor separación entre clases.

El experimento (3) se realiza con el fin de lograr modelar el estilo lingüístico del usuario y posteriormente medir la distancia de Bhattacharyya de los GMMs de los 30 usuarios que realizaron la Tarea 2 al GMM de cada uno de

los 3 modelos basados en los 111 usuarios de la Tarea 1 (G1, G2 o G3). Donde se tiene que, entre menor distancia, mayor es la posibilidad de pertenencia a ese grupo según el algoritmo.

4.1.1. Validación cruzada

Esta técnica se utiliza para los métodos de clasificación (SVM y RF), donde se realiza validación cruzada (*Cross-validation*, CV) de 10 particiones para el proceso de entrenamiento, es decir, los datos se dividen en 10, elegidos al azar, 9 de ellos se usan para el entrenamiento y 1 para la prueba. El proceso se repite 10 veces. Se aclara que el mismo usuario nunca se encuentra en los grupos de entrenamiento y de prueba al mismo tiempo. Se utiliza optimización de los parámetros C y γ (para el caso de la SVM), y de los parámetros $n-estimators$ y $max-depth$ (para el caso de RF), esto, con el fin de obtener mejores resultados. Para el caso de la SVM, se busca entre $C \in \{0.001, 0.01, ... ,1000, 10000\}$, y $\gamma \in \{0.001, 0.01, ... ,100,1000\}$, mientras que para RF, se busca entre $n-estimators \in \{5, 10, 15, 20, 50\}$ y $max-depth \in \{1, 2, 5, 10\}$ el mejor valor. Al final de las 10 repeticiones, se reporta en los resultados el valor de C y γ (SVM), $n-estimators$ y $max-depth$ (RF) que más se utilizan. Debe aclararse que los resultados que se obtienen con estos experimentos son optimistas, porque los parámetros se escogen con base en el acierto en el conjunto de prueba. Por esta razón, se realiza el experimento que será explicado a continuación.

4.1.2. Entrenamiento con Tarea 1 y prueba con Tarea 2

Se utiliza en los métodos de clasificación SVM y RF, al igual que para el método de los GMM. Para el caso de los métodos de clasificación, ya no se realiza validación cruzada, el procedimiento es el siguiente: se utilizan los 111 usuarios de la Tarea 1 para entrenar haciendo uso de los parámetros entrenados que arrojan los mejores resultados (de acuerdo a los resultados obtenidos con validación cruzada) y se prueba con los 30 usuarios que realizan la Tarea 2.

Para el método que usa los GMM, se generan 1 modelo para cada uno de los 3 grupos de usuario (G1, G2 y G3), es decir, 3 modelos de grupos, usando los 111 usuarios que realizaron la Tarea 1. Posteriormente, se generan 30 modelos, uno para cada usuario que realizó la Tarea 2 y se mide la distancia de cada uno de estos GMM a los 3 modelos de grupos de usuario generados inicialmente, así, se sabrá cual es la predicción del algoritmo respecto a qué grupo pertenece (G1, G2 o G3). Tanto los modelos de entrenamiento como los modelos de prueba, se realizaron con dos grupos de características: un gru-

po considerando Word2vec y otro considerando GloVe. Para ambos casos, se realizó el siguiente procedimiento: por cada palabra que apareciera en el(los) texto(s) considerado(s), se calcula un vector de dimensión 100, que hace referencia a la representación vectorial (*word embedding*) de la palabra según las reglas explicadas en las secciones 2.1.3 y 2.1.4. Finalmente, se realiza optimización del número de Gaussianas, buscando en $N_{Gauss} \in \{2, 3, 4, ..., 15, 16\}$, y se generan los modelos. El rango donde se busca optimizar el número de Gaussianas es debido a que el mínimo número de palabras por usuario (el cual es 16) será el mayor número de Gaussianas posible para la comparación entre modelos.

5. Resultados

Los siguientes resultados muestran distintas métricas de desempeño para los experimentos realizados. Tanto para los experimentos realizados de clasificación como para los exprimentos con GMMs, se tuvo en cuenta la normalización estadística para el conjunto de datos de entrenamiento y para el conjunto de prueba.

La Figura 9 muestra la nube de palabra (*Word Cloud*) de los textos escritos por los 111 usuarios que realizaron la Tarea 1 y la Figura 10 muestra los textos escritos de los 30 usuarios que realizaron la Tarea 2. *Word Cloud* se refiere a un tipo de análisis exploratorio de datos para el procesamiento de lenguaje natural, y consiste en una representación gráfica de la frecuencia y la importancia de cada palabra en un corpus, donde una mayor frecuencia de la palabra implicará un mayor tamaño [20].

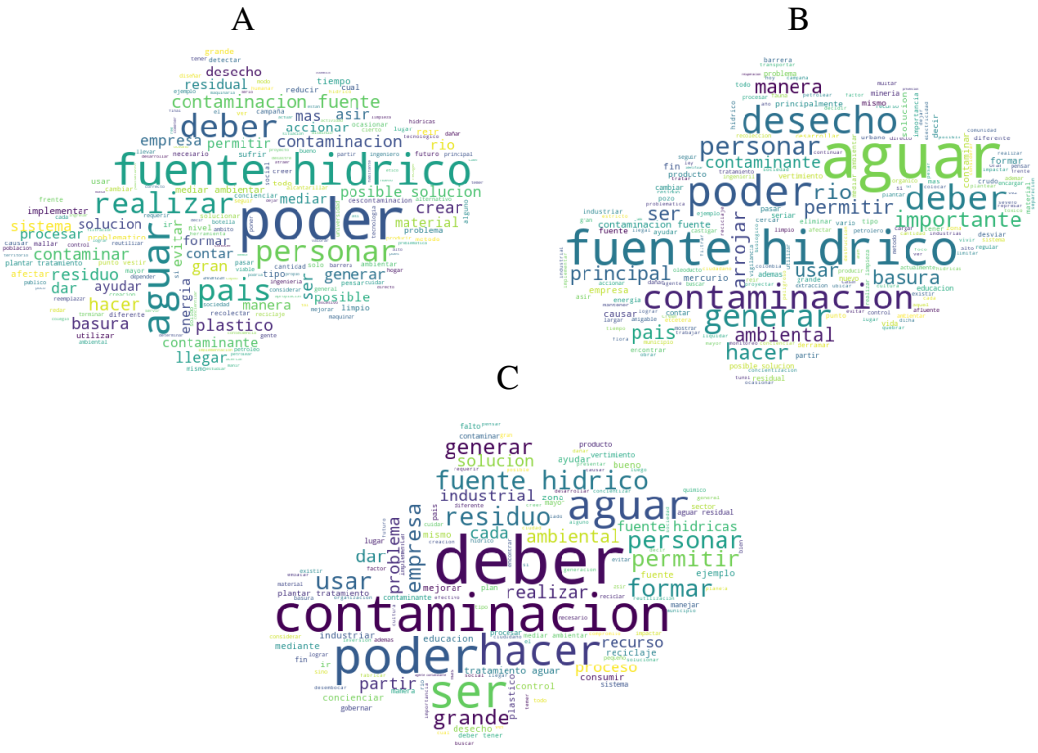

Figura 9. Nube de palabras (Word Cloud) para usuarios que realizaron la Tarea 1. A) Grupo 1, B) Grupo 2, C) Grupo 3.

Se observa en la Figura 9 y en la Figura 10, que a pesar de que los textos en cada una de ellas son del mismo tema, el tipo de palabras que utilizan con mayor frecuencia cada uno de los grupos es distinto, es decir, cada grupo tiene

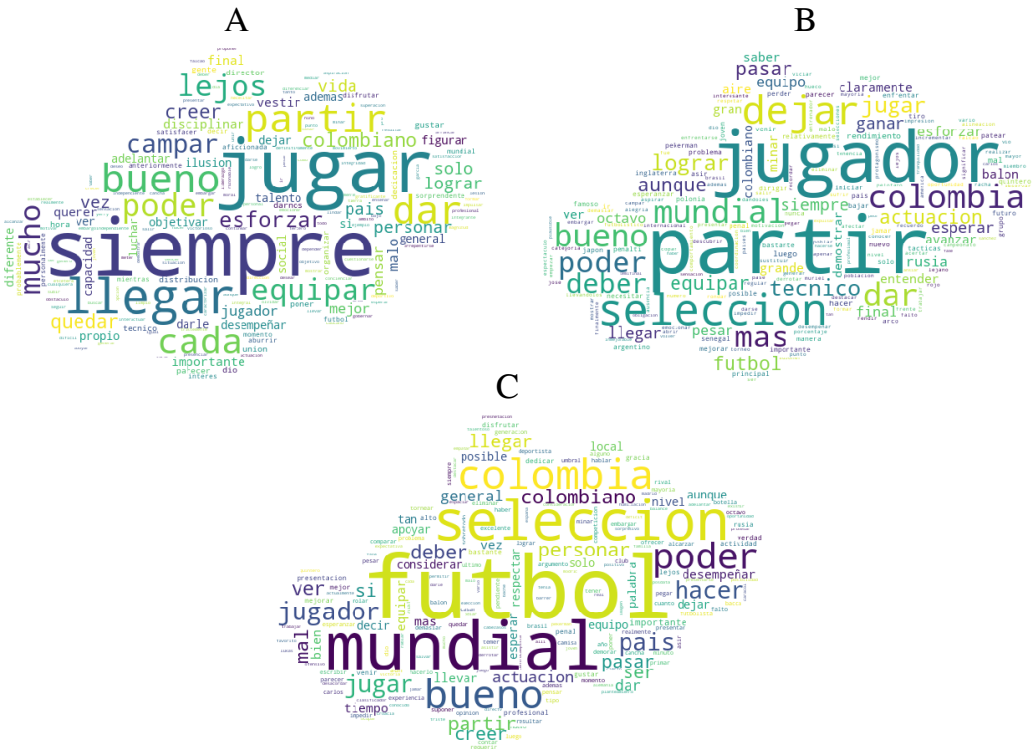

Figura 10. Nube de palabras (Word Cloud) para usuarios que realizaron la Tarea 2. A) Grupo 1, B) Grupo 2, C) Grupo 3.

sus palabras favoritas y las usan con mayor o menor frecuencia comparándolas con las palabras de los otros grupos. Esto ayuda a poder diferenciar los grupos acorde al tipo de palabras que escriben y la frecuencia con que lo hacen. Por ejemplo, el uso de la palabra "aguar" en los tres grupos, se hace notable especialmente en el Grupo 2. Debe aclararse que los usuarios no dijeron esa palabra, sino que es un error del lematizador, que transformó la palabra "agua" en "aguar" (esto también pasa con otras palabras).

De igual forma, se observa varios aspectos a considerar: las etiquetas grandes atraen más atención que las etiquetas pequeñas, esto pasa, entre otras cosas, por el número de caracteres, la posición y las etiquetas vecinas. Las etiquetas en el medio de la nube atraen más atención que las etiquetas cerca de los bordes, lo que pasa por el diseño de la nube. También, el cuadrante superior izquierdo recibe más atención que los demás, por el hábito de lectura occidental [21]. Por lo anterior, palabras como "contaminacion", "fuente", "hidrico", "aguar", "deber" sobresalen en Figura 9 y palabras como "jugar", "futbol", "seleccion", "mundial" sobresalen en Figura 10 de las demás palabras.

A parte de la anterior imagen, en la Figura 11 se muestra cómo se distribuyen los 3 Grupos considerados en este trabajo en un mismo espacio de dos dimensiones, mediante una reducción de dimensionalidad utilizando LDA. Esto se realizó considerando el grupo de características GloVe, ya que con este se lograba una mejor distinción entre clases.

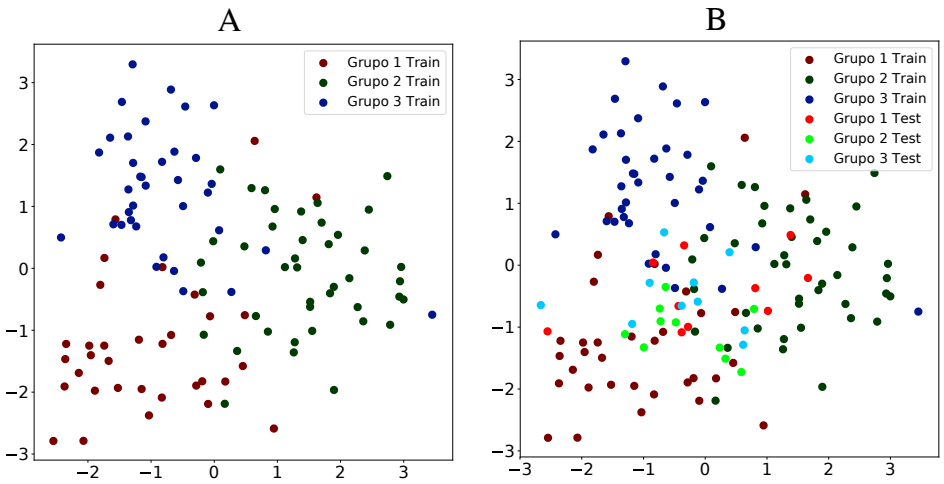

Figura 11. G1, G2 y G3 en un espacio de dos dimensiones de A) 111 usuarios que realizaron la Tarea 1 y B) 111 Usuarios de la Tarea 1 ("Train") más los 30 usuarios que realizaron la Tarea 2 ("Test").

En la Figura 11, se observa, que es posible diferenciar muy bien los 3 grupos considerando los 111 usuarios de la Tarea 1, ver (A) y que, desafortunadamente, los 30 usuarios de la Tarea 2 se alejan a su respectivo grupo (G1, G2 o G3), ver (B).

Nota: En las siguientes secciones respectivas a resultados en las cuales se entrena y se prueba con los 111 usuarios de la Tarea 1, algunos de los valores de las matrices de confusión se pueden repetir con valores de métricas (Eficiencia, F1-score, sensibilidad y especificidad) diferentes, esto pasa debido a que las matrices de confusión mostradas se sacaron a partir de valores de eficiencia cercanos al promedio y dentro de la desviación estándar para cada uno de los casos, y no de valores exactos, ya que el experimento se repitió 10 veces (lo explicado en la sección 4.1.1).

5.1. Resultados SVM Biclase (G1 vs G3)

La Tabla 4 y la Tabla 5 muestran los resultados para la clasificación de los usuarios del Grupo 1 vs los usuarios del Grupo 3 entrenando y probando con los 111 usuarios que realizaron la Tarea 1. Se puede observar que, el mejor

resultado se obtiene usando la característica GloVe, tanto para el vector de características original como para el vector de características después de realizar LDA, con un valor de eficiencia al clasificar de hasta 65.53 %.

Tabla 4. Clasificación mediante SVM de los textos del G1 vs textos del G3 entrenando y probando con los 111 usuarios que realizaron la Tarea 1. **Caract**: Caracteristica(s) implementada(s), **K**: Kernel, **Efi**: Eficiencia en el conjunto de prueba, **F1**: F1-score, **Sen**: Sensibilidad, **Esp**: Especificidad, **Mat**: Matriz de confusión.

Caract	**K**	**C**	γ	**Efi (%)**	**F1 (%)**	**Sen (%)**	**Esp (%)**	**Mat**
Fusión	Rbf	0.5	0.0001	64.1 ± 3.0	59.6 ± 3.9	75.9 ± 3.8	49.3 ± 8.5	[18 15] [12 30]
BoW	Rbf	0.5	1	63.9 ± 3.2	58.6 ± 3.6	77.7 ± 7.9	46.3 ± 5.8	[17 16] [9 33]
TF-IDF	Rbf	0.05	1	62.6 ± 4.2	57.6 ± 5.1	73.7 ± 6.4	48.8 ± 5.3	[18 15] [13 29]
Word2vec	Rbf	0.001	1	58.7 ± 2.1	48.5 ± 3.7	81.6 ± 7.3	28.4 ± 12.9	[7 26] [5 37]
GloVe	Rbf	10	1	**66.8 ± 2.5**	64.2 ± 3.4	78.2 ± 4.2	52.7 ± 6.3	[18 15] [9 33]
Gramaticales	Linear	0.001	-	58.6 ± 1.3	50.8 ± 1.8	73.9 ± 5.8	39.8 ± 8.7	[14 19] [14 28]

Con estos resultados presentes, se procedió a realizar el experimento de entrenar la SVM con los 111 usuarios de la Tarea 1 y probando con los 30 usuarios que realizaron la Tarea 2, utilizando los parámetros C y γ que dieron los mejores resultados anteriormente descritos. La Tabla 6 y la Tabla 7 muestran las métricas obtenidas, donde, en la mayoría de los casos, se obtienen resultados regulares, y que, el clasificador se ajusta para distinguir una de las dos clases.

El mejor resultado, para este experimento de clasificación biclase haciendo uso de SVM es de una eficiencia en la clasificación de 75 %, el cual se obtiene usando la característica GloVe mediante el entrenamiento con los textos de la Tarea 1 y la prueba con los textos de la Tarea 2, haciendo uso de LDA. Esto indica, que para realizar una distinción entre usuarios del Grupo 1 y usuarios del Grupo 3, es útil tener en cuenta GloVe, realizar el entrenamiento y la prueba con tareas distintas y realizar reducción de dimensionalidad a 2 mediante LDA.

5.2. Resultados SVM Triclase (G1 vs G2 vs G3)

En esta sección, se muestran los resultados de clasificación triclase haciendo uso de la SVM. Los resultados de la validación cruzada están en la Tabla 8 y Tabla 9. De nuevo, se observa que los mejores resultados corresponden a

Tabla 5. Clasificación mediante SVM de los textos del G1 vs textos del G3 entrenando y probando con los 111 usuarios que realizaron la Tarea 1, aplicando LDA. **Caract**: Caracteristica(s) implementada(s), **K**: Kernel, **Efi**: Eficiencia en el conjunto de prueba, **F1**: F1-score, **Sen**: Sensibilidad, **Esp**: Especificidad, **Mat**: Matriz de confusión.

Caract	**K**	**C**	γ	**Efi (%)**	**F1 (%)**	**Sen (%)**	**Esp (%)**	**Mat**
Fusión	Rbf	0.5	0.0001	61.2 ± 1.9	52.9 ± 2.8	84.6 ± 3.7	31.0 ± 6.6	[18 15] [12 30]
BoW	Rbf	0.05	0.0001	61.4 ± 2.6	53.3 ± 4.1	83.5 ± 3.5	33.6 ± 7.1	[17 16] [9 33]
TF-IDF	Rbf	0.5	0.0001	59.2 ± 1.9	46.6 ± 3.6	93.7 ± 5.2	15.0 ± 9.4	[18 15] [13 29]
Word2vec	Linear	0.05	-	63.6 ± 2.6	61.4 ± 3.5	64.4 ± 6.5	62.6 ± 6.3	[7 26] [5 37]
GloVe	Linear	0.05	-	**65.5 ± 3.0**	63.4 ± 3.6	73.2 ± 4.1	55.8 ± 8.1	[18 15] [9 33]
Gramaticales	Rbf	0.5	0.0001	62.8 ± 2.5	59.4 ± 2.7	64.9 ± 6.6	60.1 ± 7.3	[14 19] [14 28]

Tabla 6. Clasificación mediante SVM de los textos del G1 vs textos del G3, entrenando con los 111 usuarios de la Tarea 1 y probando con los 30 usuarios de la Tarea 2. **Caract**: Caracteristica(s) implementada(s), **K**: Kernel, **Efi**: Eficiencia en el conjunto de prueba, **F1**: F1-score, **Sen**: Sensibilidad, **Esp**: Especificidad, **Mat**: Matriz de confusión.

Caract	**K**	**C**	γ	**Efi (%)**	**F1 (%)**	**Sen (%)**	**Esp (%)**	**Mat**
Fusión	Rbf	0.5	0.0001	50.0	33.3	100.0	0.0	[0 10] [0 10]
BoW	Rbf	0.5	1	50.0	33.3	0.0	100.0	[10 0] [10 0]
TF-IDF	Rbf	0.05	1	55.0	52.0	80.0	30.0	[3 7] [2 8]
Word2vec	Rbf	0.001	1	50.0	33.3	100.0	0.0	[0 10] [0 10]
GloVe	Rbf	10	1	**55.0**	54.9	50.0	60.0	[6 4] [5 5]
Gramaticales	Linear	0.001	-	50.0	33.3	100.0	0.0	[0 10] [0 10]

Tabla 7. Clasificación mediante SVM de los textos del G1 vs textos del G3, entrenando con los 111 usuarios de la Tarea 1 y probando con los 30 usuarios de la Tarea 2, aplicando LDA. **Caract**: Caracteristica(s) implementada(s), **K**: Kernel, **Efi**: Eficiencia en el conjunto de prueba, **F1**: F1-score, **Sen**: Sensibilidad, **Esp**: Especificidad, **Mat**: Matriz de confusión.

Caract	**K**	**C**	γ	**Efi (%)**	**F1 (%)**	**Sen (%)**	**Esp (%)**	**Mat**
Fusión	Rbf	0.5	0.0001	50.0	33.3	100.0	0.0	[0 10] [0 10]
BoW	Rbf	0.05	0.0001	50.0	49.5	40.0	60.0	[6 4] [6 4]
TF-IDF	Rbf	0.5	0.0001	65.0	62.7	90.0	40.0	[4 6] [1 9]
Word2vec	Linear	0.05	-	60.0	59.6	70.0	50.0	[5 5] [3 7]
GloVe	Linear	0.05	-	**75.0**	74.4	90.0	60.0	[6 4] [1 9]
Gramaticales	Rbf	0.5	0.0001	50.0	33.3	100.0	0.0	[0 10] [0 10]

los obtenidos mediante la característica GloVe, y que, desafortunadamente, no son muy altos, siendo el mejor de un porcentaje de eficiencia de 43.74 %.

En busca de mejorar los resultados anteriores, se realiza el entrenamiento de la SVM con los textos de la Tarea 1 y la prueba con los textos de la Tarea 2, haciendo uso de los parámetros C y γ entrenados. Los resultados se observan en la Tabla 10 y Tabla 11. Como se ve, no se logra una mejor eficiencia y los valores son muy similares. Note que los mejores resultados, tanto con el vector original de características como con el vector después de aplicar LDA, no se obtienen con la característica GloVe, sino que son obtenidos con BoW. Para algunos experimentos, el clasificador se sesga por una de las 3 clases (3 Grupos) y no logra distinguir entre ellos.

Para este caso de clasificación triclase, el mejor resultado que se logra es de 43.74 % con GloVe, lo que nos da una idea, de que no es posible distinguir con una alta eficiencia entre usuarios. También se resalta que el Coeficiente de Cohen's Kappa es muy bajo en la mayoría de los casos, siendo su mayor valor de 0.150, lo que indica, que, no hay acuerdo entre evaluadores o, a lo mucho, es muy leve, debido a que la probabilidad de asignación de etiquetas iguales por parte de los dos evaluadores es muy baja.

5.3. Resultados RF Biclase (G1 vs G3)

Similar a lo realizado con la SVM, se procede a realizar una clasificación biclase con RF. La Tabla 12 y la Tabla 13 muestran los resultados para la

Tabla 8. Clasificación mediante SVM de los textos del G1 vs textos del G2 vs textos del G3, entrenando y probando con los 111 usuarios que realizaron la Tarea 1. **Caract**: Caracteristica(s) implementada(s), **K**: Kernel, **Efi**: Eficiencia en el conjunto de prueba, **F1**: F1-score, κ: Coeficiente Kappa de Cohen, **Mat**: Matriz de confusión.

Caract	**K**	**C**	γ	**Efi** (%)	**F1** (%)	κ	**Mat**
Fusión	Rbf	0.001	0.0001	41.3 ± 2.8	38.4 ± 3.5	0.105 ± 0.044	[10 15 8] [7 24 11] [10 11 15]
BoW	Rbf	5	0.0001	39.9 ± 1.9	35.8 ± 1.9	0.077 ± 0.025	[7 18 8] [3 31 8] [8 18 10]
TF-IDF	Rbf	0.005	0.0001	39.9 ± 2.9	36.9 ± 2.9	0.081 ± 0.043	[7 16 10] [6 25 11] [8 15 13]
Word2vec	Rbf	10	0.0001	40.9 ± 2.8	36.9 ± 2.1	0.092 ± 0.041	[5 24 4] [5 31 6] [5 21 10]
GloVe	Rbf	10	0.0001	**43.7 ± 1.9**	41.1 ± 1.5	0.148 ± 0.025	[12 14 7] [6 22 14] [8 12 16]
Gramaticales	Linear	0.05	-	34.2 ± 1.3	27.2 ± 1.7	-0.026 ± 0.020	[3 24 6] [1 31 10] [4 27 5]

Tabla 9. Clasificación mediante SVM de los textos del G1 vs textos del G2 vs textos del G3, entrenando y probando con los 111 usuarios que realizaron la Tarea 1, aplicando LDA. **Caract**: Caracteristica(s) implementada(s), **K**: Kernel, **Efi**: Eficiencia en el conjunto de prueba, **F1**: F1-score, κ: Coeficiente Kappa de Cohen, **Mat**: Matriz de confusión.

Caract	**K**	**C**	γ	**Efi** (%)	**F1** (%)	κ	**Mat**
Fusión	Rbf	0.5	0.1	38.8 ± 1.2	30.9 ± 1.7	0.037 ± 0.020	[7 22 4] [2 32 8] [3 28 5]
BoW	Rbf	0.5	0.1	35.9 ± 1.9	27.2 ± 1.8	-0.009 ± 0.033	[3 23 7] [3 34 5] [2 30 4]
TF-IDF	Rbf	1	0.1	35.9 ± 2.7	23.7 ± 2.1	-0.015 ± 0.039	[4 29 0] [3 35 4] [1 33 2]
Word2vec	Rbf	0.5	0.0001	35.4 ± 3.7	33.1 ± 3.6	0.024 ± 0.057	[10 12 11] [9 17 16] [8 12 16]
GloVe	Rbf	0.05	0.0001	**43.2 ± 1.9**	40.9 ± 2.0	0.135 ± 0.029	[12 15 6] [10 23 9] [7 15 14]
Gramaticales	Rbf	5	0.0001	35.3 ± 4.3	31.9 ± 4.5	0.012 ± 0.066	[4 17 12] [5 24 13] [3 19 14]

Tabla 10. Clasificación mediante SVM de los textos del G1 vs textos del G2 vs textos del G3, entrenando con los 111 usuarios de la Tarea 1 y probando con los 30 usuarios de la Tarea 2. **Caract**: Caracteristica(s) implementada(s), **K**: Kernel, **Efi**: Eficiencia en el conjunto de prueba, **F1**: F1-score, κ: Coeficiente Kappa de Cohen, **Mat**: Matriz de confusión.

Caract	**K**	**C**	γ	**Efi (%)**	**F1 (%)**	κ	**Mat**
Fusión	Rbf	0.05	0.0001	36.7	27.9	0.050	[9 0 1] [7 1 2] [9 0 1]
BoW	Rbf	0.005	0.0001	**43.3**	39.6	0.149	[7 3 0] [5 5 0] [9 0 1]
TF-IDF	Rbf	0.005	0.0001	36.7	35.1	0.050	[6 3 1] [6 3 1] [7 1 2]
Word2vec	Rbf	10	0.0001	26.7	19.7	-0.100	[0 9 1] [3 7 0] [2 7 1]
GloVe	Rbf	10	0.0001	26.7	27.5	-0.100	[3 5 2] [7 2 1] [4 3 3]
Gramaticales	Rbf	10	0.0001	33.3	16.7	0.000	[0 10 0] [0 10 0] [0 10 0]

clasificación de los usuarios del Grupo 1 vs los usuarios del Grupo 3 entrenando y probando con los 111 usuarios que realizaron la Tarea 1. El mejor resultado en este caso, se obtiene mediante la reducción de dimensión con LDA al vector de características de GloVe, obteniendo una eficiencia en la clasificación de 66.63 %. Se procede a hacer el entrenamiento con la Tarea 1 y la prueba con la Tarea 2, usando los parámetros $n-estimators$ y $max-depth$ entrenados. Los resultados son los mostrados en Tabla 14 y Tabla 15, donde se observa que, primero, no hubo mejora considerable a los resultados obtenidos anteriores y segundo, que son menores a los obtenidos con SVM, lo que indica que este tipo de entrenamiento y prueba no es adecuado usando el clasificador RF, ya que el mejor resultado de eficiencia es de 65 %.

En esta clasificación biclase con RF, el mejor resultado es de una eficiencia de 66.63 %, con la característica GloVe mediante el entrenamiento y la prueba con los textos de la Tarea 1, haciendo uso de LDA. Se observa en algunos resultados, que el clasificador correspondiente es bueno para distinguir el Grupo 1, pero descuida el hecho de poder diferenciar los usuarios del Grupo 3 de los usuarios del Grupo 1, ya que la mayoría de usuarios del Grupo 3 los toma como si fuesen del Grupo 1 (Ver Tabla 12, Tabla 13, Tabla 14 y Tabla 15).

Tabla 11. Clasificación mediante SVM de los textos del G1 vs textos del G2 vs textos del G3, entrenando con los 111 usuarios de la Tarea 1 y probando con los 30 usuarios de la Tarea 2, aplicando LDA. **Caract**: Caracteristica(s) implementada(s), **K**: Kernel, **Efi**: Eficiencia en el conjunto de prueba, **F1**: F1-score, κ: Coeficiente Kappa de Cohen, **Mat**: Matriz de confusión.

Caract	**K**	**C**	γ	**Efi (%)**	**F1 (%)**	κ	**Mat**
Fusión	Rbf	0.5	0.1	33.3	16.7	0.000	[0 10 0] [0 10 0] [0 10 0]
BoW	Rbf	50	0.1	40.0	38.3	0.099	[4 1 5] [3 2 5] [4 0 6]
TF-IDF	Rbf	1	0.1	43.3	38.3	0.150	[4 2 4] [2 1 7] [2 0 8]
Word2vec	Rbf	0.5	0.0001	**43.3**	41.2	0.150	[4 6 0] [2 7 1] [2 6 2]
GloVe	Rbf	0.05	0.0001	23.3	21.7	-0.149	[4 4 2] [8 2 0] [5 4 1]
Gramaticales	Rbf	1	0.0001	30.0	24.0	-0.050	[1 8 1] [2 7 1] [1 8 1]

Tabla 12. Clasificación mediante RF de los textos del G1 vs textos del G3 entrenando y probando con los 111 usuarios que realizaron la Tarea 1. **Caract**: Caracteristica(s) implementada(s), N_t: Número de árboles, M_d: Máxima profundidad de los árboles, **Efi**: Eficiencia en el conjunto de prueba, **F1**: F1-score, **Sen**: Sensibilidad, **Esp**: Especificidad, **Mat**: Matriz de confusión.

Caract	N_t	M_d	**Efi (%)**	**F1 (%)**	**Sen (%)**	**Esp (%)**	**Mat**
Fusión	5	2	62.9 ± 3.9	58.8 ± 4.6	68.9 ± 5.9	55.8 ± 7.8	[17 16] [9 33]
BoW	5	10	61.4 ± 1.9	53.5 ± 3.4	79.9 ± 9.3	37.7 ± 12.6	[9 24] [4 38]
TF-IDF	20	10	63.3 ± 3.0	56.6 ± 4.5	76.4 ± 5.8	46.9 ± 10.3	[14 19] [8 34]
Word2vec	15	1	64.4 ± 3.3	61.1 ± 3.4	70.7 ± 6.9	56.3 ± 8.2	[16 17] [8 34]
GloVe	10	1	**64.8 ± 4.3**	61.7 ± 4.9	66.8 ± 7.7	62.6 ± 11.1	[26 7] [17 25]
Gramaticales	5	1	62.7 ± 2.5	58.8 ± 3.3	68.5 ± 3.5	55.3 ± 5.5	[20 13] [13 29]

Tabla 13. Clasificación mediante RF de los textos del G1 vs textos del G3 entrenando y probando con los 111 usuarios que realizaron la Tarea 1, aplicando LDA. **Caract**: Caracteristica(s) implementada(s), N_t: Número de árboles, M_d: Máxima profundidad de los árboles, **Efi**: Eficiencia en el conjunto de prueba, **F1**: F1-score, **Sen**: Sensibilidad, **Esp**: Especificidad, **Mat**: Matriz de confusión.

Caract	N_t	M_d	**Efi (%)**	**F1 (%)**	**Sen (%)**	**Esp (%)**	**Mat**
Fusión	10	1	62.4 ± 2.3	56.3 ± 3.2	76.3 ± 7.9	44.8 ± 10.1	[17 16] [10 32]
BoW	5	1	63.4 ± 2.5	57.6 ± 2.8	77.0 ± 7.9	45.9 ± 8.4	[21 12] [14 28]
TF-IDF	10	1	58.9 ± 2.1	46.9 ± 3.0	92.3 ± 4.8	16.7 ± 9.1	[8 25] [5 37]
Word2vec	5	1	64.2 ± 3.4	61.6 ± 3.9	66.1 ± 5.1	61.9 ± 5.9	[19 14] [12 30]
GloVe	5	1	**66.6 ± 2.1**	64.4 ± 2.3	71.2 ± 6.9	60.8 ± 8.7	[22 11] [14 28]
Gramaticales	5	1	63.3 ± 1.5	60.3 ± 1.9	65.8 ± 3.2	60.0 ± 5.1	[21 12] [15 27]

Tabla 14. Clasificación mediante RF de los textos del G1 vs textos del G3, entrenando con los 111 usuarios de la Tarea 1 y probando con los 30 usuarios de la Tarea 2. **Caract**: Caracteristica(s) implementada(s), **K**: Kernel, **Efi**: Eficiencia en el conjunto de prueba, **F1**: F1-score, **Sen**: Sensibilidad, **Esp**: Especificidad, **Mat**: Matriz de confusión.

Caract	N_t	M_d	**Efi (%)**	**F1 (%)**	**Sen (%)**	**Esp (%)**	**Mat**
Fusión	5	2	**65.0**	60.1	100.0	30.0	[3 7] [0 10]
BoW	5	10	55.0	48.7	20.0	90.0	[9 1] [8 2]
TF-IDF	20	10	60.0	56.0	30.0	90.0	[9 1] [7 3]
Word2vec	15	1	60.0	60.0	60.0	60.0	[6 4] [4 6]
GloVe	10	1	55.0	53.9	70.0	40.0	[4 6] [3 7]
Gramaticales	5	1	60.0	52.4	100.0	20.0	[2 8] [0 10]

Tabla 15. Clasificación mediante RF de los textos del G1 vs textos del G3, entrenando con los 111 usuarios de la Tarea 1 y probando con los 30 usuarios de la Tarea 2, aplicando LDA. **Caract**: Caracteristica(s) implementada(s), **K**: Kernel, **Efi**: Eficiencia en el conjunto de prueba, **F1**: F1-score, **Sen**: Sensibilidad, **Esp**: Especificidad, **Mat**: Matriz de confusión.

Caract	N_t	M_d	**Efi (%)**	**F1 (%)**	**Sen (%)**	**Esp (%)**	**Mat**
Fusión	10	1	50.0	33.3	0.0	100.0	[10 0] [10 0]
BoW	5	1	55.0	53.9	40.0	70.0	[7 3] [6 4]
TF-IDF	10	1	60.0	56.0	30.0	90.0	[9 1] [7 3]
Word2vec	5	1	**65.0**	64.9	70.0	60.0	[6 4] [3 7]
GloVe	5	1	65.0	60.1	100.0	30.0	[3 7] [0 10]
Gramaticales	5	1	55.0	43.6	100.0	10.0	[1 9] [0 10]

5.4. Resultados RF Triclase (G1 vs G2 vs G3)

De nuevo, similar a lo realizado con la SVM, se realiza clasificación triclase haciendo uso de RF. Los primeros resultados son los obtenidos mediante el entrenamiento y prueba con los textos de la Tarea 1 y se muestran en Tabla 16 y la Tabla 17. Con el vector original de características, el mejor resultado es con Word2vec (aproximadamente 40 % de eficiencia al clasificar), y, con el vector de características después de aplicar LDA, se obtiene el mejor resultado con GloVe (43.54 % de eficiencia).

Los resultados con los parámetros $n - estimators$ y $max - depth$ entrenados y haciendo uso de la Tarea 1 para entrenar y la Tarea 2 para probar se muestran en Tabla 18 y Tabla 19. Como se ve, se logra una pequeña mejora respecto a los experimentos anteriores, logrando subir el porcentaje de eficiencia de clasificación hasta 46.67 %, el cual es obtenido con la característica Word2vec, mediante reducción de dimensionalidad con LDA.

Con este experimento de clasificación triclase mediante RF, fue posible subir un poco el porcentaje de eficiencia del sistema de clasificación triclase, a 46.7 %.

Tabla 16. Clasificación mediante RF de los textos del G1 vs textos del G2 vs textos del G3 entrenando y probando con los 111 usuarios que realizaron la Tarea 1. **Caract**: Caracteristica(s) implementada(s), N_t: Número de árboles, M_d: Máxima profundidad de los árboles, **Efi**: Eficiencia en el conjunto de prueba, **F1**: F1-score, κ: Coeficiente Kappa de Cohen, **Sen**: Sensibilidad, **Esp**: Especificidad, **Mat**: Matriz de confusión.

Caract	N_t	M_d	**Efi** (%)	**F1** (%)	κ	**Mat**
Fusión	50	2	39.0 ± 3.7	35.1 ± 3.9	0.059 ± 0.056	[8 17 8] [6 28 8] [2 24 10]
BoW	15	10	38.9 ± 3.2	31.9 ± 3.9	0.045 ± 0.048	[5 22 6] [5 33 4] [4 26 6]
TF-IDF	50	10	38.6 ± 3.0	32.6 ± 3.3	0.046 ± 0.045	[6 15 12] [6 29 7] [2 24 10]
Word2vec	50	2	**39.9 ± 4.5**	36.9 ± 4.7	0.080 ± 0.068	[2 19 12] [3 30 9] [4 17 15]
GloVe	50	5	39.2 ± 2.9	36.2 ± 3.1	0.067 ± 0.047	[9 14 10] [5 24 13] [10 14 12]
Gramaticales	5	1	33.8 ± 3.4	28.4 ± 3.5	-0.201 ± 0.052	[1 22 10] [3 25 14] [3 20 13]

Tabla 17. Clasificación mediante RF de los textos del G1 vs textos del G2 vs textos del G3 entrenando y probando con los 111 usuarios que realizaron la Tarea 1, aplicando LDA. **Caract**: Caracteristica(s) implementada(s), N_t: Número de árboles, M_d: Máxima profundidad de los árboles, **Efi**: Eficiencia en el conjunto de prueba, **F1**: F1-score, κ: Coeficiente Kappa de Cohen, **Sen**: Sensibilidad, **Esp**: Especificidad, **Mat**: Matriz de confusión.

Caract	N_t	M_d	**Efi** (%)	**F1** (%)	κ	**Mat**
Fusión	50	2	37.9 ±1.8	31.4 ± 2.1	0.036 ± 0.028	[5 23 5] [7 28 7] [7 19 10]
BoW	20	5	33.8 ± 2.6	27.3 ± 2.7	-0.029 ± 0.041	[3 18 12] [3 28 11] [6 23 7]
TF-IDF	15	5	33.9 ± 2.6	25.0 ± 2.5	-0.033 ± 0.036	[0 25 8] [1 29 12] [2 24 10]
Word2vec	15	2	33.7 ± 2.8	31.2 ± 3.5	0.003 ± 0.042	[11 12 10] [14 13 15] [9 13 14]
GloVe	50	2	**43.5 ± 3.6**	40.8 ± 3.5	0.139 ± 0.054	[10 16 7] [9 26 7] [6 16 14]
Gramaticales	15	1	34.8 ± 3.0	30.8 ± 3.2	0.006 ± 0.046	[4 13 16] [8 21 13] [4 17 15]

Tabla 18. Clasificación mediante RF de los textos del G1 vs textos del G2 vs textos del G3, entrenando con los 111 usuarios de la Tarea 1 y probando con los 30 usuarios de la Tarea 2. **Caract**: Caracteristica(s) implementada(s), N_t: Número de árboles, M_d: Máxima profundidad de los árboles, **Efi**: Eficiencia en el conjunto de prueba, **F1**: F1-score, κ: Coeficiente Kappa de Cohen, **Sen**: Sensibilidad, **Esp**: Especificidad, **Mat**: Matriz de confusión.

Caract	N_t	M_d	**Efi (%)**	**F1 (%)**	κ	**Mat**
Fusión	50	2	**40.0**	28.2	0.099	[0 9 1] [0 10 0] [0 8 2]
BoW	15	10	16.7	14.8	-0.250	[0 9 1] [6 4 0] [3 6 1]
TF-IDF	50	10	40.0	37.8	0.099	[7 3 0] [7 3 0] [6 2 2]
Word2vec	50	2	26.7	25.2	-0.100	[2 4 4] [1 5 4] [2 7 1]
GloVe	50	5	20.0	19.1	-0.200	[1 8 1] [6 4 0] [6 3 1]
Gramaticales	5	1	30.0	23.5	-0.050	[3 0 7] [6 0 4] [4 0 6]

Tabla 19. Clasificación mediante RF de los textos del G1 vs textos del G2 vs textos del G3, entrenando con los 111 usuarios de la Tarea 1 y probando con los 30 usuarios de la Tarea 2, aplicando LDA. **Caract**: Caracteristica(s) implementada(s), N_t: Número de árboles, M_d: Máxima profundidad de los árboles, **Efi**: Eficiencia en el conjunto de prueba, **F1**: F1-score, κ: Coeficiente Kappa de Cohen, **Sen**: Sensibilidad, **Esp**: Especificidad, **Mat**: Matriz de confusión.

Caract	N_t	M_d	**Efi** (%)	**F1** (%)	κ	**Mat**
Fusión	50	2	36.7	26.0	0.050	[2 0 8] [4 0 6] [1 0 9]
BoW	20	5	33.3	25.9	0.00	[1 0 9] [0 1 9] [1 1 8]
TF-IDF	15	5	30.0	27.6	-0.050	[5 3 2] [5 1 4] [7 0 3]
Word2vec	15	2	**46.7**	46.0	0.199	[6 4 0] [0 5 5] [5 2 3]
GloVe	50	2	23.3	22.7	-0.149	[4 4 2] [8 1 1] [4 4 2]
Gramaticales	15	1	40.00	40.9	0.099	[3 1 6] [5 4 1] [4 1 5]

5.5. Resultados GMM

A parte de los anteriores resultados, se realizan dos pruebas: clasificación biclase (G1 vs G3) y clasificación triclase (G1 vs G2 vs G3) haciendo uso de GMM. Para el caso de la clasificación biclase, en el entrenamiento, se toman 75 de los 111 usuarios que realizaron la Tarea 1 (42 usuarios del G1 y 33 del G3) y se generan 2 modelos, uno para cada grupo. Después, para probar, se toman 20 de los 30 usuarios que hicieron la Tarea 2 (10 del G1 y 10 del G3) y se generan 20 modelos, para poder observar a qué grupo pertenece (G1 o G3) según el algoritmo, considerando la menor distancia de Bhattacharyya. Los resultados se muestran en la Tabla 20 y Tabla 21, donde se resalta el mejor valor de eficiencia para distinguir entre grupos y se puede observar que los valores no son superiores a los obtenidos con SVM y RF, de hecho, el mejor valor de eficiencia obtenido en este caso es de 75 % y es muy similar al mejor resultado de la Tabla 7.

Para el caso de clasificación triclase, se generan modelos para cada uno de los 3 Grupos de usuarios que realizaron la Tarea 1, y así, tener 3 modelos entrenados. Después, por cada usuario que realizó la Tarea 2, se genera un modelo de prueba para medir la distancia de Bhattacharyya de estos 30 modelos a los 3 modelos entrenados, y así, poder observar a qué grupo pertenece, de nuevo, considerando la menor distancia. En la Tabla 22 y Tabla 23 se muestran las métricas obtenidas, donde también se subraya el mejor valor. Como podemos ver, el mayor valor de eficiencia obtenido es de 53.3 % considerando la característica GloVe.

Observando los resultados de la Tabla 20, Tabla 21, Tabla 22 y Tabla 23, se puede decir que es posible distinguir a los usuarios de los Grupos 1 y 3 con una precisión aceptable (53.3 %) de los usuarios del Grupo 2, y que no es posible distinguir los usuarios del Grupo 2, ya que los modelos GMM considerados aquí los confunden con usuarios pertenecientes al Grupo 1 o 3. También, los resultados con GMM muestran que la característica GloVe arroja los mejores resultados, mejorando hasta un valor de 53.3 % de eficiencia en la caracterización triclase, que es el mejor valor obtenido en este trabajo e igualando el valor de 75 % de eficiencia (obtenido con SVM) en la clasificación biclase.

Por los resultados, es notable que este tipo de modelos discriminan bien los usuarios de niveles de escolaridad intermedios y superiores y logran diferenciarlos de los usuarios de niveles bajos, por lo cuál es útil utilizar este método si el objetivo principal es discriminar a usuarios de niveles intermedios en adelante.

Tabla 20. Clasificación mediante GMM y el grupo de características Word2vec de los textos del G1 vs textos del G3, entrenando con 75 usuarios de la Tarea 1 y probando con 20 usuarios de la Tarea 2. N_{Gauss}: Número de Gaussianas, **Efi**: Eficiencia en el conjunto de prueba, **F1**: F1-score, **Sen**: Sensibilidad, **Esp**: Especificidad, **Mat**: Matriz de confusión.

N_{Gauss}	**Efi (%)**	**F1 (%)**	**Sen (%)**	**Esp (%)**	**Mat**
2	55.0	53.9	40.0	70.0	[7 3] [6 4]
3	55.0	54.9	60.0	50.0	[5 5] [4 6]
4	**65.0**	60.1	30.0	100.0	[10 0] [7 3]
5	55.0	54.9	50.0	60.0	[6 4] [5 5]
6	60.0	59.6	50.0	70.0	[7 3] [5 5]
7	60.0	58.3	40.0	80.0	[8 2] [6 4]
8	50.0	33.3	0.0	100.0	[10 0] [0 10]
9	50.0	45.1	80.0	20.0	[2 8] [2 8]
10	50.0	40.5	90.0	10.0	[1 9] [1 9]
11	35.0	33.5	20.0	50.0	[5 5] [8 2]
12	40.0	37.5	20.0	60.0	[6 4] [8 2]
13	55.0	48.7	20.0	90.0	[9 1] [8 2]
14	55.0	48.7	20.0	90.0	[9 1] [8 2]
15	60.0	52.4	20.0	100.0	[10 0] [8 2]
16	60.0	52.4	20.0	100.0	[10 0] [8 2]

Tabla 21. Clasificación mediante GMM y el grupo de características GloVe de los textos del G1 vs textos del G3, entrenando con 75 usuarios de la Tarea 1 y probando con 20 usuarios de la Tarea 2. N_{Gauss}: Número de Gaussianas, **Efi**: Eficiencia en el conjunto de prueba, **F1**: F1-score, **Sen**: Sensibilidad, **Esp**: Especificidad, **Mat**: Matriz de confusión.

N_{gauss}	**Efi (%)**	**F1 (%)**	**Sen (%)**	**Esp (%)**	**Mat**
2	50.0	33.3	0.0	100.0	[10 0] [10 0]
3	55.0	53.9	40.0	70.0	[7 3] [6 4]
4	50.0	45.1	80.0	20.0	[2 8] [2 8]
5	**75.0**	74.9	70.0	80.0	[8 2] [3 7]
6	65.0	62.7	90.0	40.0	[4 6] [1 9]
7	75.0	73.33	100.0	50.0	[5 5] [0 10]
8	55.0	53.9	70.0	40.0	[4 6] [3 7]
9	50.0	33.3	100.0	0.0	[0 10] [0 10]
10	50.0	33.3	100.0	0.0	[0 10] [0 10]
11	50.0	33.3	100.0	0.0	[0 10] [0 10]
12	50.0	33.3	100.0	0.0	[0 10] [0 10]
13	50.0	33.3	100.0	0.0	[0 10] [0 10]
14	50.0	33.3	100.0	0.0	[0 10] [0 10]
15	45.0	31.0	90.0	0.0	[0 10] [9 1]
16	45.0	31.0	90.0	0.0	[0 10] [1 9]

Tabla 22. Clasificación mediante GMM y el grupo de características Word2vec de los textos del G1 vs textos del G2 vs textos del G3, entrenando con los 111 usuarios de la Tarea 1 y probando con los 30 usuarios de la Tarea 2. N_{Gauss}: Número de Gaussianas, **Efi**: Eficiencia en el conjunto de prueba, **F1**: F1-score, κ: Coeficiente Kappa de Cohen, **Mat**: Matriz de confusión.

N_{Gauss}	**Efi (%)**	**F1 (%)**	κ	**Mat**
2	30.0	27.6	-0.050	[1 6 3] [2 5 3] [1 6 3]
3	33.3	31.1	0.000	[2 4 4] [2 2 6] [3 1 6]
4	**46.7**	42.3	0.200	[9 1 0] [6 2 2] [7 0 3]
5	30.0	29.4	-0.050	[3 6 1] [4 1 5] [3 2 5]
6	30.0	30.6	-0.050	[2 7 1] [5 2 3] [3 2 5]
7	30.0	25.9	-0.050	[5 3 2] [7 0 3] [6 0 4]
8	30.0	15.8	-0.050	[9 1 0] [10 0 0] [9 1 0]
9	33.3	31.0	0.000	[1 6 3] [2 4 4] [1 4 5]
10	33.3	21.9	0.000	[1 1 8] [2 0 8] [1 0 9]
11	20.0	20.4	-0.200	[3 5 2] [9 1 0] [5 3 2]
12	20.0	21.00	-0.200	[3 6 1] [9 1 0] [5 3 2]
13	33.3	30.1	0.000	[1 8 1] [3 7 0] [4 4 2]]
14	33.3	30.1	0.000	[1 8 1] [3 7 0] [4 4 2]]
15	23.3	21.1	-0.150	[5 5 0] [9 1 0] [4 5 1]
16	36.7	27.9	0.050	[9 1 0] [9 1 0] [7 2 1]

Tabla 23. Clasificación mediante GMM y el grupo de características GloVe de los textos del G1 vs textos del G2 vs textos del G3, entrenando con los 111 usuarios de la Tarea 1 y probando con los 30 usuarios de la Tarea 2. N_{Gauss}: Número de Gaussianas, **Efi**: Eficiencia en el conjunto de prueba, **F1**: F1-score, κ: Coeficiente Kappa de Cohen, **Mat**: Matriz de confusión.

N_{Gauss}	**Efi** (%)	**F1** (%)	κ	**Mat**
2	33.3	21.3	0.000	[9 1 0] [9 1 0] [10 0 0]
3	36.7	35.5	0.050	[5 2 3] [4 2 4] [4 2 4]
4	33.3	24.0	0.000	[2 0 8] [2 0 8] [2 0 8]
5	**53.3**	46.8	0.300	[8 0 2] [4 1 5] [3 0 7]
6	36.7	31.8	0.050	[2 4 4] [4 1 5] [1 1 8]
7	43.3	39.0	0.150	[4 2 4] [4 1 5] [0 2 8]
8	40	40.4	0.100	[3 2 5] [0 4 6] [1 4 5]
9	30.0	23.9	-0.050	[0 5 5] [0 4 6] [0 5 5]
10	30.0	23.9	-0.050	[0 5 5] [0 4 6] [0 5 5]
11	36.7	28.8	0.050	[0 5 5] [0 4 6] [0 3 7]
12	26.7	18.1	-0.100	[0 3 7] [0 1 9] [0 3 7]
13	26.7	18.1	-0.100	[0 3 7] [0 1 9] [0 3 7]
14	30.0	21.9	-0.050	[0 2 8] [0 2 8] [0 3 7]
15	23.3	16.7	-0.150	[0 2 8] [1 1 8] [1 3 6]
16	30.0	16.2	-0.050	[0 1 9] [1 0 9] [1 0 9]

6. Conclusiones

A pesar de tener las siguientes limitaciones: poca cantidad total de usuarios (no supera 150 personas), la cantidad de texto escrita por cada uno de los usuarios (ya que, en promedio, no superan las 100 palabras), se logra el objetivo principal, el cual es encontrar diferencias entre estilos de escritura de los usuarios de acuerdo a su nivel escolar, porque se alcanza un máximo de eficiencia de clasificación biclase (G1 vs G3) de 75.0 % y de 53.3 % para clasificación triclase (G1 vs G2 vs G3) . En general, los mejores resultados, se obtienen con GloVe, lo que indica, que este tipo de característica es útil cuando se quiere diferenciar entre textos por la forma en que están escritos y por su contenido. Lo anterior se sustenta en que, por ejemplo, para los experimentos de clasificación biclase con SVM y GMM, la característica GloVe permite diferenciar de una mejor manera los usuarios (realizando el entrenamiento con la Tarea 1 y las pruebas con la Tarea 2) con un porcentaje de eficiencia en la clasificación de hasta 75 % y para los experimentos de clasificación triclase, mediante GMM y con la característica GloVe, se logra una eficiencia de 53.3 %.

Respecto a los experimentos de clasificación biclase, es notable que si se entrena con los usuarios de la Tarea 1 y se prueba con la Tarea 2, el algoritmo de clasificación SVM es superior al algoritmo RF pero igual de efectivo al método implementado mediante GMM, por lo tanto, se concluye, que si se desea ditinguir entre usuarios con un nivel escolar bajo y usuarios con un nivel escolar alto, el método indicado para realizar la clasificación es considerando una SVM o GMM. Por otro lado, para la clasificación triclase, GMM es superior (53.3 % de eficiencia al clasificar) a los enfoques SVM y RF (43.3 % y 46.7 % respectivamente).

Por lo mostrado en Figura 11, y teniendo en cuenta los resultados de la clasificación triclase tanto con SVM, como con RF y con GMM, se concluye que estos tres métodos no son capaces de separar las 3 clases de una forma confiable (eficiencias superiores a 70 %) usando las características mencionadas en este trabajo, dando como mejor resultado un valor de eficiencia de 53.3 % en la clasificación con GMM, usando la característica GloVe. Por lo cual, como trabajo futuro, se propone extraer características que tengan en cuenta de fondo el estilo lingüístico de los usuarios, tales como características léxicas, sintácticas, estructurales y específicas del contenido del texto original, sin realizar ningún tipo de pre procesamiento, y, medir de nuevo, el desempeño con dichas características usando los algoritmos de clasificación trabajados aquí.

7. Referencias

[1] A. K. Jain, A. Ross, S. Prabhakar y col., "An introduction to biometric recognition", *IEEE Transactions on circuits and systems for video technology*, vol. 14, n.º 1, 2004.

[2] D. Ravelo Méndez, *¿Por qué está aumentando la educación virtual en el país?*, 2018. dirección: https://www.eltiempo.com/vida/educacion/asi-va-la-educacion-virtual-en-colombia-177598.

[3] U. de Antioquia., *Quiénes somos*, 2019. dirección: https://udearroba.udea.edu.co/.

[4] C. Borromeo Garcıa, "Entornos virtuales de aprendizaje y el plagio académico", *Revista ECE-DIGITAL. Revista de Investigación e Innovación Educativa para el Desarrollo Profesional*, vol. 7, n.º 12, págs. 79-100, 2017.

[5] R. S. Kuzu y A. A. Salah, "Chat biometrics", *IET Biometrics*, vol. 7, n.º 5, págs. 454-466, 2018.

[6] J. H. Suh, "Comparing writing style feature-based classification methods for estimating user reputations in social media", *SpringerPlus*, vol. 5, n.º 1, pág. 261, 2016.

[7] S. Argamon, C. Whitelaw, P. Chase, S. R. Hota, N. Garg y S. Levitan, "Stylistic text classification using functional lexical features", *Journal of the American Society for Information Science and Technology*, vol. 58, n.º 6, págs. 802-822, 2007.

[8] S. Mukherjee y P. K. Bala, "Gender classification of microblog text based on authorial style", *Information Systems and e-Business Management*, vol. 15, n.º 1, págs. 117-138, 2017.

[9] Praveen Dubey, *An introduction to Bag of Words and how to code it in Python for NLP*, [Online; accessed 01-April-2019], 2016. dirección: https://medium.freecodecamp.org/an-introduction-to-bag-of-words-and-how-to-code-it-in-python-for-nlp-282e87a9da04.

[10] Neeraj Singh Sarwan, *An Intuitive Understanding of Word Embeddings: From Count Vectors to Word2Vec*, [Online; accessed 01-April-2019], 2017. dirección: https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/.

[11] Claudio Bellei, *The backpropagation algorithm for Word2Vec*, [Online; accessed 03-April-2019], 2018. dirección: http://www.claudiobellei.com/2018/01/06/backprop-word2vec/.

[12] J. Pennington, R. Socher y C. Manning, "Glove: Global vectors for word representation", en *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, págs. 1532-1543.

[13] J. P. Kincaid, R. P. Fishburne Jr, R. L. Rogers y B. S. Chissom, "Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel", 1975.

[14] B. Schölkopf, A. J. Smola, F. Bach y col., *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[15] P. O. Gislason, J. A. Benediktsson y J. R. Sveinsson, "Random forests for land cover classification", *Pattern Recognition Letters*, vol. 27, n.º 4, págs. 294-300, 2006.

[16] D. Yu y L. Deng, *Automatic Speech Recognition*. Springer, 2016.

[17] S. Balakrishnama y A. Ganapathiraju, "Linear discriminant analysis-a brief tutorial", *Institute for Signal and information Processing*, vol. 18, págs. 1-8, 1998.

[18] D. M. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation", 2011.

[19] M. L. McHugh, "Interrater reliability: the kappa statistic", *Biochemia medica: Biochemia medica*, vol. 22, n.º 3, págs. 276-282, 2012.

[20] Duong Vu, *Generating WordClouds in Python*, [Online; accessed 03-April-2019], 2018. dirección: https://www.datacamp.com/community/tutorials/wordcloud-python.

[21] S. Lohmann, J. Ziegler y L. Tetzlaff, "Comparison of tag cloud layouts: Task-related performance and visual exploration", en *IFIP Conference on Human-Computer Interaction*, Springer, 2009, págs. 392-404.