UNIVERSIDAD DE CONCEPCIÓN
DIRECCIÓN DE POSTGRADO
FACULTAD DE INGENIERÍA - DOCTORADO EN CIENCIAS DE LA INGENIERÍA CON MENCIÓN EN INGENIERÍA ELÉCTRICA

Desarrollo de algoritmos para la clasificación de textos biomédicos utilizando expresiones regulares y aprendizaje activo

---

Profesor guía: Dr. Jorge Pezoa Núñez
Profesora co-guía: Dra. Rosa Figueroa Iturrieta
Departamento de Ingeniería Eléctrica
Facultad de Ingeniería
Universidad de Concepción

Tesis para optar al grado académico de Doctor en Ciencias de la Ingeniería con mención en Ingeniería Eléctrica

CHRISTOPHER ALEJANDRO FLORES JARA

CONCEPCIÓN - CHILE

2021

Hey, you're the doc, Doc

AGRADECIMIENTOS

La realización de este trabajo de tesis se debe al aporte de muchas personas y entidades. Entre todas ellas, agradezco enormemente:

El apoyo incondicional de mi familia y la motivación que me brindaron para sacar adelante este trabajo. Gracias mamá, papá, hermana y abuela desde el cielo. No fue un camino fácil, pero gracias a ustedes es que ahora estoy escribiendo estas palabras de agradecimiento.

A mis supervisores de tesis, el Dr. Jorge Pezoa y la Dra. Rosa Figueroa. Excelentes profesionales y mejores personas. Gracias por el apoyo académico en los momentos de dificultad y por los consejos personales que me ayudaron a tomar mejores decisiones estos últimos años.

A todos los "compañeros" y amigos que conocí en el laboratorio de sistemas paralelos del DIE durante mi paso por la Universidad. Gracias por hacer muy agradables los días de trabajo y por todos los momentos vividos.

Agencia Nacional de Investigación y Desarrollo (ANID), que me apoyó económicamente con la beca CONICYT-PFCHA/Doctorado Nacional 2017-21172062.

Universidad de Concepción, que me apoyó económicamente con una beca de arancel y estipendio, con el proyecto VRID-Enlace 217.092.052-1 "Developing intelligent algorithms to assist in clinical decisions and effecive delivery of health care information" y con el proyecto UCO 1866 para la realización de una pasantía de investigación en la Universidad de George Washington, Washington DC, USA. Agradezco a la directora del Centro de Informática Biomédica, la Dra. Qing Zeng-Treitler, por su apoyo para mejorar este trabajo de tesis.

Resumen

Los algoritmos de clasificación de textos, los cuales sirven de apoyo a los procesos de toma de decisiones clínicas, requieren costosos ejemplos de entrenamiento etiquetados por profesionales especializados. El aprendizaje activo (AL) busca disminuir ese costo al reducir el número de textos etiquetados que se requieren para lograr un determinado desempeño en los algoritmos de clasificación. Si bien el AL ha sido estudiado en algoritmos de clasificación lineales y probabilísticos, y recientemente, en algoritmos basados en redes neuronales profundas (DNNs), no ha sido estudiado en algoritmos de clasificación basados en expresiones regulares. Debido a ésto, el objetivo de esta tesis es desarrollar algoritmos para la clasificación de textos biomédicos utilizando expresiones regulares y AL.

Las principales contribuciones de este trabajo respecto al uso de expresiones regulares para la clasificación de textos biomédicos corresponden al desarrollo de un algoritmo denominado FREGEX (extractor de características basado en expresiones regulares) para la generación automática un espacio de características utilizando textos biomédicos en español, a un algoritmo denominado CREGEX (clasificador de textos basado en expresiones regulares) que permite clasificar textos biomédicos y a una estrategia de consulta que junto a un criterio de detención transforman a CREGEX en un algoritmo de clasificación de textos biomédicos de AL.

Los resultados indican que FREGEX generó un espacio de características representativo para CREGEX y cercano al lenguaje natural. En la mayoría de los casos, el desempeño de CREGEX fue superior a los clasificadores basados en una máquina de soporte vectorial (SVM), Naïve Bayes (NB) y en una representación de codificador bidireccional de transformadores (BERT) en términos de aciertos (ACC) y valor-F (F1), con resultados sobre el 88 % en ambas métricas de desempeño. Las curvas de aprendizaje indican que el AL redujo eficientemente el número de ejemplos de entrenamiento necesarios para obtener un mismo desempeño en términos de ACC y F1 en comparación al resto de los clasificadores. En este sentido, el criterio de detención aplicado al proceso de AL de CREGEX permitió utilizar sólo entre un 32 % a un 50 % del total de ejemplos de entrenamiento, con una diferencia de desempeño inferior al 2 % respecto del valor máximo posible de la curva de aprendizaje.

Índice general

Índice de Tablas

Índice de Figuras

Acrónimos

ACC	aciertos, del inglés *accuracy*
AL	aprendizaje activo, del inglés *Active Learning*
BERT	representación de codificador bidireccional de transformadores, del inglés *Bidirectional Encoder Representations from Transformers*
BoN	bolsa de n-gramas, del inglés *Bag of n-grams*
BoW	bolsa de palabras, del inglés *Bag of words*
CNN	red neuronal convolucional, del inglés *Convolutional Neural Network*
CREGEX	clasificador de textos basado en expresiones regulares, del inglés *Classifier based on REGEXes*
CRF	campo aleatorio condicional, del inglés *Conditional Random Fields*
DNN	red neuronal profunda, del inglés *Deep Neural Network*
ELMo	representaciones distribuidas desde modelos de lenguaje, del inglés *Embeddings from Language Models*
F1	Valor-F, del inglés *F-score*
FREGEX	extractor de características basado en expresiones regulares, del inglés *Features based on REGEXes*
GloVe	vectores globales, del inglés *Global Vectors*
HGGB	hospital Guillermo Grant Benavente
IG	ganancia de información, del inglés *Information Gain*
LSA	análisis semántico latente, del inglés *Latent Semantic Analysis*
N1	1-gramas
N2	2-gramas
NB	Bayes ingenuo, del inglés *Naïve Bayes*
NLP	procesamiento del lenguaje natural, del inglés *Natural Language Processing*
NW	Needleman-Wunsch
PL	aprendizaje pasivo, del inglés *Passive Learning*
RED	descubrimiento de expresión regular, del inglés *Regular Expression Discovery*
REDEx	extractor de descubrimiento de expresión regular, del inglés *Regular Expression Discovery Extractor*

ReLIE	aprendizaje de expresiones regulares para extracción de información, del inglés *Regex Learning for Information Extraction*
RNN	red neuronal recurrente, del inglés *Recurrent Neural Network*
SVD	descomposición en valores singulares, del inglés *Singular Value Decomposition*
SVM	máquinas de soporte vectorial, del inglés *Support Vector Machines*
SW	Smith-Waterman
TF-IDF	frecuencia de término-frecuencia inversa de término, del inglés *Term frequency – Inverse Document Frequency*
word2vec	vectores de palabras, del inglés *Word Representations in Vector Space*

Capítulo 1

Introducción

1.1. Introducción general

El progresivo desarrollo tecnológico ha permitido generar una gran cantidad de información en formatos digitales. Se estima que para el año 2025 habrá 175 *zettabytes* de información digital, gran parte de ella en forma no estructurada o texto libre [7, 8]. En el área de la salud, los registros médicos electrónicos aportan una importante fuente de información textual, razón por la cual es inminente desarrollar nuevas tecnologías que permitan descubrir automáticamente conocimiento relevante de dichas fuentes como apoyo a la toma de decisiones [9,10]. En este sentido, una de las técnicas más utilizadas para organizar automáticamente una gran cantidad de información digital es la clasificación de textos [11].

La clasificación o categorización de textos permite asignar automáticamente etiquetas predefinidas a los textos en base a su contenido [12,13]. Tradicionalmente, los algoritmos de clasificación más utilizados han sido los de tipo lineal y probabilísticos debido a la simplicidad para implementarlos y a la precisión de sus predicciones [14,15]. Aunque estos algoritmos a menudo funcionan bastante bien si son entrenados con las características correctas, aún existe espacio para mejorar [16].

En los últimos años, el uso de redes neuronales profundas (DNNs) ha revolucionado el campo del procesamiento del lenguaje natural (NLP) debido a la gran disponibilidad de datos y a las mejoras en la capacidad de procesamiento computacional [17]. Recientemente, el uso de modelos de lenguajes pre-entrenados

han aportado mejoras significativas al estado del arte de muchas tareas del NLP [18, 19]. Por otro lado, los investigadores también han explorado el uso de expresiones regulares como una alternativa a los métodos de clasificación existentes. La capacidad que tienen las expresiones regulares para representar patrones de texto y adaptarse a diferentes dominios de uso ha permitido desarrollar algoritmos con desempeños comparables a los métodos tradicionales de clasificación [20, 21]. Sin embargo, la generación automática de expresiones regulares a partir de ejemplos de entrenamiento es un problema actual de investigación [22].

Cualquiera sea el método de clasificación utilizado se requerirán textos etiquetados para el entrenamiento de los algoritmos. Sin embargo, en ciertas situaciones como en el caso de la investigación en biomedicina, etiquetar manualmente ejemplos de entrenamiento puede ser muy costoso, requiriéndose además de anotadores especializados [23]. Ante este problema surge el muestreo o aprendizaje activo (AL) que busca reducir los esfuerzos de anotación, permitiendo seleccionar progresivamente, hasta algún criterio de detención, los ejemplos considerados más informativos para que sean etiquetados [24].

El uso de AL ha sido ampliamente estudiado en clasificadores probabilísticos y lineales. Recientemente, el uso de AL ha despertado el interés de los investigadores para su aplicación en algoritmos de clasificación basados en DNNs [25, 26]. Sin embargo, previo a este trabajo, no existe una estrategia de consulta o función de selección de AL que permita identificar los ejemplos más informativos para un clasificador de textos biomédicos basado en expresiones regulares, existiendo sólo algunos trabajos relacionados a tareas de extracción de información, pero en otras áreas distintas a la biomedicina [20, 21, 27, 28].

1.2. Hipótesis y preguntas de investigación

La hipótesis de investigación de esta tesis es:

> Si se diseña una estrategia de consulta de aprendizaje activo junto con un criterio de detención que permita determinar cuáles son los ejemplos más informativos en un conjunto de datos no etiquetado, identificando los casos de ambigüedad, se mejoraría el desempeño de un clasificador de textos biomédicos basado en la generación automática de expresiones regulares, con un desempeño superior al 85 % en términos de área bajo la curva de

aprendizaje y utilizando menos del 50 % de la cantidad total de ejemplos de entrenamiento en comparación al método de aprendizaje pasivo.

Las preguntas de investigación que busca responder esta tesis son las siguientes:

(I) Para un determinado problema de clasificación de textos biomédicos, ¿Puede un algoritmo basado en expresiones regulares capturar las variantes léxicas de los términos representativos de cada clase del problema?

(II) Para un determinado problema de clasificación de textos biomédicos, ¿Puede un algoritmo basado en expresiones regulares tener un mejor desempeño que los algoritmos de clasificación más utilizados?

(III) Para un determinado algoritmo de clasificación de textos biomédicos basado en expresiones regulares, ¿Puede el aprendizaje activo reducir efectivamente la cantidad de ejemplos de entrenamiento necesarios para obtener un determinado desempeño en comparación al aprendizaje pasivo?

(IV) Para un determinado algoritmo de clasificación de textos biomédicos basado en expresiones regulares, ¿Puede el aprendizaje activo mejorar el desempeño en comparación a otras estrategias de consulta de selección de los ejemplos más informativos?

1.3. Objetivos

1.3.1. Objetivo general

Formular, diseñar e implementar un clasificador de textos biomédicos basado en expresiones regulares y un método de aprendizaje activo que involucre una estrategia de consulta y un criterio de detención para el clasificador propuesto.

1.3.2. Objetivos específicos

(I) Construir un *corpus gold standard* con textos biomédicos en español.

(II) Desarrollar e implementar un método que permita generar automáticamente un espacio de características basado en expresiones regulares.

(III) Desarrollar e implementar un clasificador de textos biomédicos utilizando expresiones regulares.

(IV) Formular e implementar una estrategia de consulta y un criterio de detención para el proceso de aprendizaje activo del clasificador de textos biomédicos propuesto, que permita capturar los ejemplos más informativos.

(V) Evaluar el desempeño del clasificador de textos biomédicos y del método de aprendizaje activo en términos de aciertos (ACC), Valor-F (F1) y cantidad de muestras de entrenamiento (curvas de aprendizaje), comparándolos con otros algoritmos de clasificación y estrategias de consulta para la selección de los ejemplos más informativos.

1.4. Contribuciones del autor

1.4.1. Clasificación de textos biomédicos

Las contribuciones de esta tesis en cuanto a la clasificación de textos biomédicos son las siguientes:

- Algoritmo para la construcción automática de un espacio de características en base a expresiones regulares para textos biomédicos, denominado extractor de características basado en expresiones regulares (FREGEX).
- Clasificador de textos biomédicos basado en expresiones regulares, denominado clasificador de textos basado en expresiones regulares (CREGEX).
- Clasificación de textos biomédicos en español utilizando el modelo de lenguaje pre-entrenado denominado representación de codificador bidireccional de transformadores (BERT).

1.4.2. Aprendizaje activo

Las contribuciones de esta tesis en cuanto a los métodos de aprendizaje activo en clasificación de textos biomédicos son las siguientes:

- Estrategia de consulta para el proceso de aprendizaje activo de CREGEX.
- Estrategia de consulta para el proceso de aprendizaje activo de BERT utilizando textos biomédicos en español.

1.4.3. Publicaciones

Los siguientes artículos de conferencia y de revista Web of Science (WoS) fueron obtenidos producto de este trabajo de tesis:

- R. L. Figueroa y C. A. Flores, "Extracting Information from Electronic Medical Records to Identify Obesity Status of a Patient Based on Comorbidities and Bodyweight Measures," *Journal of Medical Systems*, vol. 40, no. 8, pp. 1-9, 2016.
- C. A. Flores, R. L. Figueroa y J. E. Pezoa, "FREGEX: A Feature Extraction Method for Biomedical Text Classification using Regular Expressions," in *41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Berlín, Alemania, 2019, pp. 6085–6088.
- C. A. Flores, R. L. Figueroa, J. E. Pezoa y Q. Zeng-Treitler, "CREGEX: A Biomedical Text Classifier Based on Automatically Generated Regular Expressions," *IEEE Access*, vol. 8, pp. 29 270–29 280, 2020.
- C. A. Flores, R. L. Figueroa y J. E. Pezoa, "Active Learning for Biomedical Text Classification Based on Automatically Generated Regular Expressions," *IEEE Access*, 2020 (enviado).

1.5. Organización de la tesis

El trabajo de tesis se organiza de la siguiente forma. El Capítulo 2 presenta una revisión del estado del arte de los trabajos relacionados a la generación automática de expresiones regulares, clasificación de textos y aprendizaje activo. El Capítulo 3 describe los textos biomédicos utilizados en esta tesis y los métodos de clasificación y aprendizaje activo propuestos. El Capítulo 4 presenta el desempeño de los clasificadores implementados en términos de ACC, F1, curvas de aprendizaje y error de clasificación. Finalmente, en el Capítulo 5 se presenta un análisis de los resultados obtenidos y el trabajo futuro que se podría realizar para mejorar los resultados actuales de esta tesis.

Capítulo 2

Estado del arte y motivaciones

En este Capítulo se presenta el estado del arte sobre la generación automática de expresiones regulares, clasificación de textos y aprendizaje activo. Los trabajos que se analizan entregan los fundamentos teóricos que respaldan este trabajo de tesis y las principales motivaciones que dieron lugar a la formulación y desarrollo de algoritmos en base a expresiones regulares para la clasificación y aprendizaje activo en textos biomédicos.

2.1. Clasificación de textos

La clasificación o categorización de textos es un método de aprendizaje supervisado que permite asignar etiquetas o clases predefinidas a los textos en base a su contenido, siendo actualmente una herramienta de gran utilidad en el campo del NLP para organizar la creciente disponibilidad de información digital [12, 13].

Los problemas de clasificación de textos se resuelven utilizando algoritmos de aprendizaje supervisado, entrenando modelos en textos de etiqueta conocida [29]. Posteriormente estos modelos se utilizan para predecir las etiquetas de los textos no etiquetados. La clasificación de textos tiene muchas aplicaciones comerciales y de investigación en diversos dominios de estudio, incluyendo la biomedicina [11]. En este último caso, las principales fuentes de información provienen desde registros médicos electrónicos y artículos científicos, sin embargo, gran parte de los recursos utilizados están disponibles en inglés [30–32].

2.1.1. Representación de textos

La representación de textos busca transformar cada documento en un vector de características de tamaño constante para que puedan ser utilizados por los algoritmos de clasificación [29, 33].

El método más común para representar los textos es la denominada bolsa de palabras (BoW). En este modelo cada texto es representado por un vector de gran dimensión, donde cada *token* (palabras, números o símbolos) es una única característica representada por su frecuencia de aparición en el texto [34]. En comparación a la BoW, el método bolsa de n-gramas (BoN) considera como características la frecuencia de aparición de secuencias consecutivas de *tokens* (n-gramas) en los textos [33]. Una extensión natural a estos modelos de representación se denomina frecuencia de término – frecuencia inversa de documento (TF-IDF), que además considera la frecuencia inversa de los *tokens* en el total de documentos [35]. Si bien estos modelos han sido ampliamente utilizados en clasificación de textos debido a su simplicidad y efectividad, tienen como principales desventajas la incorporación de características consideradas "ruidosas" y la alta dimensionalidad de los vectores al representarlas [36]. Debido a ésto, es habitual el uso de métodos de selección de características [34]. Otro de los problemas de los métodos mencionados es que no consideran el orden en que los *tokens* aparecen en los textos y no representan necesariamente sus relaciones semánticas [37].

Otro método de representación de textos muy utilizado en el NLP se denomina análisis semántico latente (LSA) y permite analizar las relaciones entre los textos y los *tokens* que los contienen. Este método aplica en la matriz de frecuencias de los *tokens* una reducción de la dimensionalidad mediante la técnica estadística denominada descomposición en valores singulares (SVD) para encontrar conceptos semánticamente latentes [38]. Otros métodos permiten generar representaciones vectoriales distribuidas de baja dimensión de los *tokens* ("*embeddings*") entrenando redes neuronales en grandes colecciones de textos (*corpus*). Ejemplos de estos métodos son los denominados vectores de palabras (word2vec) y los vectores globales (GloVe), los cuales han demostrado ser capaces de capturar las relaciones semánticas de las palabras al analizar la probabilidad condicional de su contexto [39–41]. Si bien estos métodos han demostrado ser efectivos en el NLP, presentan ciertas dificultades con la polisemia, palabras infrecuentes y abreviaciones [42].

Otros modelos de representación más recientes, tales como las representaciones distribuidas desde modelos de lenguaje (ELMo) o BERT, entrenan DNNs en una gran colección de textos, pero a diferencia de las representaciones vectoriales distribuidas o *embeddings*, permiten obtener una representación contextualizada de las palabras. Estos métodos utilizan un modelo de lenguaje bidireccional que analiza el contexto global de una palabra antes de entregar una representación vectorial [42]. En el caso de BERT y ELMo se han propuesto modelos para aplicaciones en biomedicina, pero solo utilizando artículos científicos en inglés [43, 44]. La Figura 2.1 muestra un esquema de las representaciones basadas en expresiones regulares (REGEXES), BoW y BERT. Es posible observar que en el caso de las REGEXES el espacio de características está formado por secuencias de palabras y metacaracteres, más próximos al lenguaje natural, a diferencia de la BoW y BERT donde se obtienen representaciones vectoriales de los n_f *tokens*. En el caso de BERT, el primer *token* de cada secuencia corresponde al *token* especial [CLS] cuya representación vectorial en la salida de la última capa de los transformadores es utilizado para tareas de clasificación [1].

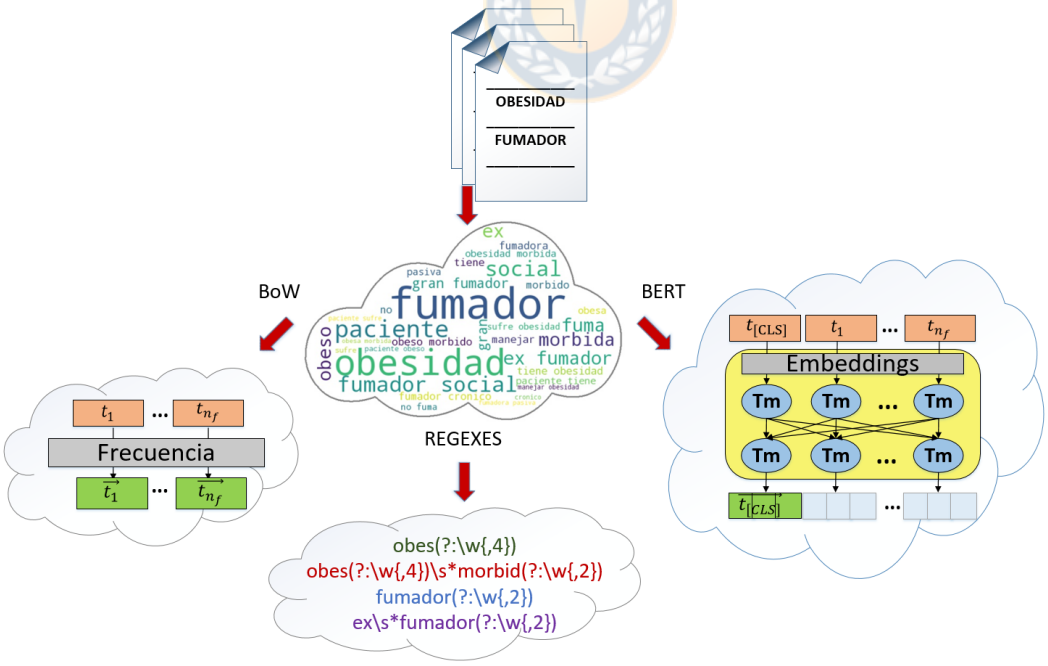

Figura 2.1: Tipos de representación de características: REGEXES, BoW y BERT. Fuente: Adaptación propia [1].

2.1.2. Algoritmos de clasificación

Naïve Bayes (NB) y la máquina de soporte vectorial (SVM) son dos de los algoritmos de aprendizaje automático más utilizados para la clasificación de textos debido a la simplicidad de implementación y precisión de sus predicciones [14]. Mientras NB es un clasificador probabilístico basado en el teorema de Bayes, SVM es un clasificador lineal que busca construir un hiperplano óptimo en el espacio multidimensional de características para separar los datos en dos clases [45].

En la actualidad, debido a la gran disponibilidad de datos y a las mejoras en la capacidad de procesamiento computacional ha existido un auge en el uso de algoritmos de clasificación basados en DNNs [46]. También se han propuesto mejoras a estos algoritmos basadas en arquitecturas de redes neuronales, tales como las redes neuronales convolucionales (CNNs) y las redes neuronales recurrentes (RNNs) [47,48]. Más recientemente, en el campo del NLP se ha utilizado el modelo de lenguaje pre-entrenado BERT junto con una red neuronal y una función de activación *softmax* para ajustar la representación vectorial a una tarea de clasificación específica [5, 49, 50]. Este enfoque de aprendizaje pertenece a una técnica denominada "transferencia de aprendizaje" [51].

Si bien las soluciones basadas en redes neuronales han permitido obtener buenos resultados en clasificación de textos, tienen el inconveniente que son poco interpretables a nivel de lenguaje natural. Debido a ésto los investigadores han considerado el uso de expresiones regulares porque permiten realizar un análisis de los *tokens* que los contienen, obteniéndose además resultados comparables a los algoritmos de clasificación más utilizados [20, 21]. En el área de la biomedicina, las expresiones regulares han sido utilizadas debido a la capacidad que tienen para modelar patrones secuenciales complejos, especialmente aquellos que incluyen atributos numéricos [20, 52]. En este sentido, se ha demostrado que BERT puede presentar una menor capacidad de generalización de números decimales tras el entrenamiento [53].

2.2. Generación automática de expresiones regulares

Las expresiones regulares se definen como una secuencia de caracteres que permiten representar patrones en los textos [22, 54]. Las expresiones regulares pueden ser creadas manualmente para diversos dominios de uso, como por ejemplo en tareas de extracción de información, validación de formularios, detección de correos no deseados (*spam*), extracción de *tokens* (*tokenización*), detección de negaciones, entre otros [54, 55]. En biomedicina gran parte de los trabajos que utilizan expresiones regulares se enfocan en tareas de extracción de información [52, 56–59].

Por otro lado, la generación automática de expresiones regulares desde ejemplos de entrenamiento sigue siendo un problema actual de investigación [60]. Uno de los métodos utilizados para generar expresiones regulares consiste en realizar una serie de transformaciones a una expresión regular de entrada para mejorar su desempeño en la tarea para la cual fue diseñada [27, 61–64]. Por lo tanto, el desempeño de estos métodos depende en gran medida del uso de un buen ejemplo inicial provisto por un experto en el dominio de estudio. Por ejemplo, Li *et al.* proponen un método denominado aprendizaje de expresiones regulares para extracción de información (ReLIE). Este método realiza múltiples transformaciones a una expresión regular de entrada utilizando meta-caracteres o caracteres especiales sin un significado literal (por ejemplo, grupos especiales, cuantificadores y disyunciones) hasta que el desempeño de dicha expresión regular no pueda ser mejorado. Los resultados de este trabajo indican que ReLIE obtuvo un mejor desempeño que el algoritmo denominado campo aleatorio condicional (CRF) y que también se pudo mejorar el desempeño de este último en términos de ACC al entrenarlo con características extraídas con ReLIE.

Otros métodos no requieren una expresión regular de entrada, pero requieren ejemplos de entrenamiento etiquetados con los segmentos de textos de interés [52, 65, 66]. Por ejemplo, Murtaugh *et al.* proponen un método denominado extractor de descubrimiento de expresión regular (REDEx) para extraer información antropométrica (por ejemplo, altura, peso, índice de masa corporal y circunferencia abdominal) desde textos biomédicos [52]. Este método construye una expresión regular a partir de un valor numérico a extraer, convirtiendo los segmentos

de textos que anteceden y preceden al valor en expresiones regulares (por ejemplo, reemplazando puntuación, números y espacios por meta-caracteres). Posteriormente, de forma progresiva REDEx añade estas expresiones regulares al valor numérico a extraer hasta no obtener falsos positivos en el conjunto de entrenamiento. Los autores de este trabajo mencionan que REDEx obtuvo desempeños sobre el 98 % en términos de ACC y F1.

Otros métodos incluyen programación genética o programación dinámica, diferenciándose de los trabajos previos en la forma de evaluar las expresiones regulares generadas en el conjunto de entrenamiento y en el uso de distintos meta-caracteres [20, 67–69]. Por ejemplo, Bartoli *et al.* utilizaron programación genética para generar una población de expresiones regulares a partir de un ejemplo etiquetado de entrada [68]. Estas expresiones regulares son modificadas iterativamente utilizando operadores genéticos, tales como mutación y cruzamiento, hasta un número máximo de generaciones o se haya alcanzado un desempeño máximo de acuerdo a una función *fitness*, la cual considera el tamaño de la expresión regular generada y una medida de distancia (*Levenstein*) respecto al segmento etiquetado de interés.

Por otro lado, Bui y Zeng-Treitler proponen un método denominado descubrimiento de expresión regular (RED), el cual usa programación dinámica para generar automáticamente expresiones regulares [20]. Este método genera expresiones regulares combinando secuencias de *tokens* (frases), las cuales son extraídos tras un proceso de alineación local de los textos de entrenamiento mediante el algoritmo de Smith-Waterman (SW). Además, se normalizan los números y se controla la cantidad de caracteres que pueden ser insertados entre los *tokens* mediante el uso de meta-caracteres. Posteriormente, las expresiones regulares son evaluadas en el conjunto de entrenamiento para luego ser filtradas por un umbral de desempeño. Los autores indican que RED obtuvo desempeños sobre el 80 % en términos de ACC y F1 en tareas de clasificación de textos biomédicos, superando el desempeño obtenido por SVM.

2.3. Aprendizaje activo

Para el entrenamiento de los algoritmos de clasificación es necesario disponer de suficientes textos correctamente etiquetados. Sin embargo, etiquetar manualmente

ejemplos de entrenamiento puede ser altamente costoso en términos de tiempo y recursos, requiriéndose además anotadores especializados [23]. Ante este problema, el AL ofrece una alternativa para reducir la cantidad de ejemplos necesarios para entrenar a los algoritmos de clasificación [24]. A diferencia del aprendizaje pasivo (PL) donde los ejemplos de entrenamiento son seleccionados aleatoriamente, el AL permite tener un control de los ejemplos seleccionados dependiendo del algoritmo de aprendizaje automático que se esté utilizando [70]. La Figura 2.2 muestra el esquema de funcionamiento del proceso de AL. A diferencia del enfoque clásico de clasificación, en AL los parámetros del modelo se ajustan progresivamente según la disponibilidad de datos (ver Figura 2.2) [2].

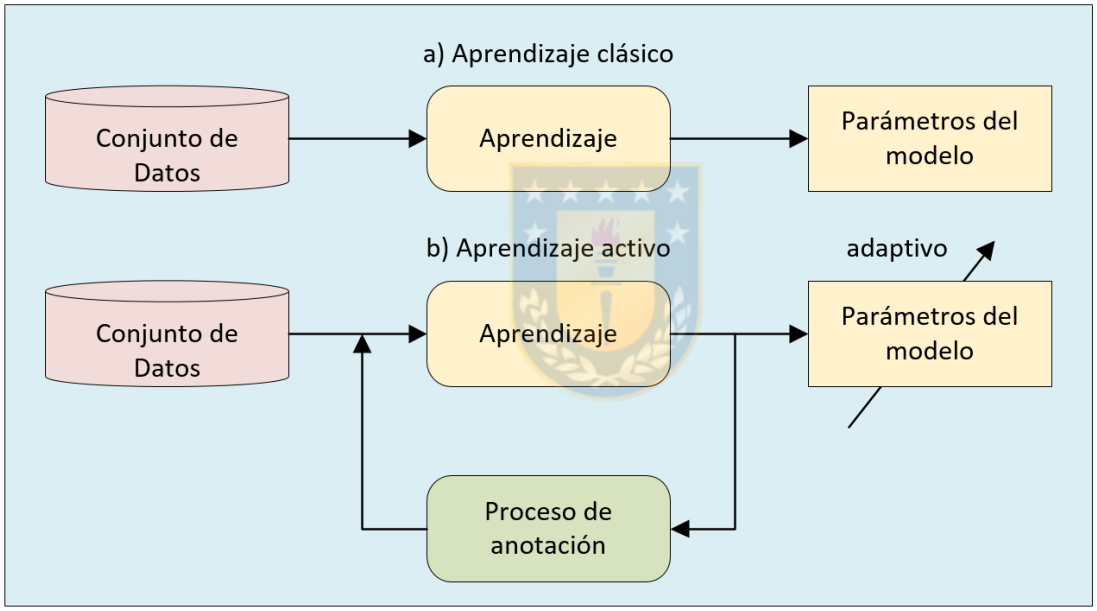

Figura 2.2: Esquema comparativo entre el enfoque de aprendizaje clásico y activo. Fuente: Adaptación propia [2].

En clasificación de textos el enfoque de AL más utilizado selecciona los ejemplos de entrenamiento desde un gran conjunto de datos no etiquetado ("*pool*" de datos) [29]. En este enfoque se definen tres conjuntos de datos: el conjunto de datos no etiquetado (X_U), un conjunto inicial de datos de entrenamiento (X_I, Y_I) y el conjunto de pruebas (X_T). El proceso de AL consiste en una etapa de inicialización y una etapa de selección de los ejemplos considerados más informativos (ver Algoritmo 1) [71]. Durante la etapa de inicialización se selecciona aleatoriamente al menos un ejemplo por clase desde el conjunto de datos no etiquetados para conformar un conjunto inicial de entrenamiento. Posteriormente,

se utiliza alguna estrategia de consulta de un algoritmo de clasificación para seleccionar iterativamente n_q ejemplos considerados más informativos. En cada iteración, los ejemplos seleccionados X_q son etiquetados por un experto E en el dominio de estudio para ser incorporados al conjunto de entrenamiento y, de esta forma, re-entrenar al clasificador utilizado hasta que se cumpla un criterio de detención. En el conjunto de pruebas se evalúa la efectividad del algoritmo de clasificación utilizado, permitiendo obtener curvas de aprendizaje.

Algoritmo 1: Enfoque de aprendizaje activo tipo *pool*

I. Sea:
X, Y: textos de entrenamiento etiquetados
X_U, X_q: textos no etiquetados y textos seleccionados
Y_q: etiquetas para X_q
E, q: experto en el dominio, estrategia de consulta
n_q: cantidad de ejemplos seleccionados
SC: criterio de detención
II. Inicialización:
Seleccionar n_q ejemplos desde X_U y luego etiquetarlos con E para definir un conjunto inicial de entrenamiento (X_I, Y_I)
Actualizar $X \leftarrow X \cup X_I$
Actualizar $Y \leftarrow Y \cup Y_I$
III. Selección:
mientras *el SC no se haya cumplido* **hacer**
    Entrenar un algoritmo de clasificación usando (X, Y)
    Seleccionar n_q ejemplos X_q desde X_U usando $q(\cdot)$
    Actualizar $X_U \leftarrow X_U \backslash X_q$
    Solicitar etiquetas Y_q para X_q utilizando E
    Actualizar $X \leftarrow X \cup X_q$
    Actualizar $Y \leftarrow Y \cup Y_q$
fin

2.3.1. Algoritmos de aprendizaje activo

Se han propuesto diversos algoritmos de AL para seleccionar progresivamente los ejemplos considerados más informativos para un algoritmo de aprendizaje

automático desde un conjunto de datos no etiquetado. Estos algoritmos se diferencian por la estrategia de consulta utilizada.

Por ejemplo, Lewis y Gale propusieron el método denominado muestreo por incertidumbre en el cual se utiliza un clasificador probabilístico para seleccionar los ejemplos que presentan entropía máxima [72, 73]. Otra estrategia de consulta denominada consulta por comité fue propuesto por Seung *et al.* y utiliza múltiples algoritmos de clasificación para seleccionar los ejemplos que presentan el mayor grado de desacuerdo entre este comité de clasificadores [74]. Otra estrategia de consulta propuesta por Tong y Koller se denomina margen simple y utiliza como criterio de selección los ejemplos que tienen una menor distancia al hiperplano de separación de las clases de una SVM [75]. Posteriormente, Brinker incorpora la similitud coseno a la estrategia de consulta de SVM para aportar mayor diversidad a los ejemplos seleccionados [76].

En otros métodos se utilizan técnicas de agrupamiento (*clustering*) para seleccionar ejemplos desde el conjunto de datos no etiquetado [77–79]. Sin embargo, la principal desventaja de estos métodos es que el desempeño del proceso de AL va a depender de la calidad de los grupos formados por el algoritmo de agrupamiento utilizado.

Otras estrategias de consulta más recientes buscan estimar la incertidumbre de las predicciones en algoritmos de clasificación basados en redes neuronales [80, 81]. En este sentido, los modelos Bayesianos permiten cuantificar la incertidumbre de las predicciones, pero tienen un alto costo computacional. Como una alternativa a este tipo de modelos, Gal propone el método *Monte Carlo dropout* que utiliza la técnica de regularización de redes neuronales denominada *dropout* como aproximación a una inferencia del tipo Bayesiana [80]. La técnica de *dropout* permite desconectar aleatoriamente unidades dentro de la red neuronal para evitar el sobreajuste (*overfitting*) durante la etapa de entrenamiento [82]. Para medir el nivel de incertidumbre se realiza una determinada cantidad de predicciones en un mismo ejemplo, obteniéndose de esta forma distribuciones de probabilidades que pueden ser analizadas estadísticamente.

2.3.2. Criterio de detención

Un aspecto muy importante a considerar en el AL es establecer un criterio para detener el proceso de aprendizaje. Sin un criterio de detención, el proceso de AL

seleccionará la totalidad de los ejemplos no etiquetados para conformar el conjunto de entrenamiento.

Algunos criterios de detención analizan el costo de obtener nuevas etiquetas, establecen un valor máximo de desempeño del clasificador o tamaño de muestra de entrenamiento, o analizan la calidad de los ejemplos en los distintos conjuntos de datos [73, 83]. En este último caso es posible analizar los ejemplos sin etiquetar, un conjunto de datos adicional, los ejemplos de entrenamiento seleccionados, o una combinación de estos conjuntos de datos [84].

Por ejemplo, Bloodgood y Vijay-Shanker proponen analizar las predicciones en términos de nivel de acuerdo (*kappa*) de diferentes modelos sobre el conjunto de datos no etiquetado hasta que dichas predicciones se estabilicen [85]. Por otro lado, Vlachos propone analizar el desempeño de un clasificador en términos de entropía en un conjunto de datos adicional hasta determinar una disminución consistente de este desempeño durante el proceso de aprendizaje [86]. Sin embargo, este método implica disponer de un conjunto representativo de datos, y en el caso que sea anotado, contradiría el principio de AL de reducir la cantidad de ejemplos etiquetados.

Finalmente, en el caso de analizar los ejemplos de entrenamiento seleccionados, Ghayoomi propone analizar la varianza de los resultados obtenidos desde la estrategia de consulta en cada iteración [87]. La justificación de este método se fundamenta en que al comienzo del aprendizaje el clasificador no está lo suficientemente entrenado por lo que los resultados de la estrategia de consulta no presentarán un alto nivel de variabilidad. A medida que el conjunto de entrenamiento aumenta, el clasificador pasa de estar no entrenado a entrenado, reflejándose en una mayor variabilidad de los resultados de la estrategia de consulta. Finalmente, una vez que el clasificador está lo suficientemente entrenado, los resultados de la estrategia de consulta estarán cercanos a la media con un bajo nivel de variabilidad. En esta última etapa se debe detener el proceso de AL.

2.4. Discusión

Las expresiones regulares han demostrado ser una alternativa confiable para la clasificación de textos biomédicos debido a que son fácilmente analizables a nivel de lenguaje natural por un experto en el dominio de estudio y a la capacidad

que tienen dichas expresiones para representar patrones secuenciales en lo textos, incluyendo atributos numéricos. En este sentido, el uso de algoritmos de alineación de secuencias, tales como SW o Needleman-Wunsch (NW), han demostrado ser útiles para la extracción de patrones comunes en los textos [20, 88]. Sin embargo, los trabajos que utilizan expresiones regulares para la clasificación de textos biomédicos utilizan como base de comparación los algoritmos de clasificación tradicionales, tales como SVM y NB, y no consideran métodos más recientes como BERT [20, 21]. Además, gran parte de los trabajos relacionados se enfocan en el idioma inglés, lo cual motiva a investigar aplicaciones en español para contribuir al estado del arte.

Debido a la gran disponibilidad de información textual, el enfoque de AL más utilizado busca seleccionar los ejemplos más informativos desde una gran colección de datos no etiquetado (*pool* de datos). En este sentido, los principales algoritmos de selección utilizan los clasificadores SVM y NB [72, 75, 76]. Recientemente se ha propuesto un método que permite medir la incertidumbre en algoritmos de clasificación basados en redes neuronales, pero no ha sido estudiado el impacto que podría tener en algoritmos como BERT [80, 81]. Por otro lado, otra de las principales motivaciones de esta tesis es que previo a este trabajo no existe una estrategia de consulta de AL que permita determinar cuáles son los ejemplos más informativos para un clasificador de textos biomédicos basado en expresiones regulares, existiendo solo algunos trabajos relacionados a la extracción de información [27, 28].

Los algoritmos de AL utilizan alguna métrica de incertidumbre del clasificador como estrategia de consulta para determinar cuáles son los ejemplos más informativos, siendo los más utilizados las distancias al hiperplano (SVM) o la probabilidad de pertenecer a alguna determinada clase (redes neuronales, NB). En el caso de un clasificador de textos biomédicos basado en expresiones regulares se podría construir una estrategia de consulta o función de selección a partir del desempeño que dichas expresiones regulares tuvieron durante el entrenamiento, y/o considerando la cantidad de coincidencias que éstas tienen en el conjunto de datos no etiquetados.

Capítulo 3

Materiales y métodos

En este Capítulo se presenta una descripción de los textos biomédicos utilizados, así como los algoritmos de clasificación y aprendizaje activo basados en la generación automática de expresiones regulares. La generación automática de expresiones regulares, denominada FREGEX, se fundamenta principalmente en el uso de dos algoritmos de alineación de secuencias para la extracción de *tokens* representativos desde los textos: el algoritmo de NW y el algoritmo de SW. El clasificador, denominado CREGEX, considera en su función de decisión la cantidad de coincidencias que las expresiones regulares tienen en un ejemplo de prueba para asignar la clase de una expresión regular o de un ejemplo de entrenamiento. Por otro lado, el proceso de aprendizaje activo de CREGEX considera como función de selección de los ejemplos más informativos o estrategia de consulta el desempeño que las expresiones regulares tienen durante su generación automática y la similitud de SW.

3.1. Conjuntos de datos y pre-procesamiento

Como conjuntos de datos se utilizaron textos biomédicos de-identificados provenientes del hospital Guillermo Grant Benavente (HGGB) de Concepción, previa autorización del comité de ética de este recinto de salud (ver Anexo A1.1). Los textos biomédicos contienen información sobre la obesidad, los tipos de obesidad y el hábito de tabaquismo, los cuales fueron anotados por un grupo de estudiantes de Ingeniería Civil Biomédica para obtener un *corpus gold standard*. El proceso de anotación se realizó utilizando una herramienta programada en *Python*

y diseñada en *Qt Designer* (ver Anexo A2.1), permitiendo recolectar palabras claves para cada problema de clasificación. Para medir el nivel de acuerdo entre el grupo de anotadores se utilizó el índice de kappa (k), obteniéndose en todos los casos $k > 0{,}81$ (acuerdo casi perfecto) [1]. La Tabla 3.1 muestra la distribución de las clases en los conjuntos de datos, así como las palabras claves obtenidas tras el proceso de anotación.

Los textos biomédicos fueron pre-procesados, removiendo los excesos de espacio y convirtiéndolos a minúsculas, para finalmente *tokenizarlos* considerando los espacios. Para facilitar el proceso de *tokenización* se añadieron espacios entre los caracteres no alfanuméricos para obtener *tokens* más específicos. Por ejemplo, como resultado de este proceso es posible extraer los *tokens* "paciente", "diabético", "(", "ex", "fumador" y ")" a partir del texto "paciente diabético (ex fumador)".

Tabla 3.1: Descripción de los conjuntos de datos. Fuente: Elaboración propia.

Conjunto de datos	Palabras claves	Distribución de Clases	Índice de kappa
OBESIDAD	obes*, imc, peso, sobrepeso	Negativa (303), positiva (858)	$0.98^{(a)}$
TIPOS DE OBESIDAD	obes*, imc, peso, sobrepeso	Moderada (185), severa (152), mórbida (572)	$0.97^{(a)}$
TABAQUISMO	tab*, fum*, cig*, caj*	Negativa (505), positiva (582)	$0.86^{(b)}$

$^{(a)}$Índice de kappa de Cohen. $^{(b)}$Índice de kappa de Fleiss.
*Raíz de la palabra.

3.2. Definición del problema

Esta tesis busca desarrollar algoritmos para la clasificación de textos biomédicos en base a la generación automática de expresiones regulares y aprendizaje activo.

El método propuesto para la clasificación de textos biomédicos genera automáticamente expresiones regulares a partir de una colección de n textos biomédicos etiquetados utilizando $l \geq 2$ clases, es decir, es posible aplicar el método tanto en problemas binarios como multiclases. Más formalmente, se define

[1] El índice de kappa es una medida estadística utilizada para evaluar el nivel de acuerdo entre un grupo de anotadores (ver Anexo A3.1)

el conjunto de textos como $X = \{x_1, \ldots, x_n\}$, el conjunto de etiquetas como $L = \{1, \ldots, l\}$, y el proceso de etiquetado supervisado según el mapeo $\mathcal{L} : X \to L$, el cual para cada $x_i \in X$ crea una etiqueta $y_i = \mathcal{L}(x_i),\ y_i \in L$. La colección de todas las etiquetas del conjunto de entrenamiento se puede obtener según $Y = \mathcal{L}(X)$. Además, para evaluar la efectividad del método propuesto se utiliza un conjunto de datos independiente X_T que contiene n_t textos de prueba.

El método propuesto se divide en dos etapas, tal como se muestra en la Figura 3.1: (i) construcción de un espacio de características basado en la generación automática de expresiones regulares; y (ii) definición de un clasificador de textos biomédicos. En una primera etapa denominada FREGEX el método utiliza algoritmos de alineación de secuencias para generar automáticamente un espacio de características basado en expresiones regulares para X mediante el mapeo biyectivo $\Phi(x_i) : X \to R_i \subseteq R$, el cual genera n_i expresiones regulares para el texto de entrenamiento x_i, etiquetado como y_i, donde $R_i = (r_1^i(x_i), \ldots, r_{n_i}^i(x_i))$. De esta forma, una vez que la función de mapeo $\Phi(\cdot)$ es aplicado en todos los textos de entrenamiento X, se genera la colección $R = \cup_{i=1}^{n} R_i$, el cual contiene un número total de $|R|$ expresiones regulares que representan al conjunto de textos biomédicos. Posteriormente se realiza una selección de características, filtrando las expresiones regulares mediante palabras claves y se evalúan para obtener una medida de desempeño en el conjunto de entrenamiento. En una segunda etapa el método asigna a las expresiones regulares resultantes la clase del texto de entrenamiento donde fueron generadas automáticamente. De esta forma, el clasificador denominado CREGEX permite asignar una clase y_i a un texto de prueba x_i mediante la función de decisión $\delta(x_i) : X_T \to L$, donde $y_i = \delta(x_i)$.

Por otro lado, para el AL se define el conjunto de datos $D_{AL} = X \cup X_U$, donde X contiene n textos de entrenamiento, etiquetados según $Y = \mathcal{L}(X)$, mientras que X_U contiene n_u textos no etiquetados. El objetivo del AL es seleccionar iterativamente el conjunto $X_q \subseteq X_U$, el cual contiene los ejemplos considerados más informativos por el algoritmo de aprendizaje automático utilizado. Para determinar la importancia de cada texto en X_U una función asigna u_i, $i = 1, \ldots, |X_U|$ valores mediante la estrategia de consulta $q(x_i)$ (ver Algoritmo 1 para mayores detalles). De esta forma, en cada iteración el subconjunto X_q es etiquetado por un oráculo experto en D_{AL} (por ejemplo, un experto humano) de tal manera que X, Y, y X_U cambian iterativamente según: $X \leftarrow X \cup X_q$, $Y \leftarrow Y \cup \mathcal{L}(X_q)$, y $X_U \leftarrow X_U \backslash X_q$.

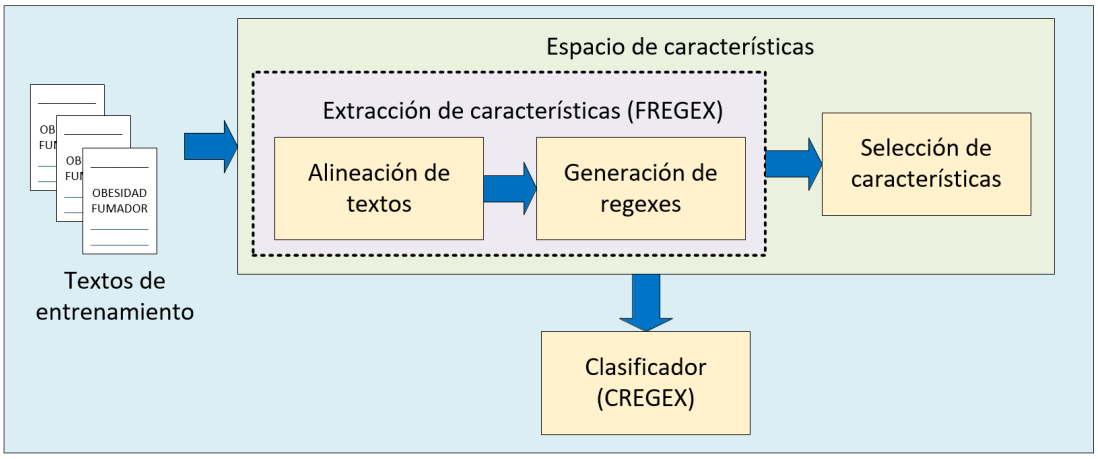

Figura 3.1: Esquema general para la clasificación de textos biomédicos basada en la generación automática de expresiones regulares (REGEXES). Fuente: Elaboración propia.

La Figura 3.2 muestra el proceso de aprendizaje activo en CREGEX. Se puede observar que el proceso de AL utiliza una estrategia de consulta para seleccionar progresivamente ejemplos sin etiqueta desde una gran colección de textos (*pool* de datos), los cuales son etiquetados por un experto en el dominio (oráculo) para re-entrenar el clasificador. Este proceso se repite hasta el cumplimiento de un criterio de detención.

3.3. Algoritmos para la construcción de un espacio de características

Para la construcción de un espacio de características basada en la generación automática de expresiones regulares se propone el uso de algoritmos de alineación de secuencias. En primer lugar, se aplica una alineación global de las palabras similares para representarlas por un patrón común utilizando el algoritmo de NW. En segundo lugar, se aplica una alineación local de los textos de entrenamiento que pertenecen a una misma clase para extraer secuencias de *tokens* representativas para un determinado problema de clasificación utilizando el algoritmo de SW.

3.3.1. Alineación global: algoritmo de Needleman-Wunsch

Para facilitar la extracción de *tokens* representativos para cada problema de clasificación, en primer lugar, se aplica agrupamiento (*clustering*) jerárquico en

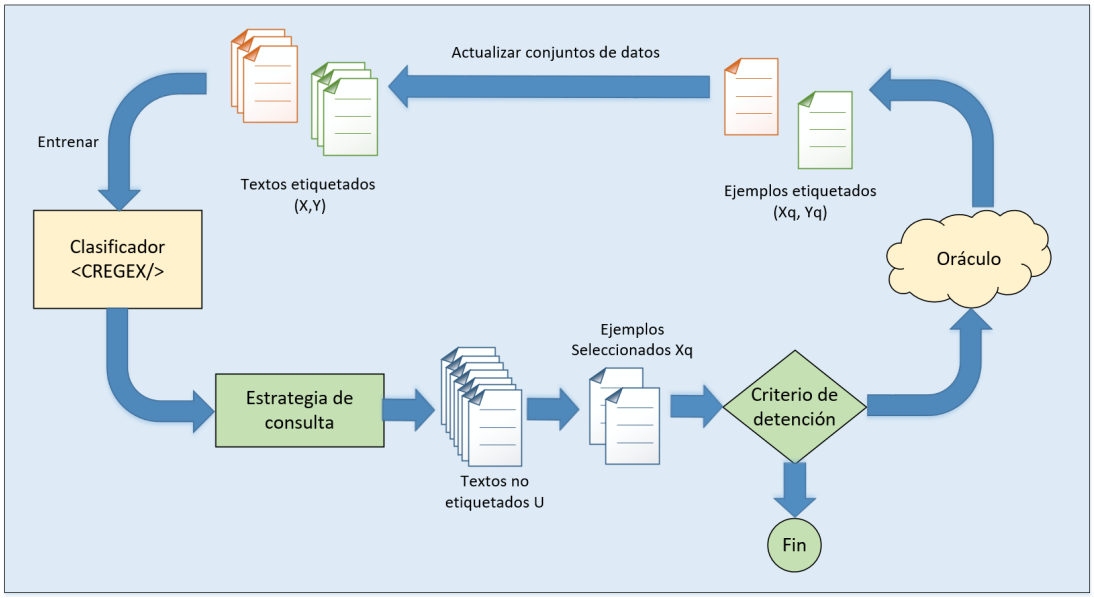

Figura 3.2: Esquema general del método de aprendizaje activo en CREGEX. Fuente: Adaptación propia [3].

los textos de entrenamiento para formar grupos de palabras similares y poder representarlos mediante un patrón común (expresión regular). De esta forma, se intenta capturar las variantes léxicas de las palabras en términos de número y género gramatical, incluyendo los errores ortográficos presentes en los textos. El agrupamiento jerárquico considera como métrica la distancia *Levenshtein* y un valor de corte igual a cuatro para el dendrograma, el cual fue determinado tras el análisis exploratorio de los datos [89]. Los verbos fueron excluídos del agrupamiento debido a que éstos contienen información temporal importante sobre las enfermedades o hábitos de los pacientes, para lo cual se consideró una lista de verbos en español [2] y las palabras en modo infinitivo. Una vez que se ha realizado el agrupamiento jerárquico a los textos de entrenamiento, se procede a aplicar el algoritmo de NW en cada uno de los grupos de palabras. Este algoritmo utiliza una matriz para obtener el alineamiento global óptimo entre las secuencias analizadas, asignando valores positivos y negativos según las coincidencias encontradas en los caracteres (ver Figura B1.1 del Anexo) [90]. Posteriormente, en una etapa de rastreo el algoritmo de NW traza la ruta de las secuencias alineadas. En este caso el algoritmo de NW es utilizado para alinear las letras de los grupos de palabras y encontrar un patrón común, considerando como base de esta alineación múltiple la palabra más frecuente según el conjunto de

[2] https://github.com/christopherfj/CREGEX/blob/master/verbos_ESP.xls

entrenamiento. Posteriormente, el método propuesto calcula la cantidad máxima de caracteres que pueden ser insertados entre las letras alineadas y utiliza el metacaracter "{, máx}" para controlar esta distancia. Finalmente, cada una de las palabras que pertenece a un mismo grupo es reemplazada por un patrón común en los textos de entrenamiento para la extracción de *tokens* representativos. Un ejemplo del agrupamiento jerárquico y posterior alineación global se muestra en la Figura 3.3.

3.3.2. Alineación local: algoritmo de Smith-Waterman

En esta etapa, el método propuesto aplica el algoritmo de SW en los textos de entrenamiento que pertenecen a una misma clase para encontrar regiones de similitud local y extraer *tokens* representativos para el problema de clasificación. Si los textos contienen números, éstos son reemplazados por un patrón que contiene metacaracteres para representar intervalos numéricos. Se consideró un rango igual a cinco debido a que este es utilizado para representar los diferentes niveles de obesidad [91]. Por ejemplo, el *token* "24.3" es reemplazado por el patrón "2[0-4]{1}(?:[\.\,]\d+)", mientras que el *token* "25.4" es reemplazado por el patrón "2[5-9]{1}(?:[\.\,]\d+)?"[3].

Al igual que el algoritmo de NW, el algoritmo de SW utiliza una matriz de alineación para analizar las secuencias. Sin embargo, una de las principales diferencias respecto al algoritmo de NW es que SW no asigna valores negativos en cada una de las celdas de la matriz de alineación. Un valor negativo indica que los elementos actuales no presentan similitudes. Al considerar solo valores positivos, es posible eliminar el efecto de las alineaciones previas, permitiendo encontrar nuevas regiones de similitud locales (ver Figura B1.1 y B2.1 del Anexo) [92]. De esta forma, el algoritmo de SW es más apropiado cuando las secuencias analizadas varían en palabras y tamaño, como es el caso de los textos biomédicos [20].

Finalmente, FREGEX construye un espacio de características, generando una colección de expresiones regulares a partir de los *tokens* extraídos desde los textos de entrenamiento. En cada *token* se reemplazan los espacios por el metacaracter "\s*" (representa cero o más espacios) y se inserta el caracter *backslash* ("\") entre los caracteres no alfanuméricos.

[3]Los números entre paréntesis cuadrados "[ ]" representan la unidad, mientras que "(?:[\.\,]\d+)?" representa la parte decimal ya sea utilizando coma o punto

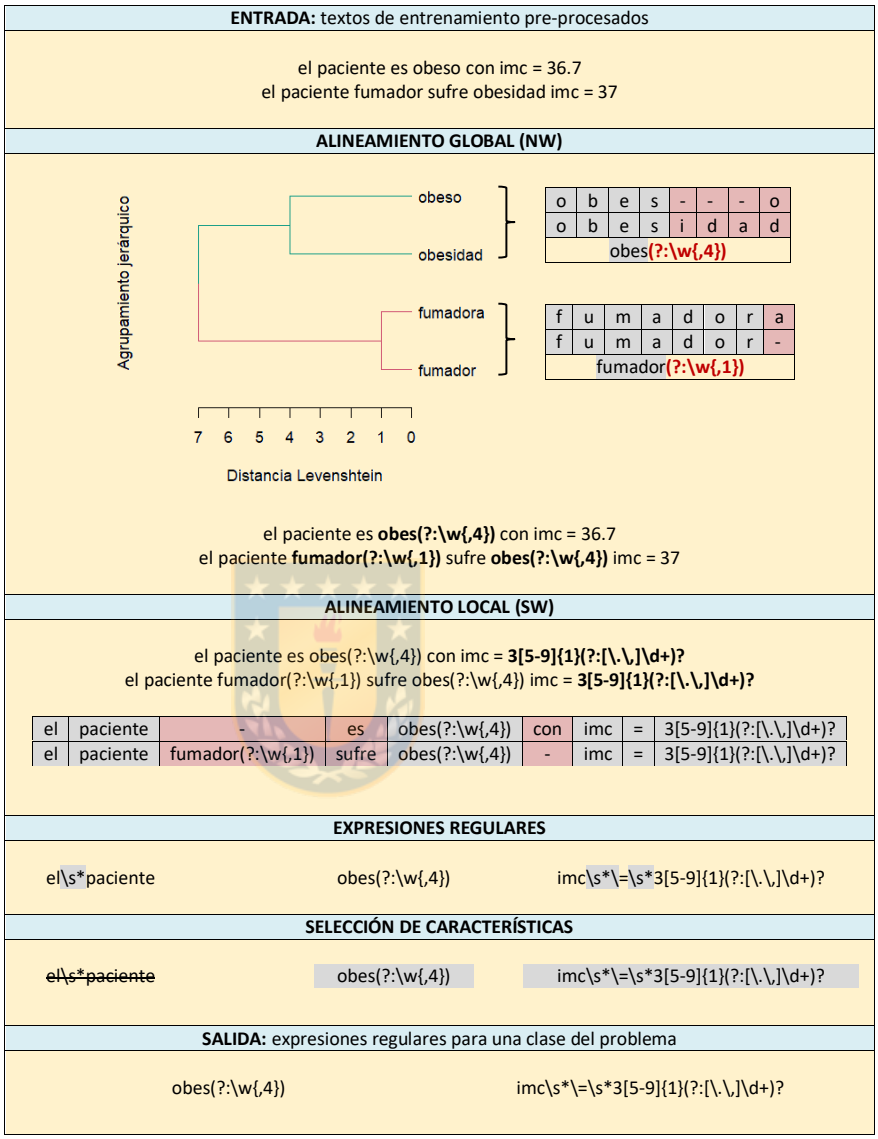

Figura 3.3: Ejemplo de generación automática de expresiones regulares para el conjunto de datos OBESIDAD (clase positiva). Fuente: Elaboración propia.

3.3.3. Selección de características

Sobre el espacio de características en la forma de expresiones regulares se aplica una selección de características, filtrando por palabras claves para cada problema de clasificación. En este trabajo se utilizaron las palabras claves de la Tabla 3.1, las cuales fueron obtenidas tras el proceso de anotación. El objetivo de esta etapa es reducir la cantidad de expresiones regulares que se generaron desde el conjunto de entrenamiento, manteniendo solo aquellas que están relacionadas al problema de clasificación. En el ejemplo de la Figura 3.3, la expresión regular "el\s*paciente" fue filtrada porque no contiene una palabra clave para el conjunto de datos OBESIDAD. Finalmente, el método asigna a cada expresión regular la clase del texto de entrenamiento donde el *token* fue extraído. Además, se evalúa la capacidad predictiva de cada expresión regular, considerando las coincidencias correctas que dichas expresiones regulares tienen en los textos de entrenamiento.

3.4. Algoritmo para la clasificación de textos

Las expresiones regulares generadas automáticamente pueden ser utilizadas para clasificar textos de prueba $x_i \in X_T$. En esta etapa el clasificador denominado CREGEX asigna una etiqueta y_i dependiendo de la cantidad de expresiones regulares que coinciden en el texto de prueba. En este sentido, dos casos posibles pueden surgir: (i) ninguna expresión regular coincide con un texto de prueba x_i; y (ii) n_r expresiones regulares coinciden con un texto de prueba x_i.

Si ninguna expresión regular coincide con un texto de prueba, CREGEX asigna la clase del texto de entrenamiento con la mayor similitud según el algoritmo de SW, $\text{sw\_sim}(x_i, x_j)$, $j = 1, \ldots, n$. Por otro lado, si n_r expresiones regulares coinciden con un texto de prueba, CREGEX asigna la clase $y_j = \mathcal{L}(\Phi^{-1}(r_j))$ de la j-ésima expresión regular, $r_j \equiv r(x_j)$, que tiene el mejor desempeño en términos de precisión $Pr(r_j)$, con $j = 1, \ldots, n_r$. Este valor de precisión es calculado durante la etapa de entrenamiento como el cociente entre el número de coincidencias correctas para la clase correspondiente y el número total de coincidencias. De esta forma, la función de decisión de CREGEX se obtiene según:

$$\delta(x_i) = \begin{cases} \mathcal{L}\left(\underset{j\in[1,n]}{\text{argmax}}\ \text{sw\_sim}(x_i, x_j)\right), & n_r = 0 \\ \mathcal{L}\left(\Phi^{-1}\left(\underset{j\in[1,n_r]}{\text{argmax}}\ Pr(r_j)\right)\right), & n_r > 0 \end{cases} . \tag{3.1}$$

3.5. Algoritmo de aprendizaje activo

La versión de AL de CREGEX introduce una estrategia de consulta que busca seleccionar los ejemplos asociados a expresiones regulares con un alto nivel de incertidumbre, es decir, los ejemplos considerados más informativos para este clasificador según la métrica de precisión.

Inicialmente, las expresiones regulares tenderán a tener un alto nivel de incertidumbre (bajo nivel de precisión), resultando en un enfoque de aprendizaje tipo *greedy* o voraz. Sin embargo, a medida que más ejemplos informativos son seleccionados, las precisiones de las expresiones regulares comienzan a mejorar hasta estabilizarse, resultando en un aprendizaje más conservativo inducido por la similitud de los textos. Los textos biomédicos que no coinciden con expresiones regulares o coinciden con expresiones de bajo valor de precisión satisfacen esta idea. Matemáticamente, la estrategia de consulta propuesta puede ser expresada según:

$$\underset{x_i \in X_U}{\text{argmin}}\, q(x_i) \tag{3.2}$$

donde:

$$q(x_i) = \begin{cases} \text{máx}_{j\in[1,n]}\, \text{sw\_sim}(x_i, x_j), & n_r = 0 \\ \text{máx}_{j\in[1,n_r]}\, Pr(x_j), & n_r > 0 \end{cases} . \tag{3.3}$$

Los ejemplos considerados más informativos son aquellos que tienen los valores mínimos de esta estrategia de consulta (mayor incertidumbre). Además se implementó un criterio de detención para el proceso de AL, utilizando el método de la varianza aplicado a los valores de la estrategia de consulta $q(\cdot)$ de los ejemplos considerados más informativos [87]. Este método analiza en cada iteración una

ventana histórica de n_v varianzas, $V = \{V_1, \ldots, V_{n_v}\}$ desde el valor actual V_{n_v} para determinar si se ha alcanzado o no una disminución sostenida a partir de un valor máximo. Matemáticamente, el AL debe detenerse al cumplirse el siguiente criterio: $V_2 > V_1$, y $V_2 > \text{máx}\{V_3, \ldots V_{n_v}\}$.

Capítulo 4

Resultados

En este Capítulo se muestran los resultados de los experimentos realizados para evaluar a los algoritmos de clasificación de textos biomédicos implementados. Los resultados de clasificación muestran la efectividad de los algoritmos basados en expresiones regulares y aprendizaje activo para la clasificación de textos biomédicos en comparación a los métodos existentes.

4.1. Evaluación de desempeño

Para comparar el desempeño de CREGEX, se implementaron clasificadores basados en SVM, NB y BERT. En el caso de NB se consideró un modelo multinomial, mientras que en SVM un *kernel* lineal, manteniendo los demás parámetros por defecto [6, 93]. La Figura 4.1 muestra un esquema de los clasificadores implementados basados en una SVM, NB y BERT. En el caso de SVM se muestra el hiperplano de separación de las clases (ejemplo de dos dimensiones) junto con los vectores de soporte, mientras que en NB se muestra la independencia condicional de las características para una determinada clase. Finalmente, en el caso de BERT se muestra que para tareas de clasificación de textos es posible utilizar la representación vectorial del *token* [CLS], obtenido de la salida de la última capa de los transformadores, junto con un clasificador *softmax*, donde $\{p_1, \ldots, p_l\}$ son las probabilidades predictivas para un texto de prueba [1, 5].

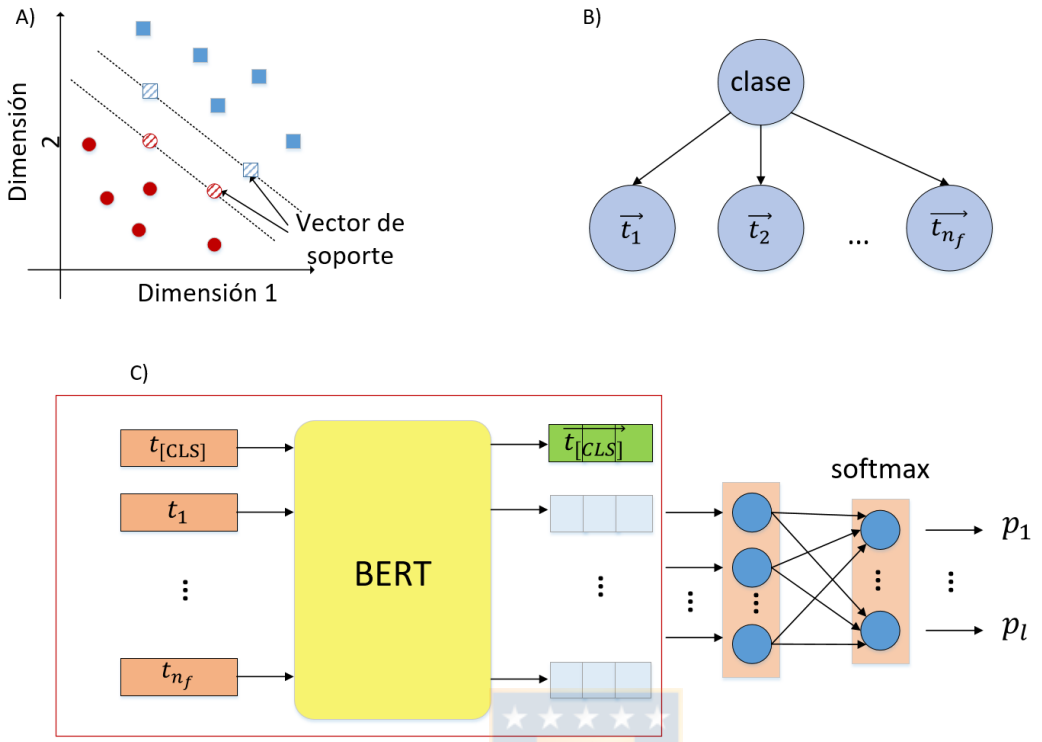

Figura 4.1: Esquema de algoritmos de clasificación. A: SVM, B:NB, C: BERT. Fuente: Adaptación propia [4–6].

En los clasificadores SVM y NB se utilizaron como características secuencias consecutivas de *tokens* (n-gramas), considerando 1-gramas (N1), es decir, secuencias consecutivas de un *token* y 2-gramas (N2), es decir, secuencias consecutivas de dos *tokens* [94]. Además, también se utilizó el espacio de características generado por FREGEX para el entrenamiento de estos clasificadores. Todos los tipos de características fueron representados matricialmente según el método TF-IDF [33, 35]:

$$TF - IDF = TF_{t,d} \times IDF_{t,D}, \tag{4.1}$$

$$IDF_{t,D} = log_{10}(\frac{|D|}{d}), \tag{4.2}$$

Donde TF representa la frecuencia absoluta de cada uno de los *tokens* t en cada uno de los textos d, mientras que IDF representa la frecuencia inversa de los *tokens* en todo el dataset D. Por otro lado, en el caso de BERT se utilizó el modelo pre-entrenado *base-multilingual-uncased*, y para fines de clasificación se

implementó un clasificador *softmax* para ajustar los parámetros del modelo a las tareas de clasificación, considerando los siguientes parámetros: número de épocas = 4, *batch size* = 8, *dropout* = 0.1, y un optimizador Adam (tasa de aprendizaje = 2^{-5}) [1, 95, 96].

Para el entrenamiento y evaluación de los clasificadores se implementó validación cruzada de 10 iteraciones (10-*fold cross validation*), repitiendo los experimentos 10 veces para luego promediar los resultados de clasificación [97, 98]. En otras palabras, en cada experimento el 90 % de los datos se utilizó para conformar el conjunto de entrenamiento, mientras que el restante 10 % fue utilizado para evaluar el desempeño de los clasificadores en términos de ACC y F1 según:

$$ACC = \frac{TP + TN}{TP + FP + TN + FN}, \tag{4.3}$$

$$F1 = \frac{2TP}{2TP + FN + FP}, \tag{4.4}$$

Donde FP y FN corresponden a los errores de clasificación (falsos positivos y negativos), mientras que TP y TN a los aciertos (verdaderos positivos y negativos). La Figura 4.2 muestra una representación gráfica de las métricas denominadas precisión (del inglés, *precision*) y exhaustividad (del inglés, *recall*). A diferencia del ACC, el par precisión-exhaustividad entregan información sobre el comportamiento del clasificador respecto del porcentaje de predicciones positivas correctamente clasificadas y sobre el porcentaje de casos positivos capturados, respectivamente [99]. En este sentido, la métrica F1 busca un balance entre la precisión y la exhaustividad en un único valor (media armónica).

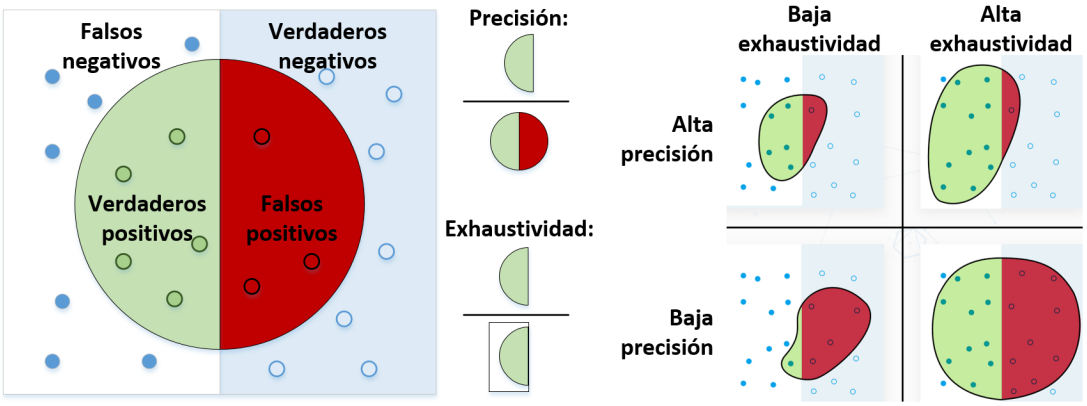

Figura 4.2: Representación gráfica de las métricas precisión y exhaustividad. Fuente: Adaptación propia [1].

Adicionalmente, se implementaron diferentes tipos de curvas de aprendizaje para evaluar los algoritmos propuestos. En primer lugar, se implementaron curvas de aprendizaje para analizar el comportamiento de los clasificadores SVM y NB en términos de cantidad de características n_f (%) y desempeño en términos de ACC (%) y F1 (%). Las características extraídas, ya sea mediante n-gramas y FREGEX, $tokens = \{t_1 \ldots, t_{n_f}\}$, fueron seleccionadas progresivamente de acuerdo a la importancia medida en términos de ganancia de información (IG) según:

$$\mathrm{IG}(\mathrm{t_i}) = \mathrm{H}(\mathrm{Y}) - \mathrm{H}(\mathrm{Y}/\mathrm{t_i}) \tag{4.5}$$

Donde H(Y) es el valor de entropía de las clases en el conjunto de datos, mientras que H(Y/t_i) es el valor de información mutua asociado al vector de características del *token* t_i y las clases Y del conjunto de datos. Por otro lado, en cuanto al AL y PL se implementaron curvas de aprendizaje para analizar el desempeño de los clasificadores en términos de cantidad de ejemplos de entrenamiento y desempeño en términos de ACC (%) y F1 (%). Para crear las curvas de aprendizaje se seleccionaron iterativamente 50 ejemplos desde el conjunto de datos no etiquetado, dependiendo de la estrategia de consulta utilizada [20]. En el caso del PL se seleccionaron aleatoriamente los ejemplos desde el conjunto de datos no etiquetado. Por otro lado, en cuanto a las estrategias de consulta de AL se consideró en el caso de SVM las distancias al hiperplano y la similitud coseno, mientras que en NB se

[1] https://towardsdatascience.com/whats-the-deal-with-accuracy-precision-recall-and-f1-f5d8b4db1021
https://medium.com/stradigiai/image-and-video-understanding-a-roadmap-for-implementation-a006f6f3fe0c
Fecha de último accesso: 25 de Diciembre de 2020 a las 16:52 hrs.

utilizó el criterio de entropía máxima [72,75,76]. En el caso del clasificador basado en BERT se utilizó el método de Monte Carlo Dropout, considerando el criterio de entropía máxima del promedio de 10 predicciones para un mismo ejemplo [80,81]. Por otro lado, en cuanto a CREGEX se analizó cómo el aprendizaje voraz o *greedy* y conservativo (normalizado) afectan conjunta e independientemente en la estrategia de consulta de la ecuación (3.2) en el proceso de AL. Matemáticamente, se utilizó una combinación convexa de la siguiente manera:

$$q'(x_i) = \lambda \max_{j\in[1,n_r]} Pr(x_j) + (1-\lambda) \max_{j\in[1,n]} \text{sw\_sim}(x_i, x_j), \tag{4.6}$$

Para analizar la ecuación (4.6) se utilizaron tres variantes del algoritmo según:

(I) AMB: $\lambda = 1$, es decir, la componente de la métrica de desempeño (Pr).

(II) CMB: $\lambda = 0{,}5$, es decir, una combinación de componentes de precisión y similitud de SW.

(III) DIV: $\lambda = 0$, es decir, la componente de diversidad (similitud de SW normalizada).

Notar que la estrategia de consulta propuesta de la ecuación (3.2) no combina las componentes, solo usa una de ellas dependiendo del valor de n_r. Además, se calculó el área bajo la curva de aprendizaje según el método del trapecio, normalizado por la cantidad de iteraciones. Por otro lado, se analizó la significancia estadística de los resultados obtenidos mediante las pruebas *T-student* para muestras relacionadas y la prueba no paramétrica *Wilcoxon signed-rank*, previo análisis de bondad de ajuste (normalidad) de los datos según las pruebas *Kolmogorov-Smirnov* y *Shapiro-Wilk* ($\alpha = 0.05$). Finalmente, para analizar los errores de clasificación de los clasificadores durante el entrenamiento y prueba se utilizó la métrica pérdida uno-cero (e) según [100]:

$$e(y_i, y_i') = \begin{cases} 1, & y_i \neq y_i' \\ 0, & \text{caso contrario,} \end{cases}, \tag{4.7}$$

Donde y_i y y_i' representan las predicciones y las etiquetas, respectivamente.

4.2. Espacio de características: FREGEX

Esta sección muestra los resultados de clasificación de los algoritmos NB y SVM entrenados con características extraídas en la forma de n-gramas y en base a expresiones regulares (FREGEX), las cuales fueron representadas matricialmente mediante el método TF-IDF .

4.2.1. Resultados de clasificación

La Tabla 4.1 muestra los resultados de clasificación de NB y SVM en términos de ACC (%) y F1 (%) utilizando N1, N2 y FREGEX. En la mayoría de los casos, excepto en el conjunto de datos TABAQUISMO en NB-N2, el uso de FREGEX mejoró el desempeño de SVM y NB en comparación al uso de n-gramas. Esto también se puede observar al analizar el desempeño por cada clase del problema. Según lo indicado en la Figura 4.3, FREGEX permitió mejorar el desempeño de SVM y NB con algunas excepciones en el conjunto de datos OBESIDAD (clase positiva) y TABAQUISMO (clase negativa y positiva). En resumen, FREGEX permitió mejorar el desempeño de SVM y NB en la mayoría de los casos en comparación al uso de n-gramas, siendo más efectivo en los conjuntos de datos con mayor presencia de información antropométrica (por ejemplo, el índice de masa corporal), especialmente en el conjunto de datos TIPOS DE OBESIDAD.

Tabla 4.1: Resultados de clasificación promedio de SVM y NB utilizando como características n-gramas y FREGEX. Fuente: Elaboración propia.

Clasificador	OBESIDAD		TIPOS OBESIDAD		TABAQUISMO	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
SVM-FREGEX	96.84	96.86	91.27	91.39	86.16	86.37
SVM-N1	96.37	96.39	82.75	82.98	84.64	84.77
SVM-N2	94.30	94.39	88.44	88.71	$85.92^{(*)(b)}$	$86.27^{(*)(a)}$
NB-FREGEX	89.96	90.28	88.32	88.60	81.49	81.75
NB-N1	87.50	87.46	74.33	74.39	76.71	76.98
NB-N2	$89.65^{(**)(b)}$	89.62	82.27	82.65	82.84	83.12

$^{(*),(**)}$ Indica que no hubo diferencias estadísticamente significativas en comparación a SVM-FREGEX y NB-FREGEX, respectivamente ($p > 0{,}05$).$^{(a)}$ T-student. $^{(b)}$ Wilcoxon signed-rank.

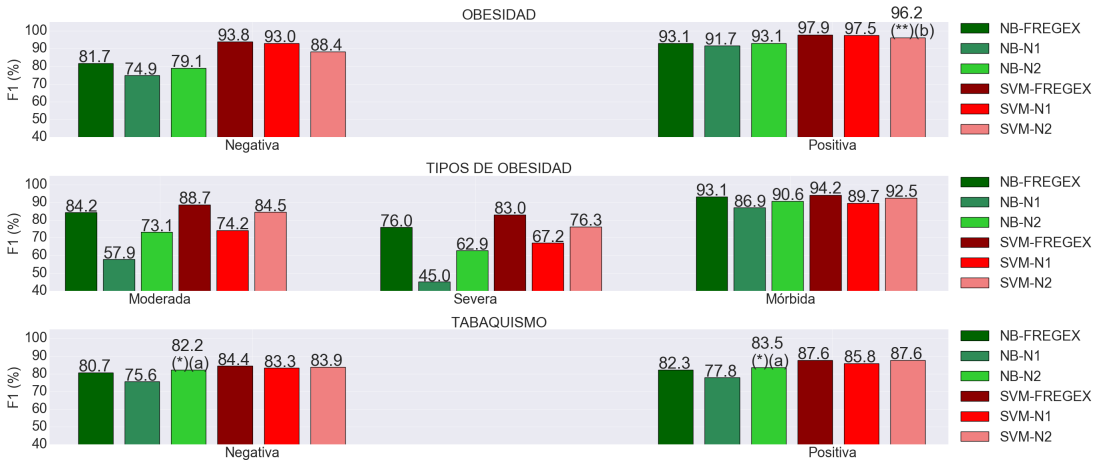

Figura 4.3: Desempeño de los clasificadores en términos de F1 (%) promedio para cada clase en cada conjunto de datos. $^{(*),(**)}$ Indica que no hubo diferencias estadísticamente significativas en comparación a SVM-FREGEX y NB-FREGEX, respectivamente ($p > 0{,}05$).$^{(a)}$ T-student. $^{(b)}$ Wilcoxon signed-rank. Fuente: Elaboración propia.

4.2.2. Curvas de aprendizaje de selección de características

La Tabla 4.2 presenta el número de características promedio extraídas mediante n-gramas y por FREGEX. Es posible observar que en todos los casos el número de características extraídas mediante FREGEX fue menor a las extraídas mediante n-gramas.

Tabla 4.2: Cantidad promedio de características extraídas mediante n-gramas y FREGEX. Fuente: Elaboración propia.

Conjunto de datos	Cantidad de características promedio		
	FREGEX	N1	N2
OBESIDAD	2558	4123	13821
TIPOS OBESIDAD	2058	4166	15272
TABAQUISMO	2672	3682	13736

La Figura 4.4 muestra las curvas de aprendizaje de los clasificadores en términos de número de características (%) y desempeño medido en términos de ACC (%) y F1 (%). En general, FREGEX permitió mejorar el desempeño de SVM y NB durante todas las curvas de aprendizaje, especialmente en el conjunto de datos TIPOS DE OBESIDAD. Además, al analizar el porcentaje mínimo de características necesarios para obtener un mismo desempeño en todos los clasificadores (ver Tablas 4.3, 4.4 y 4.5) es posible observar que FREGEX no solo mejoró el desempeño

de SVM y NB, sino que también fue más eficiente en términos de número de características que los n-gramas, permitiendo extraer patrones representativos de los textos para cada problema de clasificación.

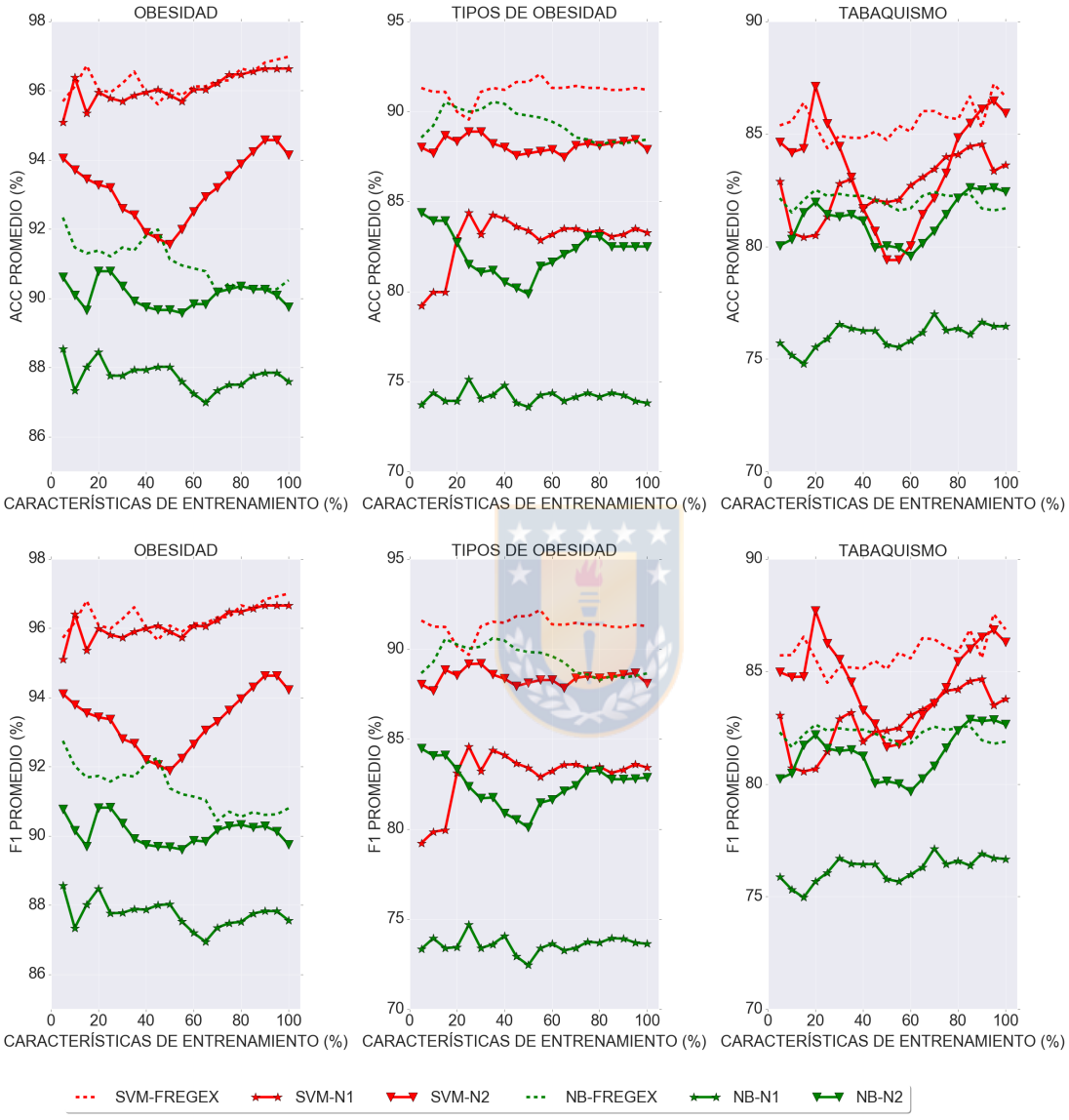

Figura 4.4: Curvas de aprendizaje de los clasificadores en términos de cantidad de características de entrenamiento (%) y desempeño en términos de ACC (%) y F1 (%). Fuente: Elaboración propia.

Tabla 4.3: Cantidad de características promedio (%) necesarias para obtener un determinado desempeño según las curvas de aprendizaje para el conjunto de datos OBESIDAD. Fuente: Elaboración propia.

Métrica (%) ≥	Cantidad mínima de características de entrenamiento (%)											
	SVM-FREGEX		SVM-N1		SVM-N2		NB-FREGEX		NB-N1		NB-N2	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
88	5	5	5	5	5	5	5	5	5	5	5	5
89	5	5	5	5	5	5	5	5	-	-	5	5
90	5	5	5	5	5	5	5	5	-	-	5	5
92	5	5	5	5	5	5	5	5	-	-	-	-
94	5	5	5	5	5	5	-	-	-	-	-	-
96	10	10	10	10	-	-	-	-	-	-	-	-
97	-	100	-	-	-	-	-	-	-	-	-	-

"-" Indica que el clasificador no alcanzó el desempeño indicado en la fila correspondiente.

Tabla 4.4: Cantidad de características promedio (%) necesarias para obtener un determinado desempeño según las curvas de aprendizaje para el conjunto de datos TIPOS DE OBESIDAD. Fuente: Elaboración propia.

Métrica (%) ≥	Cantidad mínima de características de entrenamiento (%)											
	SVM-FREGEX		SVM-N1		SVM-N2		NB-FREGEX		NB-N1		NB-N2	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
74	5	5	5	5	5	5	5	5	10	25	5	5
75	5	5	5	5	5	5	5	5	25	-	5	5
84	5	5	25	25	5	5	5	5	-	-	5	5
88	5	5	-	-	5	5	5	5	-	-	-	-
89	5	5	-	-	-	25	10	10	-	-	-	-
90	5	5	-	-	-	-	15	15	-	-	-	-
92	55	55	-	-	-	-	-	-	-	-	-	-

"-" Indica que el clasificador no alcanzó el desempeño indicado en la fila correspondiente.

Tabla 4.5: Cantidad de características promedio (%) necesarias para obtener un determinado desempeño según las curvas de aprendizaje para el conjunto de datos TABAQUISMO. Fuente: Elaboración propia.

Métrica (%) ≥	Cantidad mínima de características de entrenamiento (%)											
	SVM-FREGEX		SVM-N1		SVM-N2		NB-FREGEX		NB-N1		NB-N2	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
77	5	5	5	5	5	5	5	5	70	70	5	5
82	5	5	5	5	5	5	5	5	-	-	80	20
84	5	5	80	75	5	5	-	-	-	-	-	-
87	95	95	-	-	20	20	-	-	-	-	-	-

"-" Indica que el clasificador no alcanzó el desempeño indicado en la fila correspondiente.

4.3. Clasificación de textos: CREGEX

4.3.1. Resultados de clasificación

La Tabla 4.6 muestra los resultados de clasificación de CREGEX, NB, SVM y del clasificador basado en BERT en términos de ACC (%) y F1 (%). En todos los casos el desempeño de CREGEX fue superior a SVM y NB en todas las métricas desempeño. En cuanto a BERT, CREGEX obtuvo un mejor desempeño en todos los conjuntos de datos, pero levemente inferior en el conjunto de datos TABAQUISMO (diferencias estadísticamente no significativas, $p > 0{,}05$). Esto también se puede observar al analizar el desempeño por cada clase del problema. Según lo indicado en la Figura 4.5, el desempeño de CREGEX fue superior al resto de los clasificadores, excepto para la clase negativa del conjunto de datos TABAQUISMO. Esto indica que las expresiones regulares requieren una mayor cantidad de ejemplos de entrenamiento para capturar elementos temporales y negaciones, cuya presencia es mayor en el conjunto de datos TABAQUISMO.

Tabla 4.6: Resultados de clasificación promedio de CREGEX y el resto de los clasificadores. Fuente: Elaboración propia.

Clasificador	OBESIDAD		TIPOS OBESIDAD		TABAQUISMO	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
CREGEX	97.53	97.58	94.75	94.82	88.86	89.11
BERT	96.76	96.84	90.35	90.46	$89.22^{(*)(a)}$	$89.33^{(*)(a)}$
SVM-N1	96.37	96.39	82.75	82.98	84.64	84.77
SVM-N2	94.30	94.39	88.44	88.71	85.92	86.27
NB-N1	87.50	87.46	74.33	74.39	76.71	76.98
NB-N2	89.65	89.62	82.27	82.65	82.84	83.12

$^{(*)}$ Indica que no hubo diferencias estadísticamente significativas en comparación a CREGEX. ($p > 0{,}05$).$^{(a)}$ T-student.

Figura 4.5: Desempeño de los clasificadores en términos de F1 (%) para cada clase del problema en cada conjunto de datos. $^{(*)}$ Indica que no hubo diferencias estadísticamente significativas en comparación a CREGEX ($p > 0{,}05$). $^{(a)}$ T-student. $^{(b)}$ Wilcoxon signed-rank. Fuente: Elaboración propia.

La Figura 4.6 muestra la distribución de casos de los resultados de clasificación de CREGEX: (i) ninguna expresión regular coincide con un texto de prueba (caso 1); (ii) n_r expresiones regulares coinciden con un texto de prueba (caso 2). En todos los casos un porcentaje inferior al 5 % no coincidió con un texto de prueba (caso 1), siendo el conjunto de datos OBESIDAD el que presentó el menor porcentaje, seguido del conjunto de datos TIPOS DE OBESIDAD y TABAQUISMO. Esto puede ser explicado por el hecho que si bien el conjunto de datos OBESIDAD y TIPOS DE OBESIDAD pueden compartir terminología médica, corresponden a problemas binarios y multiclases, respectivamente. Por otro lado, el conjunto

de datos TABAQUISMO si bien es un problema binario, presenta una mayor complejidad léxica, debido a la presencia de negaciones y temporalidad del hábito tabáquico en los textos.

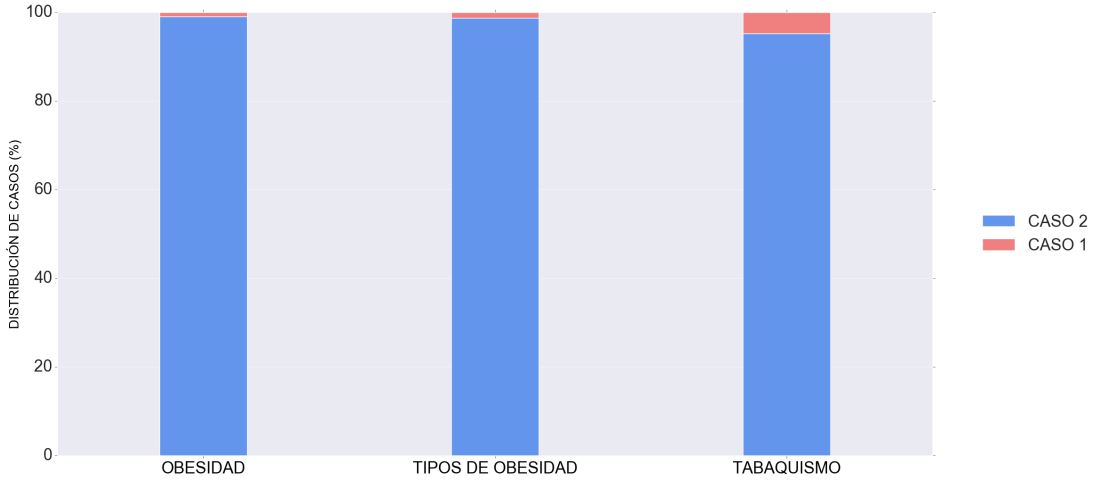

Figura 4.6: Distribución promedio de casos en la clasificación de CREGEX. Caso 1: Ninguna expresión regular coincide con un texto de prueba. Caso 2: Al menos una expresión regular coincide con un texto de prueba. Fuente: Elaboración propia.

4.3.2. Curvas de error de entrenamiento

La Figura 4.7 muestra las curvas de error de los clasificadores en términos de pérdida uno-cero (ver ecuación 4.7) durante el entrenamiento y prueba. En todos los casos el error de entrenamiento es menor al error de prueba, manteniéndose relativamente estables las curvas luego de disminuir en el primer tramo. El mismo comportamiento se puede observar en la mayoría de los casos en el resto de los clasificadores (ver Figura C1.1, C1.2, C1.3 del Anexo).

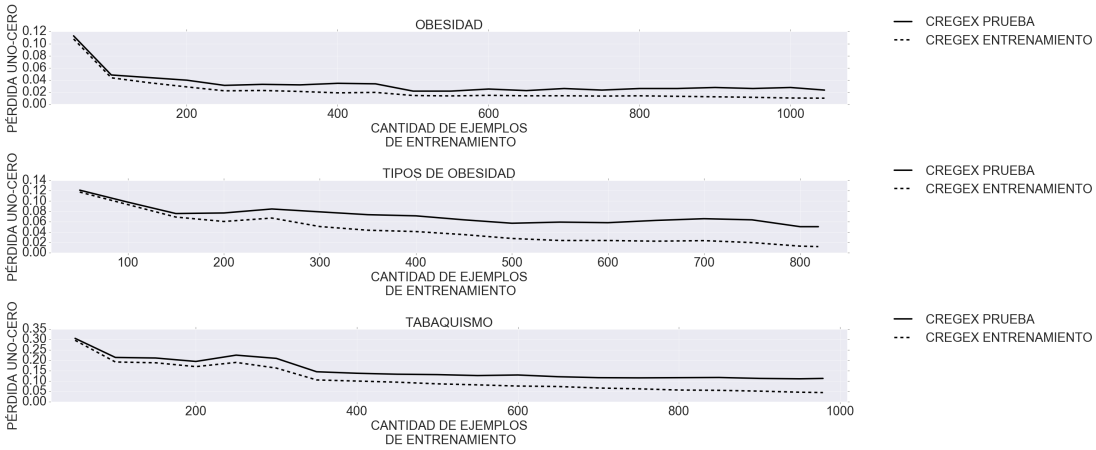

Figura 4.7: Curvas de error de CREGEX en términos de ejemplos de entrenamiento y pérdida uno-cero. Fuente: Elaboración propia.

4.4. Aprendizaje activo

4.4.1. Curvas de aprendizaje pasivo y activo

La Figura 4.8 muestra la evolución promedio de los valores de la estrategia de consulta de CREGEX en función de la cantidad de ejemplos seleccionados (%). Se observan dos fases: (i) una fase inicial o de estado transiente, donde los valores se ajustan a medida que los ejemplos son seleccionados ; y (ii) una fase de estabilización o de estado estacionario, donde los valores tienen una variabilidad baja. En este sentido, los valores de la estrategia de consulta del conjunto de datos OBESIDAD se estabilizan más rápido, seguido por el conjunto de datos TIPOS DE OBESIDAD y TABAQUISMO.

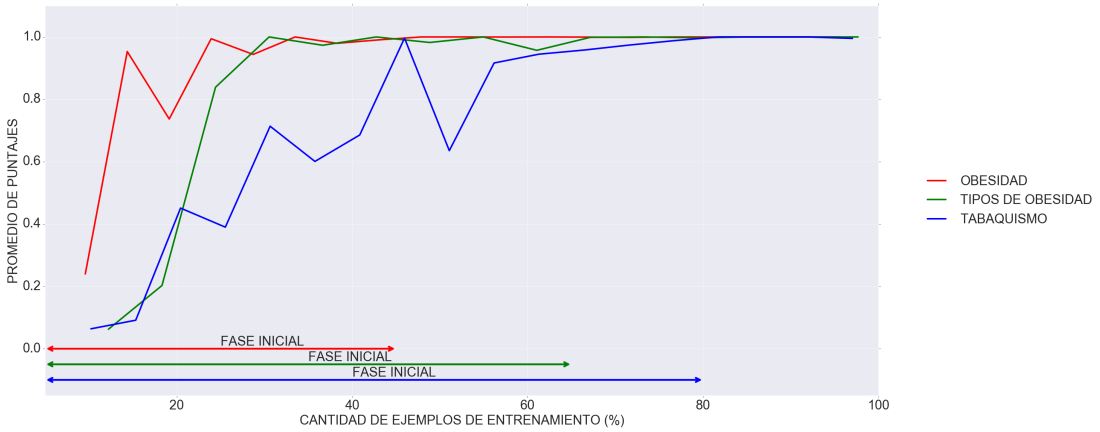

Figura 4.8: Dinámica de los valores (puntajes) de la estrategia de consulta en función del porcentaje de textos de entrenamiento seleccionados. Fuente: Elaboración propia.

La Figura 4.9 muestra las curvas de aprendizaje de CREGEX en términos de cantidad de ejemplos de entrenamiento y desempeño en términos de ACC (%) y F1 (%), según los resultados obtenidos en la combinación convexa de la ecuación (4.6). En otras palabras, se estudió el impacto del aprendizaje voraz (*greedy*) y conservativo de forma conjunta e independiente en la estrategia de consulta de la ecuación (3.2) en el proceso de AL de CREGEX. Se analizaron cuatro casos: AMB (sólo la componente voraz); DIV (sólo la componente conservativa normalizada por el valor de similitud máximo entre los ejemplos seleccionados); CMB (combinación de componentes ponderadas igualmente); y CLF (la estrategia de consulta propuesta). En términos generales se puede observar que en todos los casos la componente DIV es la que obtiene el desempeño más bajo en comparación a la estrategia de consulta propuesta CLF. Este comportamiento también se puede observar en los resultados de área bajo la curva de la Tabla. 4.7. La estrategia de consulta propuesta resultó ser más efectiva que el resto de componentes, especialmente en el conjunto de datos OBESIDAD. Además, al comparar la cantidad mínima de ejemplos de entrenamiento para alcanzar un mismo desempeño se observa que, en general, la estrategia de consulta propuesta requiere una menor cantidad de ejemplos que el resto de estrategias (ver Tabla 4.8, 4.9 y 4.10).

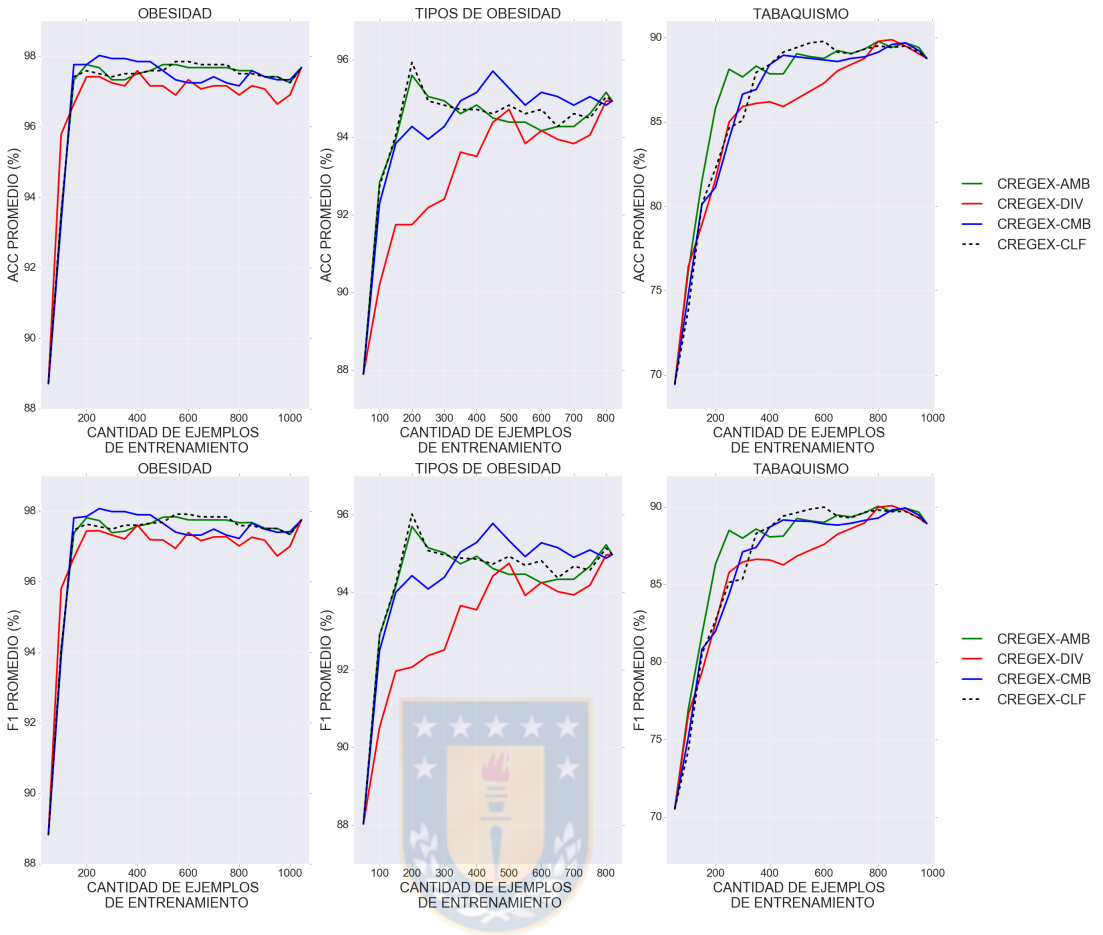

Figura 4.9: Curvas de aprendizaje activo de CREGEX en términos de cantidad de ejemplos de entrenamiento y desempeño en términos de ACC (%) y F1 (%). Fuente: Elaboración propia.

Tabla 4.7: Resultados promedio de las áreas bajo las curvas de aprendizaje de CREGEX de acuerdo a la función de combinación convexa. Fuente: Elaboración propia.

Clasificador	OBESIDAD		TIPOS OBESIDAD		TABAQUISMO	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
CREGEX-AMB	$97.14^{(*)(a)}$	$97.24^{(*)(a)}$	94.31	94.40	87.12	87.43
CREGEX-DIV	$96.85^{(*)(a)}$	$96.92^{(*)(b)}$	$93.17^{(*)(a)}$	$93.28^{(*)(a)}$	85.75	$86.13^{(*)(a)}$
CREGEX-CMB	$97.12^{(*)(a)}$	$97.21^{(*)(a)}$	$94.43^{(*)(a)}$	$94.53^{(*)(a)}$	$86.34^{(*)(a)}$	$86.68^{(*)(a)}$
CREGEX-CLF	97.15	97.25	94.41	94.52	86.56	86.87

$^{(*)}$ Indica que no hubo diferencias estadísticamente significativas en comparación a CLF ($p > 0{,}05$).$^{(a)}$ T-student. $^{(b)}$ Wilcoxon signed-rank.

Tabla 4.8: Cantidad de ejemplos de entrenamiento necesarios para obtener un determinado desempeño según la función de combinación convexa de CREGEX para el conjunto de datos OBESIDAD. Fuente: Elaboración propia.

Métrica (%) ≥	Cantidad mínima de ejemplos de entrenamiento							
	ACC (%)				F1 (%)			
	AMB	DIV	CMB	CLF	AMB	DIV	CMB	CLF
97	150	200	150	150	150	200	150	150
98	-	-	250	-	-	-	250	-

"-" Indica que el clasificador no alcanzó el desempeño indicado en la fila correspondiente.

Tabla 4.9: Cantidad de ejemplos de entrenamiento necesarios para obtener un determinado desempeño según la función de combinación convexa de CREGEX para el conjunto de datos TIPOS DE OBESIDAD. Fuente: Elaboración propia.

Métrica (%) ≥	Cantidad mínima de ejemplos de entrenamiento							
	ACC (%)				F1 (%)			
	AMB	DIV	CMB	CLF	AMB	DIV	CMB	CLF
94	200	450	200	150	150	450	200	150
95	200	-	400	200	200	-	350	200
96	-	-	-	-	-	-	-	200

"-" Indica que el clasificador no alcanzó el desempeño indicado en la fila correspondiente.

Tabla 4.10: Cantidad de ejemplos de entrenamiento necesarios para obtener un determinado desempeño según la función de combinación convexa de CREGEX para el conjunto de datos TABAQUISMO. Fuente: Elaboración propia.

Métrica (%) ≥	Cantidad mínima de ejemplos de entrenamiento							
	ACC (%)				F1 (%)			
	AMB	DIV	CMB	CLF	AMB	DIV	CMB	CLF
89	500	800	800	450	500	800	450	450
90	-	-	-	-	800	850	-	-

"-" Indica que el clasificador no alcanzó el desempeño indicado en la fila correspondiente.

Las Figuras 4.10 y 4.11 muestran las curvas de aprendizaje de los clasificadores de acuerdo al número de características y el desempeño en términos de ACC (%) y F1 (%). En cada caso se analizó un aprendizaje pasivo (PL) o una selección aleatoria y una estrategia de consulta de aprendizaje activo (CLF). En el caso de SVM se consideró además una estrategia de consulta basada en las distancias

al hiperplano y similitud coseno (CMB) [76]. Es posible observar que CREGEX tiene un mejor desempeño que el resto de los clasificadores durante las curvas de aprendizaje, especialmente en el conjunto de datos TIPOS DE OBESIDAD. Esto último también se puede observar en los resultados de área bajo la curva de la Tabla 4.11. En todos los casos el AL permitió obtener un desempeño mayor o igual al PL. En este sentido, la estrategia de consulta propuesta para CREGEX permitió obtener el mejor desempeño que el resto de las estrategias analizadas.

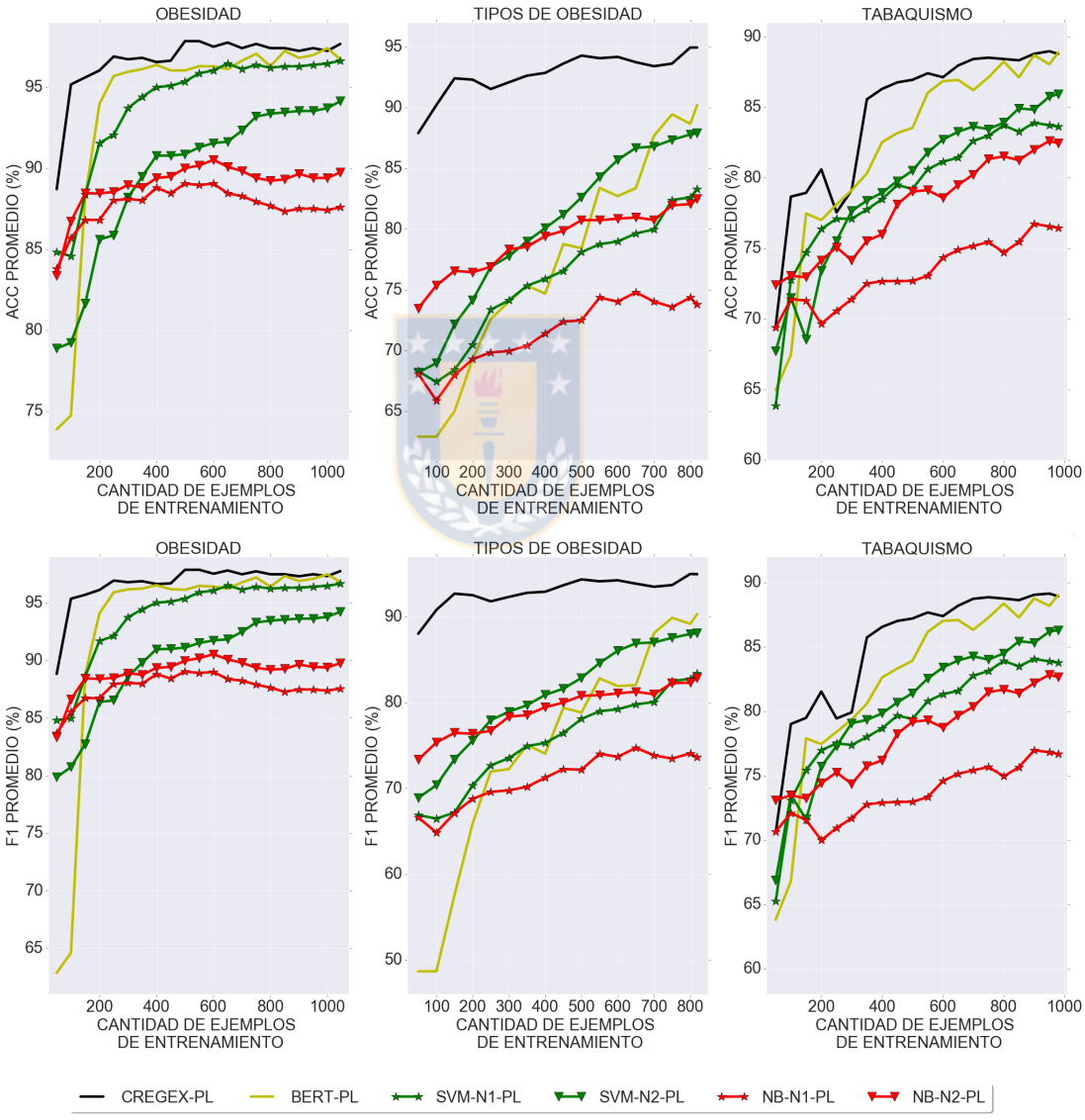

Figura 4.10: Curvas de aprendizaje pasivo de los clasificadores en términos de cantidad de ejemplos de entrenamiento y desempeño en términos de ACC (%) y F1 (%). Fuente: Elaboración propia.

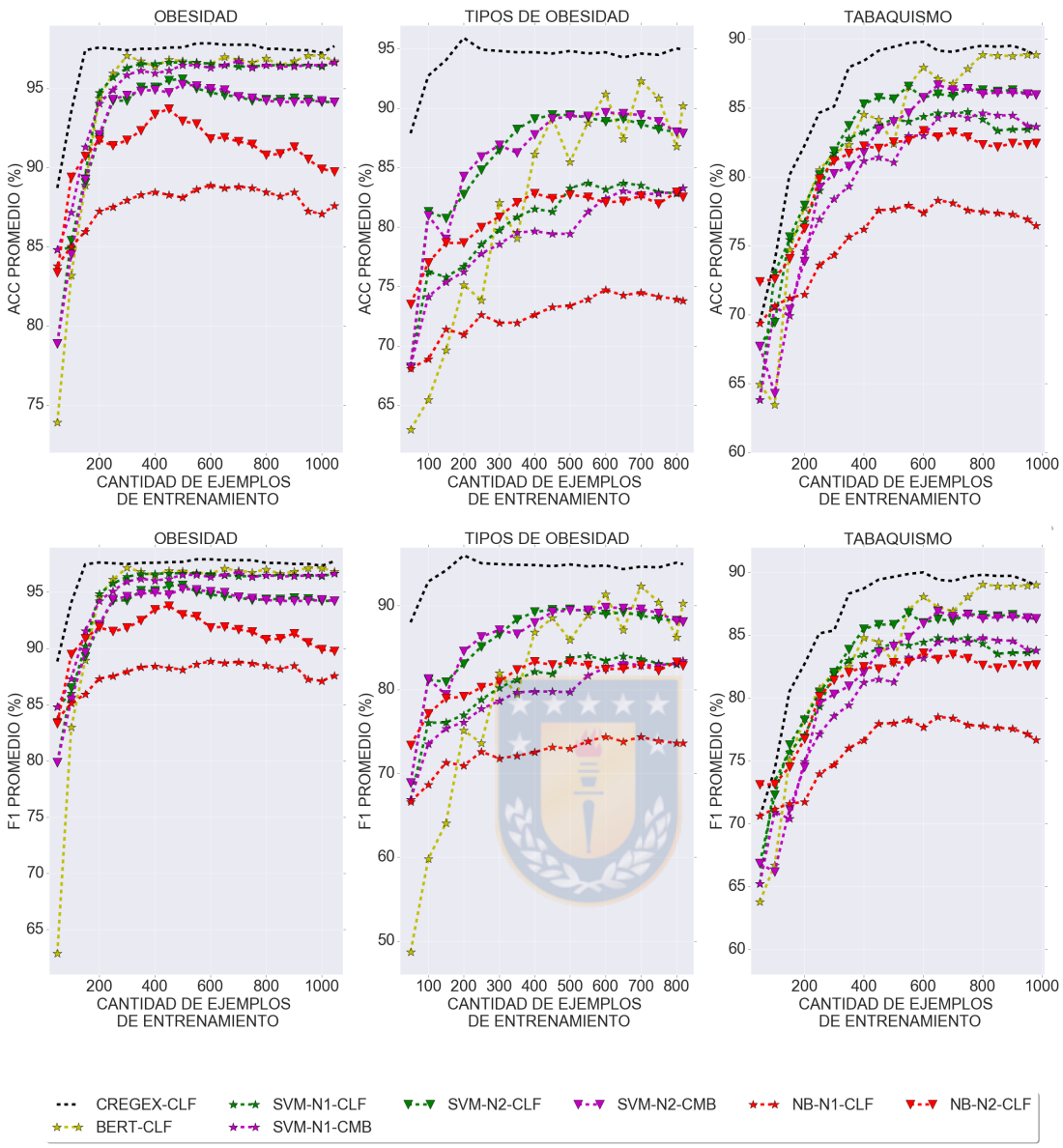

Figura 4.11: Curvas de aprendizaje activo de los clasificadores en términos de cantidad de ejemplos de entrenamiento y desempeño en términos de ACC (%) y F1 (%). Fuente: Elaboración propia.

Tabla 4.11: Resultados promedio de las áreas bajo las curvas de aprendizaje de los clasificadores en función de las diferentes estrategias de consulta. Fuente: Elaboración propia.

Clasificador	OBESIDAD		TIPOS OBESIDAD		TABAQUISMO	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
CREGEX-PL	96.82	96.88	92.95	93.12	84.91	85.37
CREGEX-CLF	$97.15^{(*)(b)}$	97.25	94.41	94.52	86.56	86.87
BERT-PL	94.28	93.74	77.68	75.78	82.67	82.83
BERT-CLF	94.97	94.81	82.49	81.49	83.16	83.55
SVM-N1-PL	94.17	94.22	76.12	75.87	79.47	79.76
SVM-N1-CLF	95.15	95.27	80.60	80.85	81.60	81.82
SVM-N1-CMB	95.16	95.22	79.44	79.49	$80.18^{(*)(a)}$	$80.40^{(*)(a)}$
SVM-N2-PL	89.83	90.19	80.59	81.28	79.75	80.80
SVM-N2-CLF	93.36	93.50	86.40	86.58	83.03	83.43
SVM-N2-CMB	93.29	93.42	86.40	86.66	$81.51^{(*)(a)}$	$81.87^{(*)(a)}$
NB-N1-PL	87.78	87.72	71.61	71.26	73.37	73.68
NB-N1-CLF	$87.78^{(*)(a)}$	$87.78^{(*)(a)}$	$72.70^{(*)(a)}$	$72.50^{(*)(a)}$	75.76	76.12
NB-N2-PL	89.12	89.08	79.21	79.31	77.97	78.18
NB-N2-CLF	91.34	91.39	81.09	81.42	80.74	81.04

$^{(*)}$ Indica que no hubo diferencias estadísticamente significativas en comparación a PL en el clasificador correspondiente ($p > 0{,}05$).$^{(a)}$ T-student. $^{(b)}$ Wilcoxon signed-rank.

Las Tablas 4.12, 4.13, 4.14 y 4.15, 4.16 y 4.17 muestran la cantidad mínima de ejemplos de entrenamiento necesarios para obtener un mismo desempeño en todos los clasificadores, según lo observado en las Figuras 4.10 y 4.11. En la mayoría de los casos, excepto en NB-N1 en el conjunto de datos OBESIDAD y TIPOS DE OBESIDAD el uso de AL permitió reducir el número de ejemplos de entrenamiento para alcanzar un mismo desempeño en términos de ACC (%) y F1 (%) en comparación al PL. En el caso de BERT, estos resultados muestran la efectividad de la estrategia de consulta utilizada para este tipo de clasificadores basados en DNNs.

Tabla 4.12: Cantidad de ejemplos de entrenamiento necesarios para obtener un determinado desempeño según las curvas de aprendizaje pasivo de los clasificadores para el conjunto de datos OBESIDAD. Fuente: Elaboración propia.

Métrica (%) ≥	Cantidad mínima de ejemplos de entrenamiento											
	CREGEX		BERT		SVM-N1		SVM-N2		NB-N1		NB-N2	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
89	100	100	200	200	200	200	350	350	500	500	400	400
90	100	100	200	200	200	200	400	400	-	-	500	550
94	100	100	250	200	350	350	1045	1045	-	-	-	-
96	200	200	350	300	600	600	-	-	-	-	-	-
97	500	500	750	750	-	-	-	-	-	-	-	-

"-" Indica que el clasificador no alcanzó el desempeño indicado en la fila correspondiente.

Tabla 4.13: Cantidad de ejemplos de entrenamiento necesarios para obtener un determinado desempeño según las curvas de aprendizaje activo de los clasificadores para el conjunto de datos OBESIDAD. Fuente: Elaboración propia.

Métrica	Cantidad mínima de ejemplos de entrenamiento															
(%) ≥	CREGEX		BERT		SVM-N1		SVM-N1$^{(*)}$		SVM-N2		SVM-N2$^{(*)}$		NB-N1		NB-N2	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
88	50	50	150	150	150	150	150	150	150	150	150	150	350	350	100	100
93	100	100	200	200	200	200	200	200	250	250	250	250	-	-	400	400
95	150	150	250	250	250	250	300	250	350	350	500	500	-	-	-	-
96	150	150	300	250	300	300	350	350	-	-	-	-	-	-	-	-
97	150	150	300	300	-	-	-	-	-	-	-	-	-	-	-	-

"-" Indica que el clasificador no alcanzó el desempeño indicado en la fila correspondiente.
$^{(*)}$ Combinación de distancia al hiperplano y similitud coseno (CMB).

Tabla 4.14: Cantidad de ejemplos de entrenamiento necesarios para obtener un determinado desempeño según las curvas de aprendizaje pasivo de los clasificadores para el conjunto de datos TIPOS DE OBESIDAD. Fuente: Elaboración propia.

Métrica (%) ≥	Cantidad mínima de ejemplos de entrenamiento											
	CREGEX		BERT		SVM-N1		SVM-N2		NB-N1		NB-N2	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
74	50	50	300	350	300	350	200	200	550	550	100	100
82	50	50	550	550	750	750	500	500	-	-	800	750
83	50	50	550	700	819	819	550	550	-	-	-	-
87	50	50	700	700	-	-	750	700	-	-	-	-
88	100	50	750	700	-	-	-	800	-	-	-	-
90	100	100	819	819	-	-	-	-	-	-	-	-
94	500	500	-	-	-	-	-	-	-	-	-	-

"-" Indica que el clasificador no alcanzó el desempeño indicado en la fila correspondiente.

Tabla 4.15: Cantidad de ejemplos de entrenamiento necesarios para obtener un determinado desempeño según las curvas de aprendizaje activo de los clasificadores para el conjunto de datos TIPOS DE OBESIDAD. Fuente: Elaboración propia.

Métrica	Cantidad mínima de ejemplos de entrenamiento															
(%) ≥	CREGEX		BERT		SVM-N1		SVM-N1(*)		SVM-N2		SVM-N2(*)		NB-N1		NB-N2	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
74	50	50	200	200	100	100	100	150	100	100	100	100	600	600	100	100
82	50	50	300	400	500	400	600	600	200	200	200	200	-	-	350	350
83	50	50	400	400	500	500	650	650	250	200	200	200	-	-	-	400
84	50	50	400	400	-	550	-	-	250	250	200	200	-	-	-	-
89	100	100	450	600	-	-	-	-	400	400	450	450	-	-	-	-
92	100	100	700	700	-	-	-	-	-	-	-	-	-	-	-	-
95	200	200	-	-	-	-	-	-	-	-	-	-	-	-	-	-
96	-	200	-	-	-	-	-	-	-	-	-	-	-	-	-	-

"-" Indica que el clasificador no alcanzó el desempeño indicado en la fila correspondiente.
(*) Combinación de distancia al hiperplano y similitud coseno (CMB).

Tabla 4.16: Cantidad de ejemplos de entrenamiento necesarios para obtener un determinado desempeño según las curvas de aprendizaje pasivo de los clasificadores para el conjunto de datos TABAQUISMO. Fuente: Elaboración propia.

Métrica (%) ≥	Cantidad mínima de ejemplos de entrenamiento											
	CREGEX		BERT		SVM-N1		SVM-N2		NB-N1		NB-N2	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
76	100	100	150	150	200	200	300	250	900	900	450	400
82	350	350	400	400	700	700	600	550	-	-	950	900
83	350	350	450	450	800	750	650	600	-	-	-	-
84	350	350	550	550	-	900	850	700	-	-	-	-
85	350	350	550	550	-	-	950	850	-	-	-	-
86	400	400	550	550	-	-	-	950	-	-	-	-
88	700	650	800	800	-	-	-	-	-	-	-	-
89	-	900	-	979	-	-	-	-	-	-	-	-

"-" Indica que el clasificador no alcanzó el desempeño indicado en la fila correspondiente.

Tabla 4.17: Cantidad de ejemplos de entrenamiento necesarios para obtener un determinado desempeño según las curvas de aprendizaje activo de los clasificadores para el conjunto de datos TABAQUISMO. Fuente: Elaboración propia.

Métrica (%) ≥	Cantidad mínima de ejemplos de entrenamiento															
	CREGEX		BERT		SVM-N1		SVM-N1(*)		SVM-N2		SVM-N2(*)		NB-N1		NB-N2	
	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1	ACC	F1
78	150	150	250	200	250	250	300	300	250	200	250	250	650	550	250	250
83	250	250	400	400	400	350	650	550	350	350	450	450	-	-	600	600
84	250	250	400	400	500	500	650	650	400	400	550	500	-	-	-	-
86	350	350	550	550	-	-	-	-	550	550	650	650	-	-	-	-
88	400	350	800	600	-	-	-	-	-	-	-	-	-	-	-	-
89	450	450	-	800	-	-	-	-	-	-	-	-	-	-	-	-

"-" Indica que el clasificador no alcanzó el desempeño indicado en la fila correspondiente.
(*) Combinación de distancia al hiperplano y similitud coseno (CMB).

4.4.2. Criterio de detención

La Figura 4.12 muestra el criterio de detención aplicado en el proceso de AL de CREGEX utilizando el método de las varianzas de los valores de la estrategia de consulta. Es posible observar que los valores de varianza de la estrategia de consulta (abajo) siguen el valor máximo de las curvas de aprendizaje (ver Figuras D1.1, D1.2 y D1.3 para el resto de los clasificadores). Finalmente, se analiza el impacto del criterio de detención en el desempeño de los clasificadores

y la cantidad de ejemplos de entrenamiento utilizados. La Tabla 4.18 muestra la reducción promedio en términos de ACC (Δ_{ACC}) y F1 (Δ_{F1}), respecto al valor máximo posible de las curvas de aprendizaje. La Tabla también muestra el porcentaje de ejemplos de entrenamiento ($\%_X$) utilizados al momento de detener el proceso de AL. Se puede observar que en todos los casos al detener el proceso de AL, los clasificadores utilizaron entre un 32 % y un 80 % del total de ejemplos de entrenamiento, con una reducción promedio inferior al 7 % en ambas métricas. En el caso de CREGEX, el criterio de detención detuvo el proceso de AL utilizando entre un 30 % a 50 % del total de ejemplos de entrenamiento con una reducción de desempeño inferior al 2 % respecto al valor máximo.

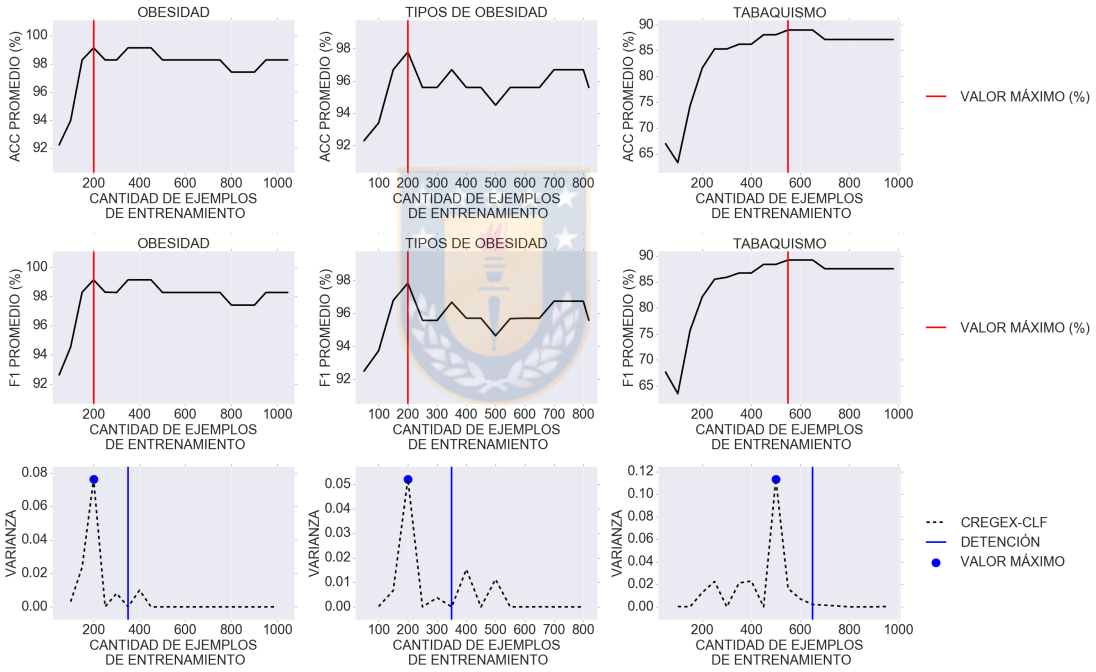

Figura 4.12: Ejemplo del criterio de detención para el proceso de aprendizaje activo de CREGEX. Fuente: Elaboración propia.

Tabla 4.18: Resultados del criterio de detención de acuerdo al método de varianza aplicado a los valores de las estrategias de consulta de los clasificadores. Fuente: Elaboración propia.

Classifier	OBESIDAD			TIPOS DE OBESIDAD			TABAQUISMO		
	Δ_{ACC}	Δ_{F1}	$\%_X$	Δ_{ACC}	Δ_{F1}	$\%_X$	Δ_{ACC}	Δ_{F1}	$\%_X$
CREGEX-CLF	1.03	0.98	32.06	1.65	1.58	44.62	1.84	1.78	50.09
BERT-CLF	1.21	1.19	43.07	6.05	6.24	73.34	4.41	4.40	78.71
SVM-N1-CLF	0.52	0.51	46.89	1.10	1.12	80.67	1.10	1.20	65.93
SVM-N1-CMB	1.29	1.26	40.67	3.96	3.72	76.40	2.48	2.44	66.95
SVM-N2-CLF	0.52	0.51	47.85	2.09	2.15	70.90	2.30	2.32	66.95
SVM-N2-CMB	1.03	1.02	41.15	0.99	1.04	75.78	1.38	1.38	68.99
NB-N1-CLF	2.15	2.20	53.59	2.86	3.07	69.06	2.12	2.09	69.51
NB-N2-CLF	1.38	1.35	46.42	2.53	2.49	66.62	2.58	2.55	55.71

Δ_{ACC} Indica una reducción en términos de ACC (%) respecto al valor máximo.
Δ_{F1} Indica una reducción en términos de F1 (%) respecto al valor máximo.
$\%_X$ Indica el número de ejemplos de entrenamiento (%) utilizados para alcanzar el determinado ACC y F1.

Capítulo 5

Conclusiones y trabajo futuro

5.1. Sumario

En esta tesis se propuso el uso de algoritmos basados en expresiones regulares para la clasificación de textos biomédicos, como una alternativa a los algoritmos de clasificación más utilizados en esta área. Se desarrolló un método de extracción de características, denominado FREGEX, como una alternativa al tradicional uso de n-gramas, el cual permite extraer automáticamente características en la forma de expresiones regulares. Las expresiones regulares fueron utilizadas para construir un espacio de características que le permitieron a CREGEX definir una función de decisión para la clasificación de textos biomédicos. En comparación a los clasificadores más utilizados, CREGEX no requiere una representación matricial de los datos, facilitando un análisis posterior a nivel de lenguaje natural de las expresiones regulares utilizadas. Por otro lado, se formuló matemáticamente una estrategia de consulta que permitió identificar los ejemplos considerados más informativos para CREGEX. Esta estrategia de consulta conforma el conjunto de entrenamiento desde un conjunto no etiquetado considerando un equilibrio entre un enfoque de aprendizaje tipo voraz o *greedy*, según los valores de precisión de las expresiones regulares, y un aprendizaje conservador, inducido por la métrica de diversidad del algoritmo de SW. Como criterio de detención del AL se consideró una reducción sostenida en la varianza de los valores de las estrategias de consulta de los clasificadores.

5.2. Conclusión

De acuerdo a la investigación realizada en este trabajo de tesis, se demuestra la hipótesis que si se diseña una estrategia de consulta de aprendizaje activo junto con un criterio de detención que permita determinar cuáles son los ejemplos más informativos en un conjunto de datos no etiquetado, identificando los casos de ambigüedad, se mejoraría el desempeño de un clasificador de textos biomédicos basado en la generación automática de expresiones regulares, con un desempeño superior al 85 % en términos de área bajo la curva de aprendizaje y utilizando menos del 50 % de la cantidad total de ejemplos de entrenamiento en comparación al método de aprendizaje pasivo.

Los resultados indican que las características extraídas mediante FREGEX, en general, mejoraron el desempeño de SVM y NB (ver Tabla 4.1). Además, la principal ventaja en el uso de FREGEX en comparación al uso de n-gramas, es que solo se extraen elementos representativos de cada conjunto de datos, reduciendo la cantidad de elementos considerados ruidosos. Ésto fue posible gracias al agrupamiento de palabras similares, a la normalización de los números presentes en los textos y al uso de los algoritmos de alineación de NW y SW, los cuales permitieron capturar las variantes léxicas de las palabras en términos de género y número gramatical. De esta forma, se responde afirmativamente la pregunta de investigación: “Para un determinado problema de clasificación de textos biomédicos, ¿Puede un algoritmo basado en expresiones regulares capturar las variantes léxicas de los términos representativos de cada clase del problema?”.

Respecto a CREGEX, los resultados indican que este clasificador obtuvo un mejor desempeño que SVM y NB en todas las métricas de desempeño (ver Tabla 4.6). Esto fue posible gracias a la capacidad que tuvieron las expresiones regulares para representar patrones complejos desde los textos, incluyendo las variantes léxicas de las palabras y los atributos numéricos (ver Figura 3.3). Además, CREGEX obtuvo un mejor desempeño que BERT en todos los casos, excepto en el conjunto de datos TABAQUISMO (aunque las diferencias no fueron estadísticamente significativas, $p > 0{,}05$). Esto podría ser explicado por dos razones. En primer lugar, los conjuntos de datos OBESIDAD y TIPOS DE OBESIDAD contienen más atributos numéricos (información antropométrica) que el conjunto de datos TABAQUISMO. En este sentido, se ha demostrado que BERT puede presentar problemas para representar

correctamente los atributos numéricos en los textos. Lo contrario ocurre con las expresiones regulares, las cuales han sido muy utilizadas en textos biomédicos con presencia de atributos numéricos [20,52,53]. En segundo lugar, el conjunto de datos TABAQUISMO presenta mayores elementos de temporalidad y negaciones en los textos, y las expresiones regulares necesitan una mayor cantidad de ejemplos para capturar dicha información. Sin embargo, la complejidad de CREGEX es menor a BERT, al comparar la cantidad de parámetros de este modelo basado en DNNs, el cual fue entrenado en una gran cantidad de textos utilizando BooksCorpus (800M palabras) y Wikipedia (2500M palabras) con 110M de parámetros [1]. De esta forma se responde afirmativamente la pregunta de investigación: "Para un determinado problema de clasificación de textos biomédicos, ¿Puede un algoritmo basado en expresiones regulares tener un mejor desempeño que los algoritmos de clasificación más utilizados?"

Los resultados de las curvas de error indican que CREGEX no presenta problemas de generalización, evitando problemas de *underfitting* (subajuste) y *overfitting* (sobreajuste). Esto queda demostrado por el hecho que la curva de error de prueba se mantiene relativamente estable una vez que disminuye junto con la curva del error de entrenamiento (ver Figura 4.7).

Por otro lado, la distribución de casos de clasificación de CREGEX (ver Figura 4.6) muestran que un porcentaje inferior al 5 % de las expresiones regulares no coincidieron con un texto de prueba. Esto indica que el método propuesto generó suficientes expresiones regulares para representar a los textos biomédicos utilizando solo el conjunto de entrenamiento. Esto presenta una ventaja respecto a otros modelos más complejos como es el caso de BERT, el cual utilizó una gran colección de documentos para ser entrenado.

Respecto al proceso de AL, en todos los casos, el área bajo la curva de aprendizaje fue superior a la de PL (ver Tabla 4.11). Además, en la mayoría de los casos, el AL permitió reducir eficientemente el número de ejemplos de entrenamiento en todos los conjuntos de datos en comparación al PL. De esta forma, se responde afirmativamente la pregunta de investigación: "Para un determinado algoritmo de clasificación de textos biomédicos basado en expresiones regulares, ¿Puede el aprendizaje activo reducir efectivamente la cantidad de ejemplos de entrenamiento necesarios para obtener un determinado desempeño en comparación al aprendizaje pasivo?". Más precisamente, el AL junto con el criterio de detención permitió usar,

en promedio, solo entre un 32 % a un 81 % del total de ejemplos de entrenamiento sin pérdidas significativas de desempeño en términos de ACC (%) y F1 (%). En este sentido, la estrategia de consulta propuesta para CREGEX resultó ser la más eficiente (ver Tabla 4.18). De esta forma, se responde afirmativamente la pregunta de investigación: "Para un determinado algoritmo de clasificación de textos biomédicos basado en expresiones regulares, ¿Puede el aprendizaje activo mejorar el desempeño en comparación a otras estrategias de consulta de selección de los ejemplos más informativos?".

5.3. Trabajo futuro

En este trabajo de tesis solo se utilizaron textos biomédicos en español. Como trabajo futuro se plantea la posibilidad de aplicar los algoritmos en base a expresiones regulares a textos biomédicos de otros idiomas o a textos de diferente temática. Del mismo modo, se plantea la posibilidad de implementar algoritmos para la extracción de información, debido a la capacidad que demostraron las expresiones regulares para representar patrones complejos de texto.

Por otro lado, se pretende mejorar algunos aspectos específicos del presente trabajo, particularmente, el agrupamiento de palabras similares para capturar las variantes léxicas y el filtrado automático de expresiones regulares con algún método de selección de características. Además, se plantea la posibilidad de incorporar mayor diversidad a las expresiones regulares, por ejemplo, incorporando algoritmos genéticos durante la etapa de generación automática. Otro de los aspectos a mejorar es reducir el tiempo de entrenamiento de los algoritmos propuestos debido a la cantidad de alineaciones que se requieren para extraer los *tokens* representativos desde los textos (ver Figura E1.1 del Anexo).

Finalmente, se comparará el desempeño de CREGEX con otros modelos de lenguaje pre-entrenados debido al auge en el desarrollo de este tipo de modelos en el último tiempo. Del mismo modo, se estudiará el impacto del AL en este tipo de modelos, tal como se hizo con BERT en el presente trabajo.

Bibliografía

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[2] H. Alemdar, T. Van Kasteren, and C. Ersoy, "Active learning with uncertainty sampling for large scale activity recognition in smart homes," *Journal of Ambient Intelligence and Smart Environments*, vol. 9, no. 2, pp. 209–223, 2017.

[3] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, X. Chen, and X. Wang, "A survey of deep active learning," *arXiv preprint arXiv:2009.00236*, 2020.

[4] L. Ornella, "Códigos correctores de error en problemas de clasificación multiclase de datos de marcadores moleculares," Ph.D. dissertation, Tesis de Doctorado, Facultad de Ciencias Exactas, Ingeniería y Agrimensura . . . , 2010.

[5] M. Munikar, S. Shakya, and A. Shrestha, "Fine-grained sentiment classification using bert," in *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, vol. 1. IEEE, 2019, pp. 1–5.

[6] V. K. Chauhan, K. Dahiya, and A. Sharma, "Problem formulations and solvers in linear svm: a review," *Artificial Intelligence Review*, vol. 52, no. 2, pp. 803–855, 2019.

[7] S. Kaisler, F. Armour, J. A. Espinosa, and W. Money, "Big data: Issues and challenges moving forward," 2013.

[8] M. Liu, L. Pan, and S. Liu, "To transfer or not: An online cost optimization algorithm for using two-tier storage-as-a-service clouds," *IEEE Access*, vol. 7, pp. 94 263–94 275, 2019.

[9] H. Cho, W. Choi, and H. Lee, "A method for named entity normalization in biomedical articles: application to diseases and plants," *BMC bioinformatics*, vol. 18, no. 1, p. 451, 2017.

[10] Z. Yang, M. Dehmer, O. Yli-Harja, and F. Emmert-Streib, "Combining deep learning with token selection for patient phenotyping from electronic health records," *Scientific Reports*, vol. 10, no. 1, pp. 1–18, 2020.

[11] J. Adeva, J. Atxa, M. Carrillo, and E. Zengotitabengoa, "Automatic text classification to support systematic reviews in medicine," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1498–1508, 2014.

[12] R. Sinoara, J. Camacho-Collados, R. Rossi, R. Navigli, and S. Rezende, "Knowledge-enhanced document embeddings for text classification," *Knowledge-Based Systems*, vol. 163, pp. 955–971, 2018.

[13] X. Deng, Y. Li, J. Weng, and J. Zhang, "Feature selection for text classification: A review," *Multimedia Tools and Applications*, vol. 78, no. 3, p. 3797–3816, 2019.

[14] S. Hassan, M. Rafi, and M. Shaikh, "Comparing svm and naive bayes classifiers for text categorization with wikitology as knowledge enrichment," pp. 31–34, 2011.

[15] S. Arora, M. Khodak, N. Saunshi, and K. Vodrahalli, "A compressed sensing view of unsupervised text embeddings, bag-of-n-grams, and lstms," in *International Conference on Learning Representations*, 2018.

[16] P. Jin, Y. Zhang, X. Chen, and Y. Xia, "Bag-of-embeddings for text classification." in *IJCAI*, vol. 16, 2016, pp. 2824–2830.

[17] D. Gissin and S. Shalev-Shwartz, "Discriminative active learning," *arXiv preprint arXiv:1907.06347*, 2019.

[18] Y. Si, J. Wang, H. Xu, and K. Roberts, "Enhancing clinical concept extraction with contextual embeddings," *Journal of the American Medical Informatics Association*, vol. 26, no. 11, pp. 1297–1304, 2019.

[19] W.-C. Chang, H.-F. Yu, K. Zhong, Y. Yang, and I. Dhillon, "Taming pretrained transformers for extreme multi-label text classification," 2019.

[20] D. D. A. Bui and Q. Zeng-Treitler, "Learning regular expressions for clinical text classification," *Journal of the American Medical Informatics Association*, vol. 21, no. 5, pp. 850–857, 2015.

[21] M. Cui, R. Bai, Z. Lu, X. Li, U. Aickelin, and P. Ge, "Regular expression based medical text classification using constructive heuristic approach," *IEEE Access*, vol. 7, pp. 147 892–147 904, 2019.

[22] Z. Zhong, J. Guo, W. Yang, T. Xie, J.-G. Lou, T. Liu, and D. Zhang, "Generating regular expressions from natural language specifications: Are we there yet?" in *Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[23] A. Sarker and G. Gonzalez, "Portable automatic text classification for adverse drug reaction detection via multi-corpus training," *Journal of biomedical informatics*, vol. 53, pp. 196–207, 2015.

[24] B. C. Wallace, K. Small, C. E. Brodley, and T. A. Trikalinos, "Active learning for biomedical citation screening," in *Proceedings of the 16th ACM SIGKDD*

international conference on Knowledge discovery and data mining, 2010, pp. 173–182.

[25] D. Yoo and I. S. Kweon, "Learning loss for active learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 93–102.

[26] N. Ostapuk, J. Yang, and P. Cudré-Mauroux, "Activelink: deep active learning for link prediction in knowledge graphs," in *The World Wide Web Conference*, 2019, pp. 1398–1408.

[27] Y. Li, R. Krishnamurthy, S. Raghavan, and S. Vaithyanathan, "Regular expression learning for information extraction," pp. 21–30, 2008.

[28] A. Bartoli, A. De Lorenzo, E. Medvet, and F. Tarlao, "Active learning of regular expressions for entity extraction," *IEEE transactions on cybernetics*, vol. 48, no. 3, pp. 1067–1080, 2017.

[29] R. Hu, "Active learning for text classification," PhD's thesis, School of Computing, Technological University Dublin, Irlanda, 2011.

[30] A. Névéol, H. Dalianis, S. Velupillai, G. Savova, and P. Zweigenbaum, "Clinical natural language processing in languages other than english: opportunities and challenges," *Journal of biomedical semantics*, vol. 9, no. 1, p. 12, 2018.

[31] M. Mitrofan and D. Tufiş, "Bioro: The biomedical corpus for the romanian language," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[32] L. Bouscarrat, A. Bonnefoy, C. Capponi, and C. Ramisch, "Multilingual enrichment of disease biomedical ontologies," *arXiv preprint arXiv:2004.03181*, 2020.

[33] B. Li, T. Liu, Z. Zhao, P. Wang, and X. Du, "Neural bag-of-ngrams," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[34] H. Altınçay and Z. Erenel, "Using the absolute difference of term occurrence probabilities in binary text categorization," *Applied Intelligence*, vol. 36, no. 1, pp. 148–160, 2012.

[35] M. Chen, K. Q. Weinberger, F. Sha *et al.*, "An alternative text representation to tf-idf and bag-of-words," *arXiv preprint arXiv:1301.6770*, 2013.

[36] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *International conference on machine learning*, 2014, pp. 1188–1196.

[37] R. Zhao and K. Mao, "Fuzzy bag-of-words model for document representation," *IEEE transactions on fuzzy systems*, vol. 26, no. 2, pp. 794–804, 2017.

[38] Z. Li, Z. Xiong, Y. Zhang, C. Liu, and K. Li, "Fast text categorization using concise semantic analysis," *Pattern Recognition Letters*, vol. 32, no. 3, pp. 441–448, 2011.

[39] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[40] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[41] S. Mitra and M. Jenamani, "Hybrid improved document-level embedding (hide)," *arXiv preprint arXiv:2006.01203*, 2020.

[42] S. Silvestri, F. Gargiulo, and M. Ciampi, "Improving biomedical information extraction with word embeddings trained on closed-domain corpora," in *2019 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2019, pp. 1129–1134.

[43] Q. Jin, B. Dhingra, W. W. Cohen, and X. Lu, "Probing biomedical embeddings from language models," *arXiv preprint arXiv:1904.02181*, 2019.

[44] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, "Biobert: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[45] Y. A. Alhaj, J. Xiang, D. Zhao, M. A. Al-Qaness, M. Abd Elaziz, and A. Dahou, "A study of the effects of stemming strategies on arabic document classification," *IEEE Access*, vol. 7, pp. 32 664–32 671, 2019.

[46] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, "Text classification algorithms: A survey," *Information*, vol. 10, no. 4, p. 150, 2019.

[47] W. Yin, K. Kann, M. Yu, and H. Schütze, "Comparative study of cnn and rnn for natural language processing," *arXiv preprint arXiv:1702.01923*, 2017.

[48] C. Li, G. Zhan, and Z. Li, "News text classification based on improved bi-lstm-cnn," in *2018 9th International Conference on Information Technology in Medicine and Education (ITME)*. IEEE, 2018, pp. 890–893.

[49] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune bert for text classification?" in *China National Conference on Chinese Computational Linguistics*. Springer, 2019, pp. 194–206.

[50] L. Akhtyamova, "Named entity recognition in spanish biomedical literature: Short review and bert model," in *2020 26th Conference of Open Innovations Association (FRUCT)*. IEEE, 2020, pp. 1–7.

[51] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," *arXiv preprint arXiv:1902.00751*, 2019.

[52] M. A. Murtaugh, B. S. Gibson, D. Redd, and Q. Zeng-Treitler, "Regular expression-based learning to extract bodyweight values from clinical notes," *Journal of biomedical informatics*, vol. 54, pp. 186–190, 2015.

[53] E. Wallace, Y. Wang, S. Li, S. Singh, and M. Gardner, "Do nlp models know numbers? probing numeracy in embeddings," *arXiv preprint arXiv:1909.07940*, 2019.

[54] D. Denis, "High-performance regular expression matching with parabix and llvm," Master's thesis, School of Computing Science, Simon Fraser University, Canada, 2014.

[55] V. Cotik, V. Stricker, J. Vivaldi, and H. Rodríguez Hontoria, "Syntactic methods for negation detection in radiology reports in spanish," in *Proceedings of the 15th Workshop on Biomedical Natural Language Processing, BioNLP 2016: Berlin, Germany, August 12, 2016*. Association for Computational Linguistics, 2016, pp. 156–165.

[56] A. Rosier, A. Burgun, and P. Mabo, "Using regular expressions to extract information on pacemaker implantation procedures from clinical reports," in *AMIA Annual Symposium Proceedings*, vol. 2008. American Medical Informatics Association, 2008, p. 81.

[57] Y. Kang and M. Kayaalp, "Extracting laboratory test information from biomedical text," *Journal of Pathology Informatics*, vol. 4, no. 1, p. 23, 2013.

[58] D. D. A. Bui, G. Del Fiol, J. F. Hurdle, and S. Jonnalagadda, "Extractive text summarization system to aid data extraction from full text in systematic review development," *Journal of biomedical informatics*, vol. 64, pp. 265–272, 2016.

[59] N. Milosevic, C. Gregson, R. Hernandez, and G. Nenadic, "A framework for information extraction from tables in biomedical literature," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 22, no. 1, pp. 55–78, feb 2019.

[60] A. Bartoli, G. Davanzo, A. D. Lorenzo, and E. S. E. Medvet, "Automatic synthesis of regular expressions from examples," *IEEE Computer Society*, vol. 47, no. 12, pp. 72–80, 2013.

[61] R. Babbar and N. Singh, "Clustering based approach to learning regular expressions over large alphabet for noisy unstructured text," in *Proceedings of the fourth workshop on Analytics for noisy unstructured text data*, 2010, pp. 43–50.

[62] K. Murthy, P. Deepak, and P. M. Deshpande, "Improving recall of regular expressions for information extraction," in *International Conference on Web Information Systems Engineering*. Springer, 2012, pp. 455–467.

[63] M. Shahbaz, P. McMinn, and M. Stevenson, "Automatic generation of valid and invalid test data for string validation routines using web searches and

regular expressions," *Science of Computer Programming*, vol. 97, pp. 405–425, 2015.

[64] P. Arcaini, A. Gargantini, and E. Riccobene, "Fault-based test generation for regular expressions by mutation," *Software Testing, Verification and Reliability*, vol. 29, no. 1-2, p. e1664, 2019.

[65] T. Wu and W. Pottenger, "A semi-supervised active learning algorithm for information extraction from textual data," *Journal of the Association for Information Science and Technology*, vol. 56, no. 3, pp. 258–271, 2005.

[66] F. Brauer, R. Rieger, A. Mocan, and W. M. Barczynski, "Enabling information extraction by inference of regular expressions from sample entities," in *Proceedings of the 20th ACM international conference on Information and knowledge management*. ACM, 2011, pp. 1285–1294.

[67] A. Cetinkaya, "Regular expression generation through grammatical evolution," in *Proceedings of the 9th annual conference companion on Genetic and evolutionary computation*. ACM, 2007, pp. 2643–2646.

[68] A. Bartoli, A. De Lorenzo, E. Medvet, and F. Tarlao, "Automatic search-and-replace from examples with coevolutionary genetic programming," *IEEE transactions on cybernetics*, 2019.

[69] P. Wang, G. R. Bai, and K. T. Stolee, "Exploring regular expression evolution," in *2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 2019, pp. 502–513.

[70] R. Chhatwal, N. Huber-Fliflet, R. Keeling, J. Zhang, and H. Zhao, "Empirical evaluations of active learning strategies in legal document review," in *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2017, pp. 1428–1437.

[71] R. L. Figueroa, Q. Zeng-Treitler, L. H. Ngo, S. Goryachev, and E. P. Wiechmann, "Active learning for clinical text classification: is it better than random sampling?" *Journal of the American Medical Informatics Association*, vol. 19, no. 5, pp. 809–816, 2012.

[72] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *SIGIR'94*. Springer, 1994, pp. 3–12.

[73] B. Settles, "Active learning literature survey," University of Wisconsin-Madison Department of Computer Sciences, Tech. Rep., 2009.

[74] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Proceedings of the fifth annual workshop on Computational learning theory*, 1992, pp. 287–294.

[75] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *Journal of machine learning research*, vol. 2, no. Nov, pp. 45–66, 2001.

[76] K. Brinker, "Incorporating diversity in active learning with support vector machines," in *Proceedings of the 20th international conference on machine learning (ICML-03)*, 2003, pp. 59–66.

[77] H. T. Nguyen and A. Smeulders, "Active learning using pre-clustering," in *Proceedings of the twenty-first international conference on Machine learning*, 2004, p. 79.

[78] A. Huang, D. Milne, E. Frank, and I. H. Witten, "Clustering documents with active learning using wikipedia," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, 2008, pp. 839–844.

[79] M. Wang, F. Min, Z.-H. Zhang, and Y.-X. Wu, "Active learning through density clustering," *Expert systems with applications*, vol. 85, pp. 305–317, 2017.

[80] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050–1059.

[81] B. An, W. Wu, and H. Han, "Deep active learning for text classification," in *Proceedings of the 2nd International Conference on Vision, Image and Signal Processing*, 2018, pp. 1–6.

[82] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[83] J. Zhu, H. Wang, E. Hovy, and M. Ma, "Confidence-based stopping criteria for active learning for data annotation," *ACM Transactions on Speech and Language Processing (TSLP)*, vol. 6, no. 3, pp. 1–24, 2010.

[84] G. Beatty, E. Kochis, and M. Bloodgood, "The use of unlabeled data versus labeled data for stopping active learning for text classification," in *2019 IEEE 13th International Conference on Semantic Computing (ICSC)*. IEEE, 2019, pp. 287–294.

[85] M. Bloodgood and K. Vijay-Shanker, "A method for stopping active learning based on stabilizing predictions and the need for user-adjustable stopping," *arXiv preprint arXiv:1409.5165*, 2014.

[86] A. Vlachos, "A stopping criterion for active learning," *Computer Speech & Language*, vol. 22, no. 3, pp. 295–312, 2008.

[87] M. Ghayoomi, "Using variance as a stopping criterion for active learning of frame assignment," in *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*, 2010, pp. 1–9.

[88] V. Naidu and A. Narayanan, "Needleman-wunsch and smith-waterman algorithms for identifying viral polymorphic malware variants," in *2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing,*

14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech). IEEE, 2016, pp. 326–333.

[89] G. Sidorov, H. Gómez-Adorno, I. Markov, D. Pinto, and N. Loya, "Computing text similarity using tree edit distance," in *2015 Annual Conference of the North American Fuzzy Information Processing Society (NAFIPS) held jointly with 2015 5th World Conference on Soft Computing (WConSC)*. IEEE, 2015, pp. 1–4.

[90] S. Ren, N. Ahmed, K. Bertels, and Z. Al-Ars, "Gpu accelerated sequence alignment with traceback for gatk haplotypecaller," *BMC Genomics*, vol. 20, no. 184, 2019.

[91] L. A. Cardozo, Y. Cuervo, and J. Murcia, "Porcentaje de grasa corporal y prevalencia de sobrepeso-obesidad en estudiantes universitarios de rendimiento deportivo de bogotá, colombia," *Nutrición clínica y dietética hospitalaria*, vol. 36, no. 3, pp. 68–75, 2016.

[92] J. Daily, "Parasail: SIMD c library for global, semi-global, and local pairwise sequence alignments," *BMC Bioinformatics*, vol. 17, no. 1, feb 2016.

[93] G. Singh, B. Kumar, L. Gaur, and A. Tyagi, "Comparison between multinomial and bernoulli naïve bayes for text classification," in *2019 International Conference on Automation, Computational and Technology Management (ICACTM)*. IEEE, 2019, pp. 593–596.

[94] M. Dickinson and A. Smith, "Simulating dependencies to improve parse error detection," pp. 76–88, 2017.

[95] Y. Song, J. Wang, Z. Liang, Z. Liu, and T. Jiang, "Utilizing bert intermediate layers for aspect based sentiment analysis and natural language inference," *arXiv preprint arXiv:2002.04815*, 2020.

[96] Y. Xu, X. Qiu, L. Zhou, and X. Huang, "Improving bert fine-tuning via self-ensemble and self-distillation," *arXiv preprint arXiv:2002.10345*, 2020.

[97] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed. Series in Data Management Systems, 2011.

[98] G. Vanwinckelen and H. Blockeel, "On estimating model accuracy with repeated cross-validation." 2012.

[99] D. Chicco and G. Jurman, "The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation," *BMC genomics*, vol. 21, no. 1, p. 6, 2020.

[100] A. Charuvaka and H. Rangwala, "Hiercost: Improving large scale hierarchical classification with cost sensitive learning," pp. 675–690, 2015.

[101] J. D. Brown, "How do we calculate rater/coder agreement and cohen's kappa?" *SRB*, p. 30, 2012.

[102] R. Falotico and P. Quatto, "Fleiss' kappa statistic without paradoxes," *Quality & Quantity*, vol. 49, no. 2, pp. 463–470, 2015.

[103] J. Cerda and L. Villarroel Del, "Evaluación de la concordancia inter-observador en investigación pediátrica: Coeficiente de kappa," *Revista chilena de pediatría*, vol. 79, no. 1, pp. 54–58, 2008.

Anexo A

Conjuntos de datos

A1. Autorización para el uso de datos

En Octubre del año 2019 se obtuvo la aprobación del comité de ética del HGGB de Concepción (ver Figura A1.1) para el uso de los textos biomédicos en este trabajo de tesis.

U. DE INVEST. Y DESARROLLO
Ing. Nº 120 11.10.18
DRA. MABS / RSC / mega

005018 17.10.2018

ORD. Nº / _____
Ant: Trabajo de Investigación
Mat: Autorización Estudio
COD. 18-09-69

DE : SR. CARLOS CAPURRO DUPRE
DIRECTOR HOSP.GMO.GRANT B.

A : SRTA ROSA FIGUEROA ITURRIETA
INVESTIGADORA RESPONSABLE

Junto con saludar, sírvase encontrar adjunto Acta de aprobación de la Comisión de Investigación del Hospital Clínico Regional Dr.G.Grant Benavente, con fecha 09.10.18,en atención a la cual esta Dirección autoriza el desarrollo del estudio titulado: **Desarrollo de algoritmos inteligentes para ayudar en la toma de decisiones clínicas y en la prestación efectiva de atención de la salud aprendizaje activo"**, bajo su responsabilidad como investigadora responsable en este Centro.

Saluda atentamente,

HOSPITAL GMO. GRANT BENAVENTE DIRECTOR CONCEPCION

ING. CARLOS CAPURRO DUPRE
DIRECTOR
HOSPITAL GUILLERMO GRANT BENAVENTE

Distribución
IR RFI
O.Partes
Archivo U.de Invest.

Figura A1.1: Autorización del HGGB de Concepción, Chile, para el uso de datos de-identificados. Fuente: HGGB de Concepción, Chile

A2. Herramienta de anotación

La Figura A2.1 muestra la herramienta de anotación que permitió etiquetar manualmente los textos biomédicos utilizados en este trabajo. Esta herramienta, diseñada en *Qt4-designer* y programada en *Python*, permite ingresar información relevante (palabras claves) durante las sesiones de anotación.

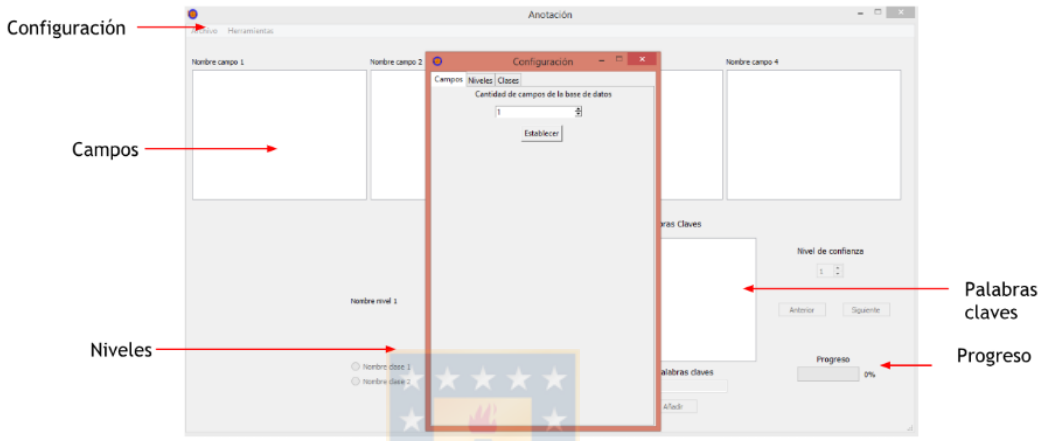

Figura A2.1: Herramienta de anotación para el etiquetado de textos biomédicos. Fuente: Elaboración propia.

A3. Índice de kappa

El índice de kappa (k) es una medida estadística utilizada para medir el nivel de acuerdo entre los observadores que participaron de un proceso de anotación. Se puede calcular según:

$$k = \frac{P_a - P_e}{1 - P_e}, \tag{A.1}$$

Con P_a = concordancias observadas y P_e = concordancias atribuibles al azar. Si el número de anotadores es igual a dos, se utiliza el índice de *kappa* de *Cohen* [101]. Las probabilidades P_a y P_e se calculan a partir de una matriz$_{CxC}$ de confusión que contabiliza las anotaciones de los observadores según:

$$P_a = \frac{1}{M} \sum_{i=1}^{C} M_{ii}, \tag{A.2}$$

$$P_e = \frac{1}{M^2} \sum_{i=1}^{C} M_{i\cdot} M_{\cdot i}, \tag{A.3}$$

Donde C es el número de clases y M el número de ejemplos anotados. Por otro lado, si el grupo de anotadores es mayor a dos se utiliza el índice de *kappa* de *Fleiss* [102]. Las probabilidades P_a y P_e se calculan a partir de una matriz$_{MxC}$ que contabiliza las anotaciones de los observadores según:

$$P_a = \frac{1}{M} \sum_{i=1}^{M} P_{e_i}, \tag{A.4}$$

Donde

$$P_i = \frac{1}{m(m-1)} (\sum_{i=1}^{C} m_{ij}^2 - m), \tag{A.5}$$

$$P_e = \sum_{i=1}^{C} P_j^2, \tag{A.6}$$

Donde

$$P_j = \frac{1}{Mm} \sum_{i=1}^{M} m_{ij}, \tag{A.7}$$

Donde m es el número de observadores. Dependiendo del valor de k es posible establecer un nivel de acuerdo entre los anotadores según lo indicado en la Tabla A3.1 [103].

Tabla A3.1: Nivel de acuerdo entre los anotadores según el índice de kappa (k). Fuente: Elaboración propia.

k	Nivel de acuerdo
<0.00	Sin acuerdo
0.00-0.20	Insignificante
0.21-0.40	Mediano
0.41-0.60	Moderado
0.61-0.80	Sustancial
0.81-1.00	Casi perfecto

Anexo B

Algoritmos de alineación

B1. Algoritmo de Needleman-Wunsch (NW)

El algoritmo de NW permite alinear globalmente dos secuencias A y B de longitudes n_A y n_B respectivamente, utilizando una matriz H de dimensiones n_{A+1} y n_{B+1}. Una vez que la matriz ha sido inicializada en la primera columna y fila con valores negativos, el algoritmo asigna valores para cada posición $H_(i,j)$ según:

$$H(i,j) = \max \left\{ \begin{array}{l} H(i-1,j-1) + w(A_i, B_j) \\ H(i-1,j) + w(A_i, -) \\ H(i,j-1) + w(-, B_j) \end{array} \right\}, n_A \geq i \geq 1, n_B \geq j \geq 1 \tag{B.1}$$

Donde $w(A_i, B_j)$ es una constante definida positiva o negativa en caso que los caracteres A_i y B_j coincidan o no coincidan, respectivamente, mientras que $w(A_i, -)$ y $w(-, B_j)$ son constantes definidas negativas. Un ejemplo de alineación se muestra en la Figura B1.1.

NW	-	B_1	B_2	B_3	B_4	B_5
-	0	-1	-2	-3	-4	-5
A_1	-1	1	0	-1	-2	-3
A_2	-2	0	2	1	0	-1
A_3	-3	-1	1	3	2	1
A_4	-4	-2	0	2	4	3
A_5	-5	-3	-1	1	3	3
A_6	-6	-4	-2	0	2	2
A_7	-7	-5	-3	-1	1	1
A_8	-8	-6	-4	-2	0	0

A:	o	b	e	s	i	d	a	d
B:	o	b	e	s	-	-	-	o

Figura B1.1: Ejemplo de alineación global mediante el algoritmo de NW. A= "obesidad". B = "obeso". Se consideró un valor igual a -1 para todas las constantes negativas y un valor igual a 1 para las constantes positivas. Fuente: Elaboración propia.

B2. Algoritmo de Smith-Waterman (SW)

El algoritmo de SW permite alinear localmente dos secuencias A y B de longitudes n_A y n_B respectivamente, utilizando una matriz H de dimensiones n_{A+1} y n_{B+1}. Una vez que la matriz ha sido inicializada en la primera columna y fila con ceros, el algoritmo asigna valores para cada posición $H_(i,j)$ según:

$$H(i,j) = \max \left\{ \begin{array}{l} 0 \\ H(i-1,j-1) + w(A_i, B_j) \\ H(i-1,j) + w(A_i, -) \\ H(i,j-1) + w(-, B_j) \end{array} \right\}, n_A \geq i \geq 1, n_B \geq j \geq 1 \tag{B.2}$$

Donde $w(A_i, B_j)$ es una constante definida positiva o negativa en caso que los caracteres A_i y B_j coincidan o no coincidan, respectivamente, mientras que $w(A_i, -)$ y $w(-, B_j)$ son constantes definidas negativas. Un ejemplo de alineación se muestra en la Figura B2.1.

SW	-	B1	B2	B3	B4	B5	B6	B7	B8
-	0	0	0	0	0	0	0	0	0
A_1	0	2	1	0	0	0	0	0	0
A_2	0	1	10	9	8	7	6	5	4
A_3	0	0	9	9	8	7	6	5	4
A_4	0	0	8	8	8	22	21	20	19
A_5	0	0	7	7	7	21	21	20	19
A_6	0	0	6	6	6	20	24	23	22
A_7	0	0	5	5	5	19	23	25	24
A_8	0	0	4	4	4	18	22	24	48

A:	el	paciente	-	es	obes(?:\w{,4})	con	imc	=	3[5-9]{1}(?:[\.\,]\d+)?
B:	el	paciente	fumador(?:\w{,1})	sufre	obes(?:\w{,4})	-	imc	=	3[5-9]{1}(?:[\.\,]\d+)?

Figura B2.1: Ejemplo de alineación local mediante el algoritmo de SW. A= "el paciente es obes(?:\w{,4}) con imc = 3[5-9]{1}(?:[\.\,]\d+)?". B = "el paciente fumador(?:\w{,1}) sufre obes(?:\w{,4}) imc = 3[5-9]{1}(?:[\.\,]\d+)?". Se consideró un valor igual a -1 para todas las constantes negativas y un valor igual a la cantidad de caracteres de los *tokens* que coinciden para las constantes positivas. Fuente: Elaboración propia.

Anexo C

Resultados de clasificación

C1. Curvas de error de los clasificadores

Las Figuras C1.1, C1.2 y C1.3 muestran las curvas de error de entrenamiento de los clasificadores en todos los conjuntos de datos. En todos los casos, el error de entrenamiento es menor al error de pruebas.

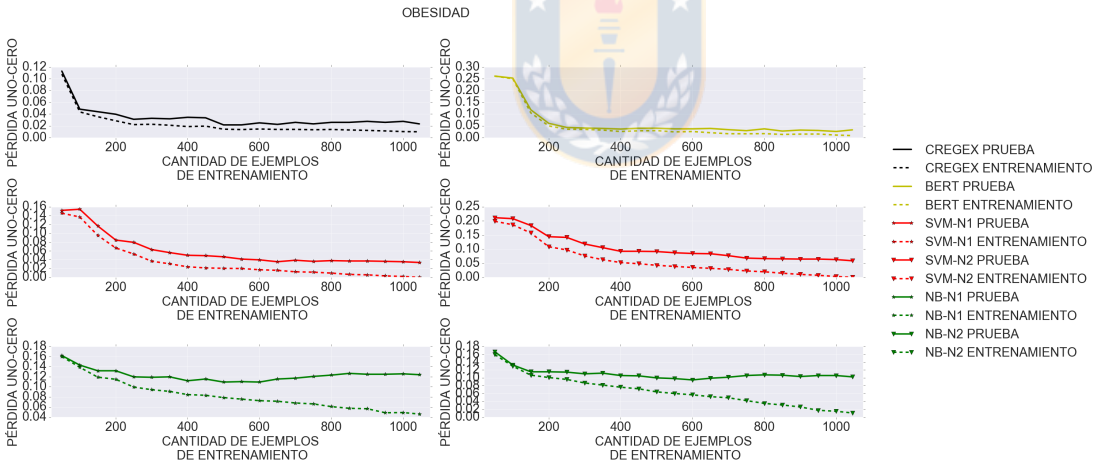

Figura C1.1: Curvas de error de los clasificadores en el conjunto de datos OBESIDAD. Fuente: Elaboración propia.

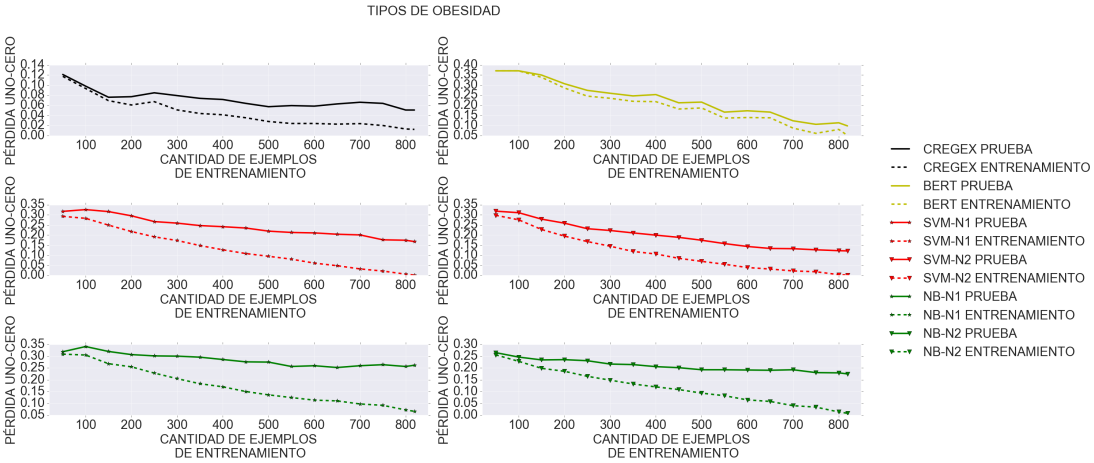

Figura C1.2: Curvas de error de los clasificadores en el conjunto de datos TIPOS DE OBESIDAD. Fuente: Elaboración propia.

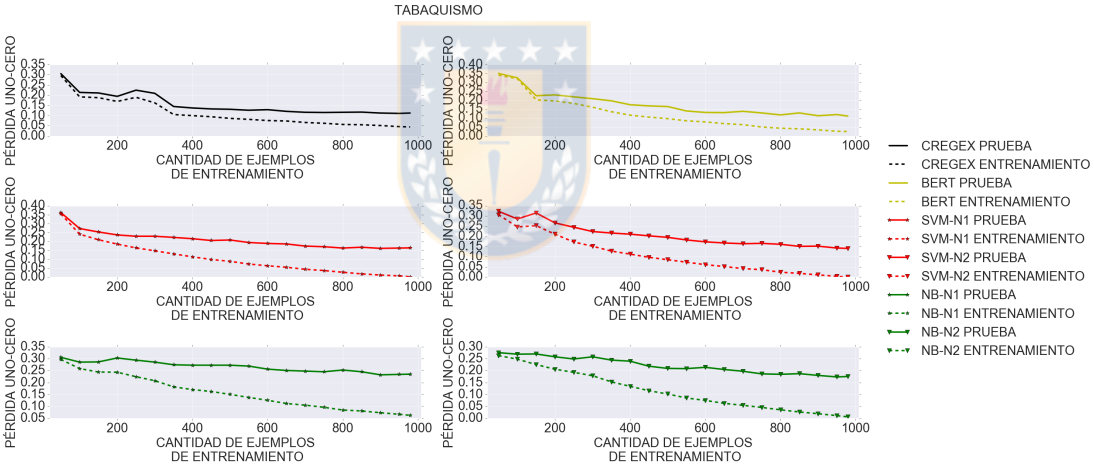

Figura C1.3: Curvas de error de los clasificadores en el conjunto de datos TABAQUISMO. Fuente: Elaboración propia.

Anexo D

Aprendizaje activo

D1. Criterio de detención del aprendizaje activo

Las Figuras D1.1, D1.2 y D1.3 muestran los valores de varianza promedio de las estrategias de consulta de los clasificadores en todos los conjuntos de datos. Es posible observar el patrón de varianzas utilizado para detener el proceso de aprendizaje activo (AL).

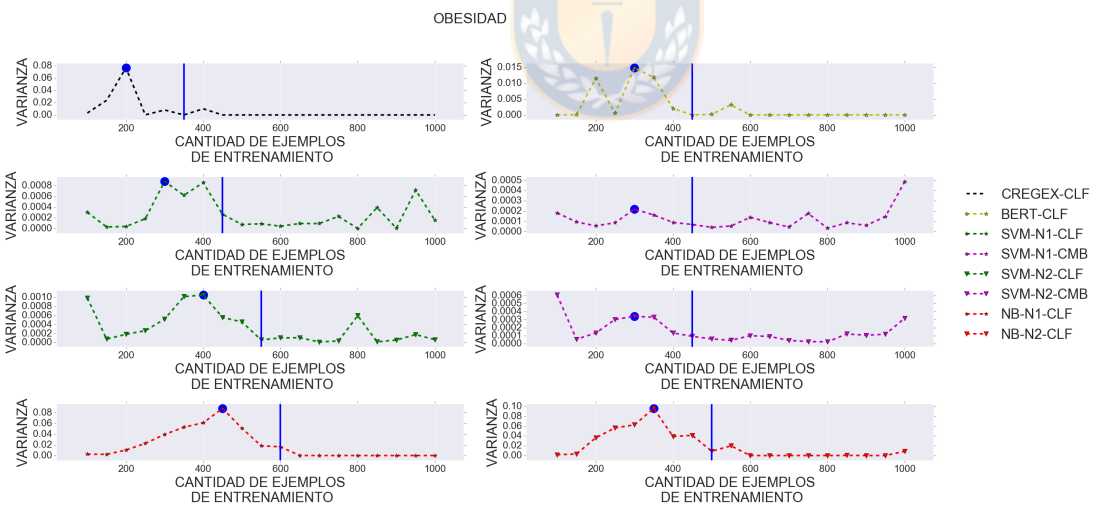

Figura D1.1: Criterio de detención para el proceso de aprendizaje activo de los clasificadores en el conjunto de datos OBESIDAD. Fuente: Elaboración propia.

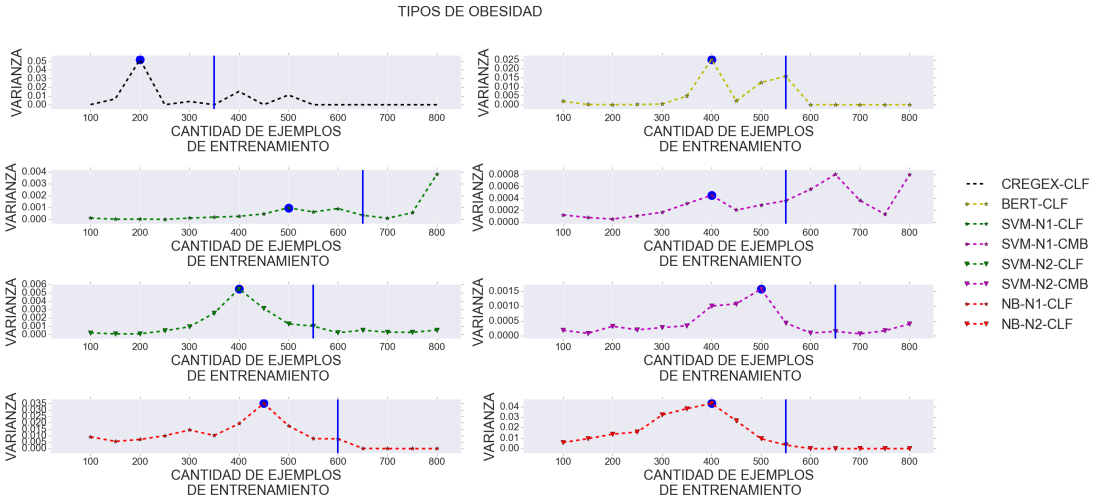

Figura D1.2: Criterio de detención para el proceso de aprendizaje activo de los clasificadores en el conjunto de datos TIPOS DE OBESIDAD. Fuente: Elaboración propia.

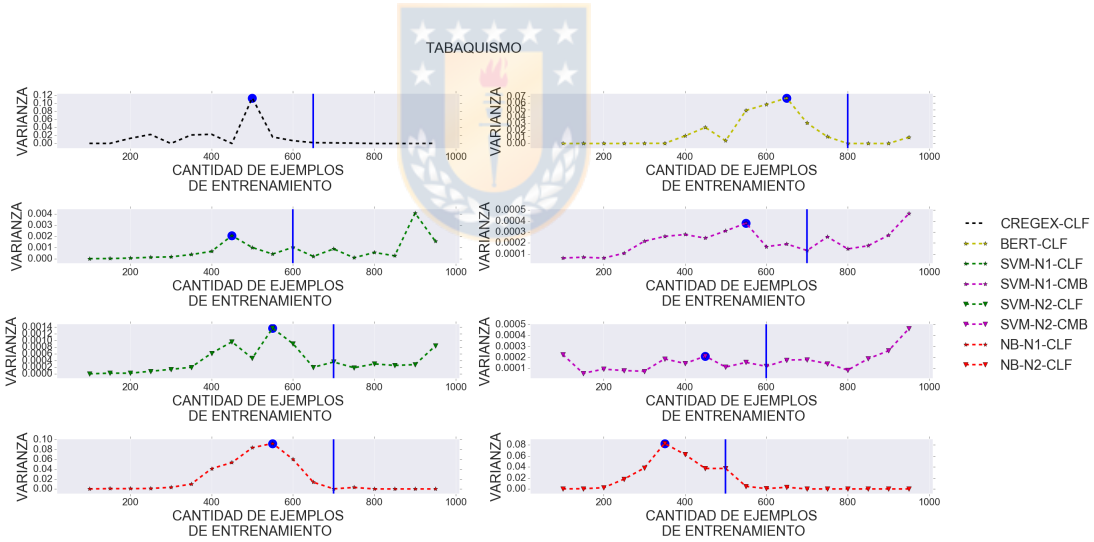

Figura D1.3: Criterio de detención para el proceso de aprendizaje activo de los clasificadores en el conjunto de datos TABAQUISMO. Fuente: Elaboración propia.

Anexo E

Tiempos de ejecución

E1. Entrenamiento de los clasificadores

Los experimentos realizados en SVM, NB, FREGEX y CREGEX fueron ejecutados en un ordenador Intel(R) Xeon(R) Silver 4110 CPU @2.10 GHz 2.10 GHz (2 procesadores), 30 GB RAM. En cuanto al clasificador basado en BERT, se utilizó la plataforma Google Colab [1] en su versión PRO: Intel(R) Xeon(R) CPU @ 2.30GHz, GPU Tesla P100-PCIE-16GB, 25 GB RAM. La Figura E1.1 indica que los algoritmos propuestos tienen un mayor tiempo de ejecución que el resto de los clasificadores. Sin embargo, en el caso de BERT no está considerado el pre-entrenamiento de este modelo de lenguaje (4 días) [1].

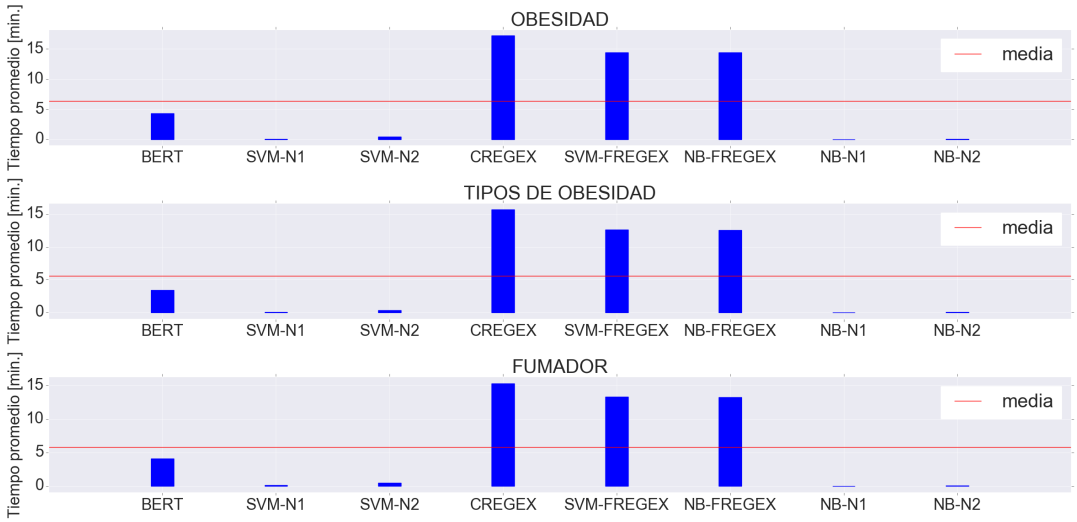

Figura E1.1: Tiempos de ejecución del entrenamiento de los clasificadores implementados. Fuente: Elaboración propia.

[1] https://colab.research.google.com