BIOINFORMATICS

Minería de textos para la extracción de asociaciones polimorfismo-enfermedad en literatura biomédica

Lucía G Ramírez*, Gilberto Durán, Luis F Altamirano and Luis E Ramírez

*Correspondence:
luciarn@lcg.unam.mx
Licenciatura en Ciencias Genómicas, Av. Universidad, Cuernavaca, México
Full list of author information is available at the end of the article

Abstract

Antecedentes: Debido al aumento de artículos publicados en los últimos años, se ha vuelto importante desarrollar nuevas técnicas para resumir esa información, la minería de textos se ha vuelto una de las herramientas más usadas en la bioinformática para el procesamiento de datos. En este trabajo nos enfocamos en saber cual clasificador (SVM o MLP) es mejor para la identificación de asociaciones polimorfismo-enfermedad.

Resultados: El mejor clasificador se obtuvo con SVM al procesar archivos convertidos a lemas que contenían categorías gramaticales, etiquetas a los números rs, enfermedades y una serie de palabras clave. Nuestro método tuvo un puntaje F de 0.7113 al ser evaluado con 607 oraciones curadas manualmente. Al implementarlo en resúmenes de literatura biomédica se obtuvo un puntaje-F de 0.1628, el cual es mejor que los obtenidos en el primer enfoque.

Conclusiones: La SVM fue el mejor clasificador obtenido permitiéndonos extraer asociaciones polimorfismo-enfermedad de literatura biomédica. Se podrían obtener valores más óptimos para la clasificación realizando todas las combinaciones posibles de parámetros e hiperparámetros. El clasificador puede ser usado para la clasificación de cualquier tipo de asociación entre una enfermedad y un ID único en literatura biomédica.

Keywords: minería de textos; SVM; MLP; Clasificador; Puntaje F

Introducción

La bioinformática es una disciplina que combina la biología, computación y las nuevas tecnologías de información. Anteriormente, esta ciencia solo se ocupaba para la creación de bases de datos de información biológica, las cuales almacenan grandes cantidades de datos y las ordena de tal manera que sean accesibles y amigables a un usuario.[1]A raíz de la explosión de información de los últimos años, el número de artículos publicados ha excedido el número de artículos que podemos leer. Por ejemplo, en el 2017, en PubMed de NCBI han publicado más de 110,000 artículos relacionados a temas de biología, más de 320,000 de medicina, más de 75,000 de genética, entre otros. Por lo cual, se ha vuelto importante la necesidad de resumir la información publicada día a día.

Una de las herramientas más utilizadas en la bioinformática para el procesamiento de datos es la minería de textos. Esta se define como el proceso de reconocimiento de patrones en un conjunto de textos para el descubrimiento de conocimiento. Es decir, esta rama de la bioinformática se encarga de generar nuevo conocimiento,

analizando una serie de artículos relacionados con el tema de interés , para ser capturado y utilizado para diversas investigaciones.[2]

Con la finalidad de realizar una base de datos sobre las interacciones entre genes, enfermedades y polimorfismos, se ha desarrollado un método de minería de textos para analizar y procesar la información contenida en literatura biomédica sobre interacciones polimorfismo-enfermedad de una forma automática y eficiente.

Para ello, se emplearon dos diferentes métodos: una Máquina de Vector de Soporte (Support Vector Machine, SVM) y Perceptrón Multicapa (Multilayer Perceptron, MLP) con 5 tipos de n-gramas, unigrama, bigrama, trigrama, la combinación unigrama-bigrama y la combinación de los tres. Se definieron hasta 5 n-gramas porque notamos que al utilizar más, el proceso se vuelve más costoso computacionalmente sin mejorar el resultado obtenido. El primer método empleado es un método lineal que, a diferencia de otros clasificadores, utiliza hiperplanos. Una de las ventajas que presenta para la minería de textos es que los clasificadores con gran margen no son capaces de tomar decisiones de baja certidumbre, por lo que sus errores de medición o variaciones son pequeñas y es menos probable que causen errores en la clasificación. Mientras que el segundo método es no lineal e intenta resolver problemas simulando al cerebro humano. Una de la ventajas de este método es que pueden resolver problemas complejos porque cada neurona del clasificador toma distintos pesos de las entradas para poder analizarlos de diferentes formas generando una o varias salidas.

El mejor resultado obtenido a partir de estas combinaciones, fue SVM con un puntaje de 0.5806 en el entrenamiento y 0.7113 en la evaluación; a comparación de MLP cuyos mejores resultados fueron 0.3613 en el entrenamiento, por lo que al ser muy bajo no se consideró este clasificador para la evaluación. Posteriormente, al saber que el SVM fue el mejor clasificador, se utilizó para la clasificación de resúmenes de literatura biomédica para lograr extraer la asociación entre un polimorfismo y su enfermedad. El clasificador logró extraer correctamente un total de 1,562 oraciones de un total de 4,830 con un puntaje-F de 0.1628.

Métodos

Clasificadores

Una de las técnicas más utilizadas en minería de textos son las de clasificación, las cuales consisten en asignar objetos a categorías predefinidas. Usualmente, los clasificadores son construidos utilizando aprendizaje máquina; por el cual, un proceso examina las características de un conjunto de documentos de entrenamiento ya clasificados y a partir de estas características, infiere las condiciones que documentos nuevos o no examinados deberían cumplir para ser clasificados bajo estas categorías.[3]

En este trabajo se hizo uso de métodos de aprendizaje máquina para clasificar y extraer asociaciones positivas entre fenotipos (enfermedades y otros trastornos de salud) y sus polimorfismos de nucleótido simple (SNP), particularmente sus números de referencia ("número de refSNP" o "número ID de rs"). Las redes neurales y las máquinas de vectores soporte han probado tener éxito como clasificadores en muchos estudios, incluyendo estudios similares en la minería de textos de la literatura biomédica [12, 13, 14, 15], por lo que fueron elegidos para resolver el problema de asociaciones.

Máquinas de Vector de Soporte

Las máquinas de vector de soporte son un sistema computacional lineal para entrenar máquinas de aprendizaje utilizadas para la clasificación como para la regresión.[4]

La principal idea detrás de estos clasificadores binarios es que a partir de unos archivos de entrada ya etiquetados, la SVM sea capaz de predecir la clase de nuevos datos que sean introducidos por el usuario. Para hacer esta clasificación, el modelo proyecta los datos pertenecientes a una dimensión n hacia un espacio de una dimensión mayor aplicando una función kernel.[5] Ya aplicada la función, el modelo representa en el espacio los llamados vectores de entrenamiento *(support vectors, SV)* los cuales son líneas paralelas con cierta dirección que toman en cuenta el punto más alejado de cada clase creando un margen *(m)*. Dentro del margen, se traza un hiperplano justo a la mitad de los vectores de entrenamiento creando el clasificador final. [*Fig 1a*[5]]. Así, cuando nuevos datos son introducidos al modelo, se colocan sobre el mismo espacio y en función de la cercanía de los grupos antes separados, puede predecir a qué clase pertenecen.

Redes Neuronales

Las redes neuronales (Neural Networks, NN) son un sistema computacional no lineal para el entrenamiento de máquinas de aprendizaje utilizadas para el reconocimiento de patrones, clasificación y predicción de series temporales. Una de las NNs más utilizadas y más simples para la clasificación es el llamado Perceptrón multicapa (MLP).

MLP es un método el cual consta de 3 fases: el ingreso de las entradas, el procesamiento de las entradas y la generación de uno o varios resultados. En la primera fase, el modelo necesita la introducción de 1 o varios datos, conocidos como descriptores, para poder ser analizados. En la segunda, se le asigna pesos a las entradas y se mandan a la capa escondida. En esta capa, cada neurona de la red procesa la información recibida de las entradas de manera distinta y produce una respuesta en común. Esas respuestas, que es la última fase, obtienen un valor entre 0 y 1 (*fig 1b*[8]). [6, 7, 8] Es importante destacar que el modelo aprende por cada iteración, es decir, cada vez que obtenga un resultado va a reorganizar los pesos de las entradas para intentar generar un mejor resultado, las veces que hace esto depende del número de iteraciones que el usuario haya elegido.

Extracción automática de interacciones gene-enfermedad

Clasificación automática de frases

Primer enfoque:

Como primer acercamiento al problema, se asumió que existía una relación positiva entre un fenotipo y un rs si los dos estaban presentes en una misma oración. Posteriormente se recuperaron todas las combinaciones posibles de cada oración, por ejemplo, si un rs estaba presente junto con dos fenotipos se recuperaba cada enfermedad con el rs. Finalmente se recuperaron las asociaciones de la manera previamente descrita utilizando etiquetas, las cuales fueron previamente colocadas a lado del rs o la enfermedad según el caso. Para esta suposición se pueden señalar las ventajas de simplicidad y bajo costo computacional, sin embargo, la clasificación

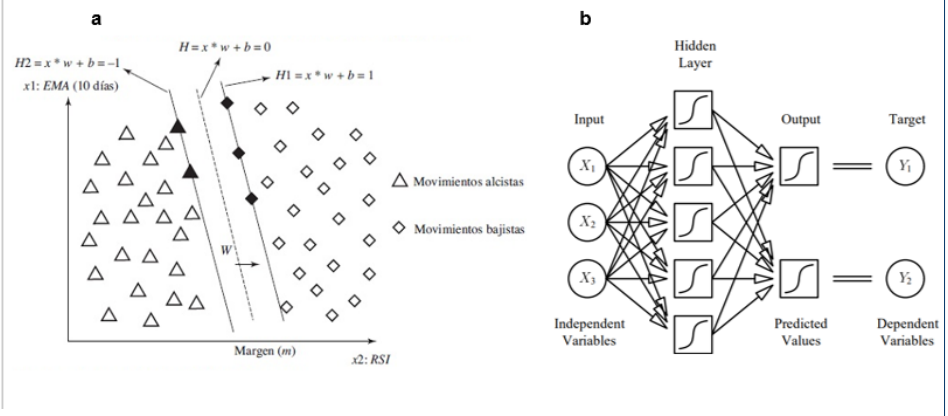


Figure 1 Métodos de clasificación utilizados en míneria de textos. (a) Máquinas de Vector de Soporte: H representa la clasificación del hiperplano. H1 y H2 son los SV (Support Vectors) definidos por los puntos más alejados de cada clase. El ancho del margen (m) esta dado por la distancia entre los hiperplanos H1 y H2. (b) Perceptrón Multicapa: la capa de entrada representa los descriptores elegidos. La capa escondida representa la transformación de los datos en una dimensión mayor. La capa de salida representa el valor final obtenido dependiendo de los descriptores elegidos.

incorrecta de varias asociaciones no hace de esta una buena opción. Por lo que este primer enfoque nos sirve como base para comparar el rendimiento de métodos más complejos que requieren un mayor tiempo computacional y que han demostrado ser mejores como clasificadores (aprendizaje máquina).

Preparación de datos:

Se partió de resúmenes de artículos de literatura biomédica previamente recolectados, los cuales son la fuente de donde se recuperaron las asociaciones. Estos resúmenes fueron procesados para separarlos por oraciones y obtener las categorías gramaticales. Posteriormente se tomó el archivo de los resúmenes que se habían preprocesado y se buscó en ésta cada enfermedad que se encontrara en el diccionario, el cual fue sacado de la base de datos de Online Mendelian Inheritance in Man (OMIM), de manera que si se encontraba se señalaba con una etiqueta. Lo mismo se hizo con los rs, pero en vez de buscarlo en un diccionario se buscó asumiendo que un rs siempre iniciaba con las letras rs y estaba seguido de 2 a 6 números.

Entrenamiento y evaluación:

En este trabajo se entrenaron diferentes SVMs y MLPs con los archivos generados en el procesamiento de datos, probando un conjunto determinado de parámetros con diferentes opciones cada uno, lo que permitió construir 600 combinaciones diferentes que serían nuestra grid de experimentos.

Los parámetros e hiperparámetros usados para la SVM y MLP, así como sus combinaciones son descritos en la figura 2. Al final se seleccionaron los parámetros e hiperparámetros que daban los mejores puntajes para así seleccionar el mejor clasificador y pasarlo a la parte de evaluación.

En la parte de evaluación, el mejor clasificador fue evaluado tomando como medida principal el puntaje-F el cual es calculado a partir de la siguiente fórmula:

$$F1 = 2 * (Precision * Recall / Precision + Recall)$$

Donde la precisión está dada por:

$$P = VP/(VP + FP)$$

Y la exhaustividad por:

$$R = VP/(VP + FN)$$

Donde VP son los verdaderos positivos, FP los falsos positivos y FN los falsos negativos.

Sin embargo, con el bajo desempeño de los clasificadores se implementaron distintas estrategias para tratar de maximizar el puntaje:

Se realizó un nuevo diccionario de enfermedades a través del parseo de datos de entradas de la base de datos Online Mendelian Inheritance in Man (OMIM), del National Center for Biotechnology Information (NCBI) y de una lista de enfermedades del Genetic and Rare Diseases Information Center (GARD) y del National Institute of Health (NIH). Estas fuentes nos permitieron generar un diccionario de enfermedades el cual se puso bajo un filtrado que consistió en conservar sólo los fenotipos que se encontraran presentes en los resúmenes, incluyendo abreviaturas, resultando en un total de 6,274 enfermedades. Posteriormente, se revisó el diccionario nuevamente y se quitaron palabras que podrían causar ruido, como abreviaturas de 3 letras que no consideramos como enfermedad (ej: OLD), dejando un total de 2,690 . Todo esto se realizó para disminuir el tiempo computacional y aumentar la eficacia en las etapas posteriores del presente proyecto.

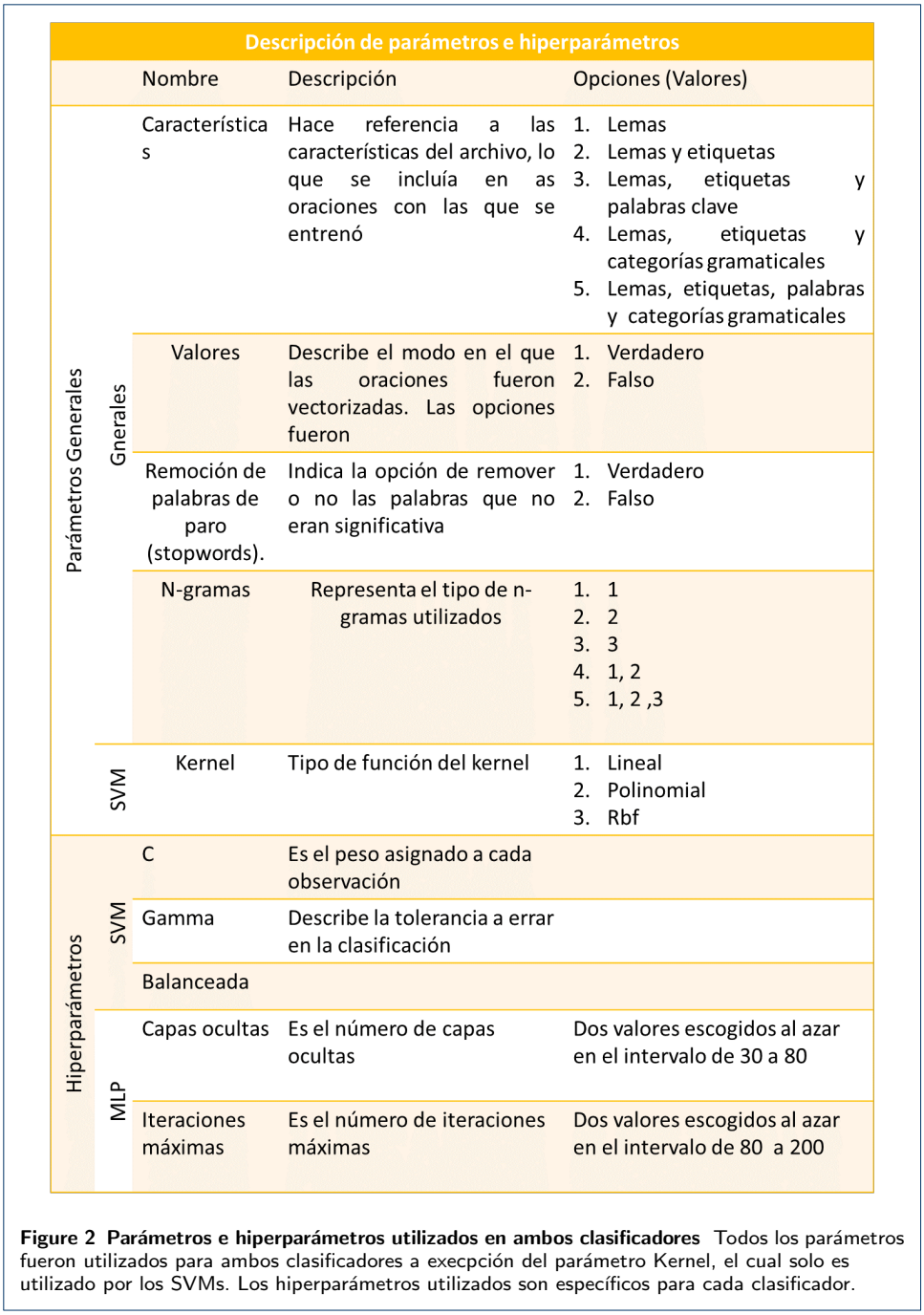
De igual manera, se etiquetaron los archivos utilizando el diccionario ya filtrado y empleando similitud de palabras. Se generaron archivos distintos con las combinaciones de etiquetados de rs number, enfermedades y palabras clave, que dieron origen a una combinatoria de 6 nuevos archivos los cuales fueron evaluados por ambos clasificadores con sus respectivos parámetros e hiperparámetros.

Extracción de Enfermedades

El listado de resúmenes, que fueron seleccionados para la creación de una base de datos, fueron pre-procesados para que cuando fueran enviados a la herramienta de CoreNLP de Standford versión 3.8.0 [11], las enfermedades tuvieran un solo token. Este procesamiento consiste en juntar las enfermedades por guiones para que, al ser evaluado por espacios, el conjunto de palabras que conforman una enfermedad sea tomada como una sola. Lo anterior se realizó para tener un mejor control de los datos. Basándonos en el archivo que obtuvo el mejor puntaje en el entrenamiento, pre-procesamos los resúmenes para que tuvieran las mismas características: procesadas por lemas y que contengan etiquetas gramaticales de los rs, las enfermedades y palabras clave.

Experimentos

Los programas que se utilizaron para correr los algoritmos fueron implementados en el lenguaje de programación Python versión 3 [9], haciendo uso principalmente de la librería SickitLearn versión 0.19 [10], la cual contiene especialmente metodologías



Descripción de parámetros e hiperparámetros				
		Nombre	Descripción	Opciones (Valores)
Parámetros Generales	Gnerales	Características	Hace referencia a las características del archivo, lo que se incluía en as oraciones con las que se entrenó	1. Lemas 2. Lemas y etiquetas 3. Lemas, etiquetas y palabras clave 4. Lemas, etiquetas y categorías gramaticales 5. Lemas, etiquetas, palabras y categorías gramaticales
		Valores	Describe el modo en el que las oraciones fueron vectorizadas. Las opciones fueron	1. Verdadero 2. Falso
		Remoción de palabras de paro (stopwords).	Indica la opción de remover o no las palabras que no eran significativa	1. Verdadero 2. Falso
		N-gramas	Representa el tipo de n-gramas utilizados	1. 1 2. 2 3. 3 4. 1, 2 5. 1, 2 ,3
	SVM	Kernel	Tipo de función del kernel	1. Lineal 2. Polinomial 3. Rbf
Hiperparámetros	SVM	C	Es el peso asignado a cada observación	
		Gamma	Describe la tolerancia a errar en la clasificación	
		Balanceada		
	MLP	Capas ocultas	Es el número de capas ocultas	Dos valores escogidos al azar en el intervalo de 30 a 80
		Iteraciones máximas	Es el número de iteraciones máximas	Dos valores escogidos al azar en el intervalo de 80 a 200

Figure 2 Parámetros e hiperparámetros utilizados en ambos clasificadores Todos los parámetros fueron utilizados para ambos clasificadores a execpción del parámetro Kernel, el cual solo es utilizado por los SVMs. Los hiperparámetros utilizados son específicos para cada clasificador.

relacionadas con el aprendizaje automático. Los experimentos que a continuación se describen se dividieron en 4 partes:

1. Un enfoque previo al uso de las herramientas del aprendizaje máquina en el cual se asumió una hipótesis simple para posteriormente extraer las asociaciones.

2. Entrenamiento de la SVM y MLP a través del método de validación cruzada usando 10 particiones.

3. Evaluación del mejor método.

4. Extracción de las asociaciones fenotipo-rs de los resúmenes de artículos científicos.

Primer enfoque:
Se realizó un script en Python 3, el cual extrae las asociaciones de las oraciones y regresa el puntaje-F de esa extracción. Este script se corrió tres veces: la primera fue utilizando el diccionario de OMIM, el segundo fue utilizando el diccionario filtrado y el tercero utilizando similitud de cadenas mayor a 85 porciento, la cual se utilizó para aumentar el número de extracciones correctas al compararlas con el Gold Standard. Esto se hizo para evitar que interacciones verdaderas fueran excluidas como verdaderos positivos si la notación de la enfermedad era ligeramente distinta entre la referencia y las asociaciones extraídas.

Preparación de datos:
Se creó un diccionario con los nombres de enfermedades el cual fue filtrado con las enfermedades de los resúmenes para disminuir tiempo computacional. Se partió de 16,661 resúmenes de artículos de literatura biomédica previamente recolectados. Usando la herramienta coreNLP de Standford versión 3.8.0, los resúmenes fueron procesados para separarlos en oraciones y convirtiéndolas a lemas. Realizamos un script que etiqueta en base al diccionario todos los rs number, enfermedades y palabras clave (como 'association', 'related', 'did not', 'is not' entre otras) y se generaron 5 archivos para ser entrenados por cada clasificador.

Grid de experimentos:
La figura 3 muestra la línea de trabajo seguida en el grid de experimentos, el cual consistió en obtener todas las combinaciones de cada parámetro con los hiperparámetros, tanto de las SVMs como de las MLPs, llegando a tener un total de 600 combinaciones sólo contando los parámetros (450 de las SVMs y 150 de las MLPs). Los parámetros usados que compartían todas las combinaciones fueron características, valores, remoción de palabras de paro (stopwords), n-gramas. Para la SVM también se incluyó el parámetro kernel (que si es polinomial se agrega el hiperparámetro de grado que puede tomar valores de 2 o 3) y los hiperparámetros C, gamma y balanced; mientras que para MLP los hiperparámetros fueron capas ocultas e iteraciones máximas (figura 2).

Entrenamiento y evaluación:
Se usó un conjunto de entrenamiento de 6,390 oraciones, de las cuales 603 tenían una asociación positiva y las restantes 5,787 tenían una clasificación negativa. Con la combinación de diferentes parámetros, se entrenaron los clasificadores utilizando la librería de SickitLearn en Python 3. Adicionalmente, se fueron implementando variaciones en los hiperparámetros permitiéndonos elegir el mejor clasificador para pasarlo a la fase de evaluación. En la fase de evaluación, realizamos un script en Python 3 para calcular el puntaje-F del clasificador mejor entrenado para poder implementarlo a los resúmenes previamente dados. Esta evaluación se hizo con un conjunto de 209 oraciones positivas y 398 negativas.

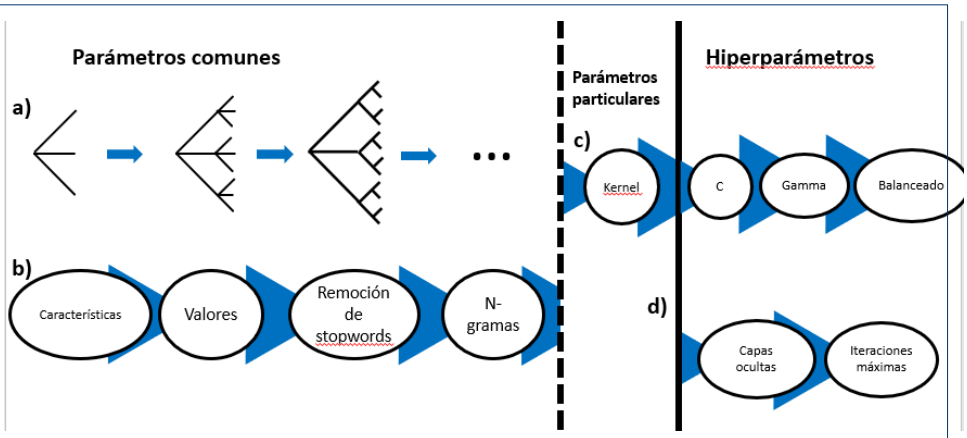


Figure 3 Línea de trabajo general seguida en el grid de experimentos a) Las combinaciones se hicieron tomando todas las relaciones posibles entre los valores de cada parámetro o hiperparámetro, tomando sólo un valor de cada parámetro o hiperparámetro, de esta manera se podría trazar un árbol que crece de una manera multiplicativa y cuyas ramas representen cada combinación. b) Parámetros compartidos en todas las combinatorias realizadas. C) Hiperparámetros especiales utilizados en la SVM, también incluye un parámetro particular utilizado en el modelo. D) Hiperparámetros encontrados para MLP.

Extracción de Enfermedades:

Se corrió el script en Python 3 del mejor clasificador aplicados a los resúmenes de literatura biomédica previamente recolectados. Los resúmenes fueron pre-procesados para que solo tuvieran un token y se prepararon con las características del mejor archivo evaluado. También se corrió el método usando las oraciones en las que se identificaban al menos un rs y al menos un trastorno de salud.

Resultados y Discusión

Los métodos de aprendizaje máquina, como se puede ver en este trabajo, muestran un mejor desempeño que otros enfoques y/o herramientas. En nuestro primer paso, donde se extrajeron directamente las asociaciones sin una clasificación previa de ella se obtuvieron tres puntajes-F: 0.0724 con el diccionario de OMIM, 0.1351 con el diccionario extendido (con datos de OMIM y de GARD) y 0.1432 empleando la similitud de cadenas mayor a 85 por ciento. Estos puntajes se pudieron haber dado por la mala clasificación implícita de nuestra hipótesis (considerar una asociación si un rs y un fenotipo se encontraban en la misma oración), porque sin importar los esfuerzos realizados para mejorar el puntaje, este sigue siendo bajo. Es por ello que se recurrieron a las metodologías de aprendizaje máquina para abordar este problema.

Comparando ahora los resultados obtenidos de la fase de entrenamiento, mostrados en la figura 4, podemos observar que las SVMs tuvieron un mejor desempeño que las MLPs, siendo las primeras aproximadamente 2 veces mejores en términos del puntaje-F, si sólo nos enfocamos en el mejor resultado de cada clasificador.

Consecuentemente en la fase posterior de evaluación se usó solo la mejor combinación de parámetros e hiperparametros, obteniendo un valor-F de 0.7113, la cual indica que el clasificador sí supo clasificar la mayoría de las oraciones de manera correcta demostrando que el entrenamiento, a pesar de su puntaje medio, si funcionó de manera correcta (figura 5b).

Mejores 5 resultados del entrenamiento de cada clasificador										
	Parámetros del clasificador						Hiperparámetros			Valor F
	Características	Con palabras clave	Valores	Remoción de palabras	N-gramas					
SVM						Kernel	C	Gamma	Balanceado	
	Categorías gramaticales	V	BINARIO	V	1	Rbf	1.6962	0.0169	V	0.5806
		V	BINARIO	F	1	Rbf	2.0067	0.0202	V	0.5769
	Categorías gramaticales	V	BINARIO	F	1	Rbf	1.6962	0.0169	V	0.5762
MLP							Capas ocultas	Iteraciones máximas		
	Categorías gramaticales	V	BINARIO	V	1,		31	93	No aplica	0.3613
	Categorías gramaticales	F	BINARIO	F	1		78	99	No aplica	0.3600
	Categorías gramaticales	V	BINARIO	F	1		41	188	No aplica	0.3533

Figure 4 Mejores puntajes-F obtenidos por los clasificadores muestra los resultados del valor F para los mejores 3 de cada clasificador, incluyendo además los parámetros e hiperparámetros con los que se entrenó. Todos fueron entrenados usando el archivo con las características de lemas y etiquetadas la enfermedad y los rs. La "V" significa verdadero, mientras la "F" significa falso

Este resultado es corroborado al ver la matriz de confusión de este modelo de la SVM (figura 5a), donde efectivamente el clasificador supo distinguir correctamente las que no tenían ninguna asociación mientras que solo logró predecir 66 oraciones correctas. También es importante notar que la SVM clasificó 150 oraciones como incorrectas que, al ser un número relativamente bajo al compararlo con el total, se puede decir que aprendió correctamente, sin embargo cabe destacar que los datos utilizados son un pequeño grupo que posiblemente pueda no estar representando a los resúmenes que posteriormente se utilizarían para obtener las asociaciones.

Finalmente, en la última etapa de experimentos, se obtuvieron dos puntajes: 0.1628 cuando se usaban todos los resúmenes y 0.1610 cuando se filtraban únicamente las oraciones que en el primer acercamiento se habían considerado que tenían una asociación, es decir, que contenían al menos un rs y un fenotipo. Se puede observar que, el puntaje es ligeramente mayor cuando se usan todos los resúmenes , esto podría indicar que nuestra hipótesis inicial de indicar una asociación si existe rs y fenotipo es incorrecta.

Asimismo, cabe mencionar que el usar un clasificador de aprendizaje máquina sí mejora el puntaje-F de 0.1432 a 0.1628, indicando que el clasificador sí aprende más allá de simplemente tener un rs y un fenotipo en la misma oración, confirmando la hipótesis de porqué se baja ligeramente el score de 0.1628 a 0.161 . No obstante, todavía es incapaz de distinguir entre ciertos patrones y expresiones del lenguaje ya que este es un problema muy complejo.

Así que para concluir, las MLPs necesitaron más tiempo computacional para entrenarse, lo que refleja una cantidad mayor de datos a computar para ajustar este modelo y consecuentemente una mayor complejidad. Es por eso que podría pensarse que tienen el potencial de dar un mejor resultado que las SVMs, pero desafortunadamente (aunque en este se muestre un mejor desempeño de las SVM), no es posible decir que las SVM son mejores que las MLP decisivamente ni viceversa, pues no se combinó exhaustivamente todos los parámetros en ninguno de los dos, sino que se sólo los que se pensaron con mayor relevancia, y los valores usados se limitaron también a un espacio, el cual sólo fue muestreado.

a

Matriz de confusión			
		Clases predichas	
		DISEASE	OTHER
Clases Verdaderas	DISEASE	66	143
	OTHER	7	391

b

Evaluación de la mejor SVM	
Precisión	0.7913
Exhaustividad	0.7528
Valor F	0.7113

Figure 5 Matriz de confusión y resultados de la evaluación a) La matriz de confusión muestra las oraciones extraidas, en la diagonal podemos ver el núero e oraciones extraídas de manera correcta mientras que en los otros dos campos las oraciones que se confundieron. b) Los puntajes obtenidos de la mejor SVM durante la etapa de evaluación.

Conclusiones

El SVM con los hiperparámetros C= 1.6962, Gamma= 0.0169 y con balanceado fue el mejor clasificador procesando archivos separador por lemas, incluyendo palabras clave y removiendo las etiquetas gramaticales con un puntaje-F de 0.7913 en la evaluación. Por lo que ese clasificador fue utilizado para la extracción de asociaciones polimorfismo-enfermedad de 20,694 resúmenes de literatura biomédica alcanzando un puntaje-F de 0.1628. Este puntaje refleja la extracción exitosa del 32.34 por ciento de las asociaciones, permitiéndoles ponerlas en una base de datos.

No obstante a los resultados obtenidos, no se puede aseverar que siempre la mejor SVM de este estudio, va a clasificar los resultados de cualquier conjunto de oraciones con un puntaje-F aproximado al que resultó de la fase de evaluación (0.7113), pues la correcta clasificación va a depender de cada conjunto de oraciones, aunque sí se puede decir que este resultado es el más probable, asumiendo que las oraciones son representativas de la literatura biomédica.

Posiblemente se pueda llegar a tener datos más concisos si se ocupara una gama más amplia de parámetros, con más valores para cada uno y con una muestra de oraciones de evaluación más grande, de manera que los valores estadísticos de la muestra se acerquen cada vez más a toda la población.

Este clasificador puede ser utilizado de manera sencilla para lograr la extracción de cualquier tipo de asociaciones entre una enfermedad con un ID único, siendo éste de polimorfismos o de genes en cualquier tipo de literatura solo cambiando el diccionario de enfermedades del que quieras buscar, el etiquetado de los IDs y los parámetros del clasificador para que se logren ajustar mejor a los nuevos datos implementados.

Bibliografía

1. Medina, J., Garzón, F., Tafurth, P. y Barbosa, J., *Recopilación Bioinformática,*, Universidad Distrital Francisco José de Caldas,2012.
2. Montes, M. y de Lenguaje Natural, G. L. , *Minería de texto: Un nuevo reto computacional,* Laboratorio de Lenguaje Natural, Centro de investigación en computación, instituto politécnico nacional, 2014.
3. Santana Mansilla, P., Costaguta, R. y Missio, D., *Aplicación de Algoritmos de Clasificación de Minería de Textos para el Reconocimiento de Habilidades de E-tutores Colaborativos*, Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial, 17(53), 2014
4. Reséndiz, J. A., *Las máquinas de vectores de soporte para identificación en línea*, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico nacional, México, DF, 2006.
5. Anzola, N. S., *Máquinas de soporte vectorial y redes neuronales artificiales en la predicción del movimiento USD/COP spot intradiario*, ODEON, (9), 113-172, 2016.
6. Pérez Valls, J., *Herramienta matlab para la selección de entradas y predicción neuronal de valores de bolsa*, (tesis de pregrado), Escuela Superior de Ingenieros de Sevilla, Sevilla, España, 2013.
7. Gardner, M. W. y Dorling, S. R., *Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences*, . Atmospheric environment, 32(14), 2627-2636, 1998.
8. Sarle, W. S., *Neural networks and statistical models*, 1994.
9. Python.org *Welcome to Python.org. [online]*, Available at: https://www.python.org/[Accessed 9 Oct. 2017].
10. Scikit-learn.org. *). scikit-learn [online]*, Available at: http://scikit-learn.org/stable/[Accessed 9 Oct. 2017].
11. Stanfordnlp.github.io. *Stanford CoreNLP – Natural language software — Stanford CoreNLP. [online]* Available at: https://stanfordnlp.github.io/CoreNLP/[Accessed 9 Oct. 2017].
12. Niu, H., Reeves, E. y Gerstoft, P. *Source localization in an ocean waveguide using supervised machine learning.* The Journal of the Acoustical Society of America, [online] 142(3), pp.1176-1188, 2017.
13. Poursartip, B., LeBel, M., McCracken, L., Escoto, A., Patel, R., Naish, M. y Trejos, A *Energy-Based Metrics for Arthroscopic Skills Assessment* Sensors, 17(8), p.1808, 2017.
14. Turner, C., Jacobs, A., Marques, C., Oates, J., Kamen, D., Anderson, P. y Obeid, J. *Word2Vec inversion and traditional text classifiers for phenotyping lupus.* BMC Medical Informatics and Decision Making, [online] 17(1), 2017.
15. Costa, L., Gago, M., Yelshyna, D., Ferreira, J., Silva, H., Rocha, L., Sousa, N y Bicho, E. *Application of Machine Learning in Postural Control Kinematics for the Diagnosis of Alzheimer's Disease.* Computational Intelligence and Neuroscience,2016.

Material Suplementario

Todos los scripts y los archivos generados en este trabajo pueden encontrarse en el github en la siguiente liga: https://github.com/Luciagrmz/Proyecto-BioNLP/