Asistencia para la categorización de documentos

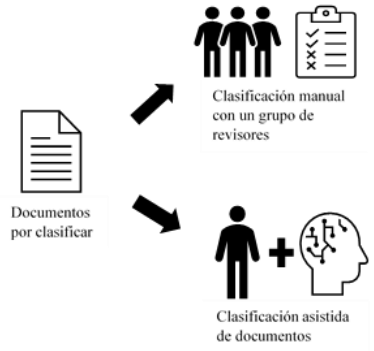
Álvaro Moncada – alvaro.moncada @javeriana.edu.co
Álvaro Pérez – perez.alvaro @javeriana.edu.co
Edwin Turizo – edwin.turizo@javeriana.edu.co

Resumen. El presente documento tiene como intención abordar la problemática de la categorización de documentos mediante el uso de métodos evolutivos (ME). Puntualmente, el objetivo consiste en utilizar un algoritmo genético que asista en la clasificación de documentos enmarcado en un sistema inteligente basado en agentes racionales. Dado el alcance de este documento, se propone una solución para la categorización de noticias utilizando algoritmos genéticos y adicionalmente se tendrán procesos de tratamiento de los datos de entrada como *tokenización* y *stop-words*. El set de datos utilizado consta de 2225 noticias de la BBC clasificadas en 5 categorías para finalmente determinar el desempeño del modelo frente a la problemática abordada.

Palabras Clave: Clasificación de documentos, Programación genética, Reglas.

I. Contextualización del problema

El descubrimiento de evidencia digital o E-Discovery, permite a investigadores o abogados identificar información almacenada electrónicamente relacionada con el objetivo de un caso. Luego de identificar aquellos archivos relevantes mediante el uso de palabras clave inicia la etapa de revisión que, usualmente, consume muchos recursos y toma mucho tiempo debido al gran volumen de información a revisar. Por lo que es relevante la implementación de técnicas que reduzcan estos tiempos y permitan a los revisores enfocar sus esfuerzos para identificar la mayor cantidad de información relevante ágilmente.



- Incremento de costos debido al alto número de revisores involucrados
- Posibles diferencias en los criterios de clasificación

- Disminución de cantidad de personas involucradas y menor costo asociado
- Homogeneidad en los criterios de clasificación

Ilustración 1. Comparación de modalidades para la revisión de documentos

*. https://www.kaggle.com/c/learn-ai-bbc/data

La **clasificación de documentos** es una problemática que ha sido ampliamente estudiada en la rama de la minería de texto dadas sus amplias aplicaciones. Esta problemática es definida de la siguiente manera. A partir de un **set de documentos de entrenamiento** $D = \{X_1, X_2, \ldots, X_n\}$, cada uno de los cuales cuenta con una "**etiqueta**" que especifica su clasificación dentro de un conjunto de valores {1, …, k}. Los documentos de entrenamiento son utilizados para construir un **modelo de clasificación**, el cual relaciona **los descriptores** *(features)* de un documento con una de las etiquetas. En un caso de prueba, el modelo de entrenamiento es usado para predecir su clasificación, esta clasificación puede ser de dos maneras *hard* y *soft*, en la primera se especifica una clasificación puntual al documento, mientras que en el segundo tipo se asigna una combinación probabilística de diferentes clasificaciones al documento.[1]

II. Análisis del estado del arte

Dado nuestro alcance, nos enfocaremos en indagar el estado del arte para soluciones relacionadas con la clasificación de noticias o de cualquier texto en general. Este tipo de soluciones constan de dos (2) etapas principales, una etapa inicial de alistamiento de los datos que consta de un **preprocesamiento y limpieza del dataset** y un posterior entrenamiento de los **algoritmos utilizados** para la clasificación de documentos [1], dado el alcance de este documento, se limitará a los métodos evolutivos.

Con respecto a los algoritmos genéticos (AG), una aplicación de estos está relacionada con la inducción de reglas que consiste en generar reglas en base al reconocimiento de patrones que se pueden extraen del conjunto de datos. En términos de clasificación de texto, uno de los campos de AG es usar reglas proposicionales que clasificaban datos en base a unas categorías definidas, basados en la lógica proposicional. Dichas reglas son de la forma:
$c \leftarrow (t1 \in d \vee \cdots \vee tn \in d) \wedge \neg (tn + 1 \in d \vee \cdots \vee tn + m \in d)$, donde c es una categoría, d es un documento y cada ti es un término del vocabulario. Se denota entonces una regla clasificadora como $Hc\ (Pos, Neg)$, donde $Pos = \{t1, \cdots, tn\}$; son los términos positivos que permiten identificar al documento y $Neg = \{tn + 1 \cdots tn + m\}$; y los términos negativos que no deberían comprender el vocabulario de esa categoría. El problema de clasificación se formula entonces como una tarea de optimización destinada a encontrar los conjuntos Pos y Neg, en donde cada individuo codifica una solución candidata (regla) que maximice el valor del fitness en el conjunto de entrenamiento y de validación.

En un primer acercamiento, Adriana Pietramala (2016) propuso el algoritmo Olex-GA, el cual se basa en generar una representación binaria eficiente de varias reglas por individuo haciendo uso de distintos datasets con el fin de clasificar documentos, además utiliza el F-Measure como función de fitness. En el caso de estudio, se manejó un tamaño de población de 500 individuos y se definió un número de

generaciones de 200, se tuvo como método de selección la opción de la ruleta y se utilizaron operadores de cruce mediante el esquema de Uniform Crossover con un crossover rate de 1.0 y adicionalmente el operador de mutación con la tasa de 0.001. La estrategia de elitismo con probabilidad de 0.2 aseguró que siempre los mejores individuos de la generación actual fueran los que evolucionaran en el proceso. En cuanto a la tasa de división de los datos, se utilizó el 70% de los datos para entrenamiento y el 30% restante para validación. Como resultados de la programación genética aplicada a este problema, se obtuvo un valor de accuracy en datos de validación de 86.40%. A partir de esto, se tomó el articulo para utilizar muchos de los parámetros del algoritmo genético en nuestra solución como lo son el número de generaciones, población, medida de fitness, entre otros, así como la interpretación de estos valores en los resultados de la solución.

En otro caso de estudio, como el que plantea De Falco et al (2001), se utilizan reglas lógicas como fronteras en un problema de clasificación con distintos datasets. El escenario plantea que la población en evolución está constituida por individuos o "programas" que representan las reglas de clasificación en forma de árboles cuyos tamaños son intrínsecamente variables en longitud. Cada regla está constituida por una serie de cláusulas condicionales, en las que se establecen condiciones sobre ciertos atributos, y por una cláusula predictiva que representa la clase. Una clase junto con su descripción forma una regla de clasificación de tipo "si descripción, entonces clase".

En uno de los datasets se buscaba categorizar el tipo de gafas, este estaba constituido por 214 instancias, 6 clases y 9 atributos numéricos relacionados con el análisis químico de impresoras de vidrio más el índice de refracción. El conjunto de datos fue dividido en 75% para entrenamiento y 25% tomados como datos de prueba. Con respecto a los parámetros de la programación genética, se estableció el tamaño de población con un valor de 200, con un número de 30 generaciones. Se realizaron 30 ejecuciones con la misma configuración inicial, pero con un nodo raíz aleatorio distinto. El mecanismo de selección ha sido mediante torneo con un tamaño de torneo de 50. Para el método de inicialización de la población, se eligio ramped half and half con el rango de profundidad del árbol de 2-5 para producir árboles con diferentes profundidades. Para evitar costes computacionales y mejores tiempos, se limitó la profundidad máxima del árbol con el valor de 7. Adicionalmente, se utilizaron los operadores genéticos de cruce con una tasa de 0.8, el operador de mutación con probabilidad de 0.1 y el operador de reproducción con la probabilidad de 0.1. Como función de fitness se estableció que se calculara el número de clasificaciones incorrectas entre el total de clasificaciones realizadas por el programa genético, denotando entonces el porcentaje de clasificaciones incorrectas realizadas por la solución evaluada. Como resultados para el dataset de *"glass"*, se obtuvo un error de clasificación del 42.63%, buscándose que este sea el valor que se minimice durante el proceso de la programación y el mejoramiento de ese fitness. A partir de este artículo, se consideró la idea de utilizar reglas lógicas de tipo *"IF*

cond THEN label" y tambien sirvió como guía en los valores de probabilidad en los operadores genéticos de crossover y mutación.

Por otro lado, para la fase de alistamiento de la información, vemos diferentes aproximaciones a esta etapa. La extracción de descriptores (*features*) para la clasificación de texto es especialmente importante para la clasificación de texto, ya que tiene esta problemática posee dos características relevantes, su alta dimensionalidad y la presencia de palabras "ruidosas", es decir que son frecuentes, pero no aportan a la tarea de clasificación. En general un texto puede ser representado de dos maneras. La primera es llamada *bag-of-words,* en la que un documento es representado como un set de palabras junto con su frecuencia asociada en el documento. Este tipo de representación es independiente del orden de las palabras. La segunda manera consiste en representar el texto como *strings,* en este caso se tiene en cuenta la secuencia de las palabras.[1]

Una de las maneras más comunes de extracción de descriptores tanto para modelos supervisados como no supervisados es el descarte de *stop-words* y *stemming.* El primero consiste en la exclusión de palabras que no son específicas para las diferentes categorías analizadas. Por otro lado, en *stemming* las diferentes formas de una palabra son consolidadas en una sola. Para nuestro caso, vamos a utilizar *stop-words*, junto con *tokenization* el cual consiste en reducir un texto a unidades individuales, que para nuestra aplicación serán palabras. [1]

III. Metodología

CRISP DM

A continuación, se presenta una descripción de cada fase y lo realizado dentro de cada subproceso que abarca la metodología CRIPS-DM 1.0 [2]:

1. **Entendimiento del negocio**
   En esta fase se busca establecer las metas y objetivos a cumplir con el proyecto de inteligencia artificial basado en los métodos de Algoritmos Genéticos. En este caso lo que se busca es clasificar noticias en un conjunto de categorías ya definidas con el fin de automatizar el proceso manual a las personas interesadas.
2. **Entendimiento de los datos**
   En esta fase se busca analizar con profundidad el *dataset* a utilizar, identificar la pertinencia de este, y los atributos y características propios, esto es, hacer un barrido sí se cumplen ciertos criterios definidos para lo que se necesita dentro del problema y con esto, la calidad con que vienen para su posterior utilización.

   Nuestra aplicación al estar relacionada con el tratamiento de texto tiene como características que es data no estructurada y tiene una alta dimensionalidad y es disperso, es decir que no todas las palabras van a aparecer en

todos los documentos. Así mismo, La calidad de la data es aceptable dado que no se identificaron campos faltantes o caracteres no legibles.

3. **Preparación de los datos**

   Se utilizó un dataset de 2.225 noticias de la BBC distribuida en 5 categorías que son: Deporte, Entretenimiento, Negocio, Tecnología y Política. Dicha base cuenta tres columnas, una es el texto de la noticia, el número de noticia y la categoría a la que pertenece, importante resaltar que se evidencia una distribución similar de las cinco categorías a lo largo del dataset. Para el procesamiento de los datos se realizó una tokenización, la cual consiste en dividir las palabras del texto de la noticia como un solo atributo del registro o segmentar las frases contenidas en la noticia, posteriormente se definió la longitud de las palabras a tener en cuenta en el modelado, en este caso se utilizaron aquellas palabras con 3 o más letras. Luego, se aplicó stopwords, lo cual permite identificar aquellas palabras que no son relevantes para clasificar el texto, suelen ser palabras frecuentes que no ayudan a segmentar la noticia a una categoría. Por último, se usó stemming con el fin de reducir palabras que tienen una misma raíz, es decir, que debido a la manera que se conjugan pueden agruparse en la misma categoría con el fin de identificar patrones de una manera más rápida y sencilla.

4. **Modelado**

   Consiste en la aplicación de la técnica a utilizar, en este caso algoritmos genéticos y para evaluar su efectividad utilizaremos como medida de desempeño el fitness y más específicamente el F-Measure, el cual consiste en combinar las medidas de precision y recall. El F-Meausre se expresa como:

$$F_{c,\alpha} = \frac{Pr_c \cdot Re_c}{(1-\alpha)Pr_c + \alpha Re_c}$$

   Se utiliza dicha medida debido a la naturaleza del problema a solucionar se debe evaluar por igual que la calidad y capacidad de clasificar correctamente la noticia. Adicionalmente, se dividirá aleatoriamente la base en registros de entrenamiento y prueba, el primero para entrenar el modelo y el segundo para validar que tan preciso es para clasificar. Para este paso se utilizó OLEX-GA que es un software para aplicar algoritmos genéticos, en el que es posible parametrizar la distribución de registros para entrenamiento y prueba, numero de atributos que en este caso son las palabras obtenidas en el preprocesamiento, numero de la población y de generaciones, tasa de cruce y mutación, etc.

5. **Evaluación**

   En primera instancia se definió como criterio de evaluación del modelo la clasificación de las noticias en alguna de las cinco categorías correctamente y como medida se seleccionó el F-Measure. Para encontrar el mejor F-Measure se manipularon los diferentes parámetros de OLEX como número de atributos, distribución entre base de entrenamiento y prueba, tamaño de

la población y de generaciones, tasa de cruce y mutación, etc. Además, se evaluará si aplicar o no stopwords tiene algún impacto en el modelo.

6. **Despliegue**
   En esta fase se busca llevar el modelo a producción de forma que utilice el modelo con datos reales, sin embargo, al ser un proyecto de investigación en el que lo que se busca es llevar a cabo la aplicación y experimentación de cada una de las técnicas principales en los distintos módulos, los resultados son presentados por medio de los distintos papers publicados.

IV. Modelo para la aplicación de los conceptos y algoritmo de la técnica IA en el caso desarrollado

Se utilizo el software OLEX-GA en la versión "Standalone" implementado en Java por investigadores del departamento de matemáticas y tecnología de la "Universidad della Calabria" en Italia [9]. A continuación, se presenta el detalle de la implementación.

Preprocesamiento

Como se ha mencionado anteriormente, se utilizó un *dataset* de 2225 noticias de la BBC que se encuentran clasificadas en 5 categorías. Antes, de poder analizar los datos mediante algoritmos genéticos, es importante realizar un preprocesamiento del texto de las noticias, el cual permita una mejor interpretación de los datos. Para esto, inicialmente se realizó un proceso limpieza del texto utilizando la herramienta RapidMiner, en el que se **tokenizaba** el texto de la notica generando un diccionario completo de todas las palabras en el dataset, luego se descartan los ***stop-words*** utilizando el vocabulario en inglés, posteriormente se aplica la técnica de ***stemming*** con el algoritmo Porter para obtener únicamente las raíces de las palabras y eliminar variantes gramaticales, y finalmente se crea la matriz binaria con la técnica *"Binary Term Frequency"* . Adicionalmente a los valores de 1s y 0s en las columnas de las palabras en cada noticia, donde 1 significa que esa palabra está presente y 0 que no, se agregaron como columnas las 5 categorías de noticias representadas con valores booleanos.

Posteriormente, fue necesario crear 5 archivos con extensión .arff (Attribute-Relation File Format), para que fuera leído por la herramienta Olex-GA que presenta un programa desarrollado en lenguaje Java con una interfaz gráfica facilitando la aplicación de algoritmos genéticos para la inducción de reglas en la clasificación de texto. Estos archivos .arff se crearon utilizando Weka 3.0 ingresándole cada archivo .csv (Separado por Coma) respectivo a la categoría de la noticia.

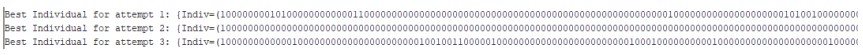
Algoritmo Genético

A partir de la revisión del estado del arte, optamos por utilizar la herramienta llamada Olex-GA, la cual está diseñada para evolucionar un set de reglas que permitan dar solución a la problemática de clasificación de documentos. Puntualmente, utiliza una representación binaria de varias reglas en un individuo (several-rules-per-individual), para luego calcular su fitness mediante la función F-Measure. Esta herramienta por debajo utiliza la API 'Jaga' para el funcionamiento del algoritmo genético.

Inicialmente, un **individuo**, es decir el conjunto de reglas para la categorización de una noticia, tiene la siguiente forma

$$c \leftarrow (t_1 \in d \vee \cdots \vee t_n \in d) \wedge \neg(t_{n+1} \in d \vee \cdots \vee t_{n+m} \in d)$$

, en la que 'c' corresponde a la categoría que va a ser clasificada, 'd' es un documento y 't' es un término del vocabulario de palabras del training set. Nótese que la regla está compuesta por dos partes, la primera, o Pos=$\{t_1,\ldots,t_n\}$ que está compuesta por t_n términos, los cuales salen de la porción de documentos de entrenamiento, y la segunda, Neg=$\{t_{n+1}, \ldots, t_{n+m}\}$ que tiene t_{n+m} términos que igualmente son escogidos del vocabulario del set de entrenamiento. Entonces, esta regla la podríamos traducir de la siguiente manera "clasifique el documento d en la categoría c si cualquiera de los términos en Pos está presente y ninguno de los términos en Neg están. Por lo que un individuo es una combinación binaria de Pos y Neg del clasificador, así mismo, se identificó que su longitud es variable, dependiendo de la cantidad de features que seleccione el usuario, a continuación, un ejemplo.



Para el cálculo del **fitness** se evaluó la calidad de la predicción de la regla, para esto se utilizó F-Measure, la cual se calcula a partir de la Precición (Prc) y el Recall (Rec) definidos de la siguiente manera:

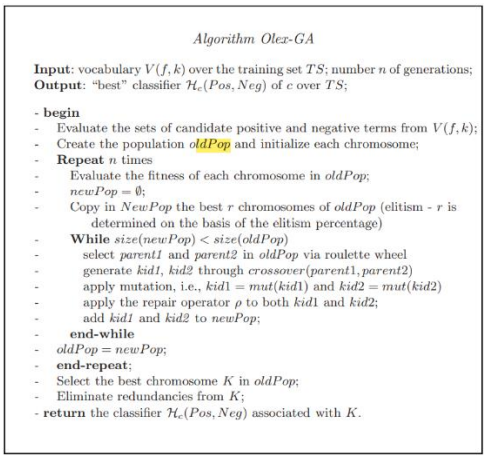
$$Pr_c = \frac{a}{a+b}; \quad Re_c = \frac{a}{a+e}; \quad F_{c,\alpha} = \frac{Pr_c \cdot Re_c}{(1-\alpha)Pr_c + \alpha Re_c}$$

Donde 'a' es el número de documentos correctamente clasificados para 'c' (i.e. TP), 'b' el número de documentos clasificados como falsos positivos para 'c', 'e' el número de falsos negativos para 'c', finalmente, tenemos a 'α' el cual se podría interpretar como el grado de importancia relativa que se le da a la Precisión y a Recall, para nuestro caso se utilizó un valor del 50%, es decir ambas son igual de importantes.

Para identificar los **features** que van a ser utilizados para la clasificación de la noticia se utilizó una función de puntuación, la cual puede ser Information Gain

(IG) o Chi Square (CHI). Dada una serie de pruebas realizadas por el equipo, la función que mejor resultados brindó para nuestra aplicación fue IG.

En cuanto a los **mecanismos de evolución**, utilizamos para la selección el método de Ruleta, operadores de Crossover y Mutación (baja probabilidad) y elitismo para asegurar que el mejor individuo pase a la siguiente generación. A continuación, un detalle del funcionamiento de Olex-GA [9].



Algorithm Olex-GA

Input: vocabulary $V(f,k)$ over the training set TS; number n of generations;
Output: "best" classifier $\mathcal{H}_c(Pos, Neg)$ of c over TS;

- **begin**
- Evaluate the sets of candidate positive and negative terms from $V(f,k)$;
- Create the population *oldPop* and initialize each chromosome;
- **Repeat** n times
- Evaluate the fitness of each chromosome in *oldPop*;
- $newPop = \emptyset$;
- Copy in $NewPop$ the best r chromosomes of *oldPop* (elitism - r is determined on the basis of the elitism percentage)
- **While** $size(newPop) < size(oldPop)$
- select *parent1* and *parent2* in *oldPop* via roulette wheel
- generate *kid1*, *kid2* through $crossover(parent1, parent2)$
- apply mutation, i.e., $kid1 = mut(kid1)$ and $kid2 = mut(kid2)$
- apply the repair operator ρ to both *kid1* and *kid2*;
- add *kid1* and *kid2* to *newPop*;
- **end-while**
- $oldPop = newPop$;
- **end-repeat**;
- Select the best chromosome K in *oldPop*;
- Eliminate redundancies from K;
- **return** the classifier $\mathcal{H}_c(Pos, Neg)$ associated with K.

Ilustración 2. Algoritmo Olex-GA

V. Protocolo experimental y análisis de los resultados obtenidos

Como se introdujo en la sección anterior, nuestro modelo base esta parametrizado de la siguiente manera:

Parámetro	**Business**	**Sport**	**Entertainment**	**Politics**	**Tech**
% Split	80	90	80	90	80
Num. Features	1000	1500	1500	500	1500
Xover Ratio	1	1	0.99	1	1
Mutation Rate	0.1	0.1	0.01	0.01	0.1
Population Size	500	500	500	500	500
Num. Generations	200	200	200	200	200
Num. Runs	5	5	3	3	5
Elitism Rate	0.2	0.3	0.2	0.2	0.2
F-Measure	63.72	75.0	62.42	63.29	55.72

Estos casos se trabajaron con 11,207 features con n-grams de tamaño 1.

A continuación, se presentan los resultados obtenidos a partir de la variación de las variables independientes, factores controlables y el respectivo valor de fitness del algoritmo genético.

Variación *% split* algoritmo genético:

% Split	Business	Sport	Entertainment	Politics	Tech
70	57.32	61.02	52.4	52.38	43.65
80	63.72	64.58	62.42	63.29	55.72
90	59.81	75.0	57.89	58.82	45.36

A partir de estos resultados, podemos ver que la cantidad de datos de entrenamiento y de prueba, resulta ser una variable importante en base a la tabla anterior que muestra en un valor medio, es decir, ni tan alto ni tan bajo, se puede llegar a que el algoritmo tenga una buena cantidad de documentos necesarios de forma que se obtenga el máximo valor de fitness y que no se sobreajuste.

Variación *Num Features* algoritmo genético:

Num Features	Business	Sport	Entertainment	Politics	Tech
100	46.32	59.52	40.31	47.3	37.93
500	56.77	66.3	51.52	63.29	41.98
1000	63.72	64.58	52.4	58.82	50.85
1500	63.06	75.0	62.42	63.29	55.72
2000	58.82	69.31	56.34	51.11	47.68

La variación del num features tiene gran impacto en los valores del F-Measure, puesto que altera el tamaño del individuo, en este caso de las reglas lógicas y en base a los resultados de la tabla, se observa que, a un mayor número, el valor del fitness tiende a aumentar, sin embargo, este no se debe exagerar puesto que con un valor muy alto el F-Score disminuye tomando características que no son importantes y no es capaz de generar reglas clasificadoras precisas.

Adicionalmente, realizamos variaciones de los parámetros Xover Ratio, Mutation Rate, Population Size, Num of Generations, Num. of Runs, sin embargo, ninguna de las categorías presentó variación en su F-Measure, es decir el fitness del modelo base para cada categoría no se alteró, los valores probados fueron los siguientes:

Xover Ratio	Mutation Rate	Population Size	Num. Generations	Num. Runs
0.8	0.01	250	100	3
0.9	0.03	500	200	5

Xover Ratio	Mutation Rate	Population Size	Num. Generations	Num. Runs
1.0	0.1	750	500	10

Sin embargo, cabe aclarar que, si por ejemplo el Mutation Rate es muy alto, la precisión de las reglas clasificadoras se ve afectada y por ende su F-Measure, en esos casos identificamos que las reglas resultantes eran de una sola palabra en la parte Pos y ninguna en la parte Neg del individuo.

A partir de la experimentación inicial de algunas de las configuraciones de Olex-GA, se identificó que este comportamiento en esos parámetros se presentó especialmente cuando utilizamos *Information Gain* como la función de puntuación (scoring function) para seleccionar las mejores *"features"* para el algoritmo, ya que si se utilizaba la medida de "*Chi Square"* si se presentaba una variación en la F-Measure. Esto es debido a que Information Gain selecciona las mejores características con una alta ganancia de información, por lo que esas variaciones no afectan directamente a las características y siempre se escogen las mismas. Con respecto a *"Chi Square"* este realiza una selección de funciones de manera supervisada pudiendo eliminar muchas características alterando la medida del fitness [8].

Aplicación de *stopwords* en el preprocesamiento de los datos

Stop Words	Business	Sport	Entertainment	Politics	Tech
Si	63.72	75.0	62.42	63.29	55.72
No	62.55	71.11	62.42	57.48	48.04

En este caso se cuenta con 11.415 features a partir de la generación de n-grams de tamaño 1.

Por otro lado, el efecto de no aplicar *stop words* durante el preprocesamiento no afecta mucho en algunos casos la precisión de nuestro sistema, por ejemplo, para las noticias de Business la precisión no se ve afectada considerablemente, por otro lado, para Tech y Polítics vemos una disminución de al menos 6 puntos porcentuales. Este comportamiento se puede deber a que al aplicar Stop words no hay una diferencia significativa del tamaño de la matriz de features (208 palábras) por lo que los individuos que están siendo evolucionados podrían simplemente no seleccionar este tipo de palabras.

VI. Conclusiones

- Los parámetros que más afectaron el F-Measure o fitness del AG fueron el % Split y el número de Features seleccionados, ya que como vimos con estos se presenta la mayor variación. Lo que nos indica que este tipo de soluciones son altamente dependientes del vocabulario seleccionado para la construcción de las reglas.

- En comparación con la solución propuesta mediante RN, esta solución toma alrededor de 1 minuto su entrenamiento, mientras que para RN su entrenamiento podría tomar alrededor de 50 minutos.
- La no implementación de stopwords no afectó en mucho el desempeño de la solución, esto se puede deber a que la matriz de features no difieren mucho, alrededor de 200 caracteres de diferencia, por lo que la mayoría de las palabras son las mismas.
- Se identificaron 5 características que aparentemente no impactaban el desempeño del AG, esto se puede deber a la función de selección de puntuación utilizada (Information Gain), ya que si esta variaba a Chi Square estos valores sí afectaban el comportamiento del AG.
- Otra diferencia en relación con los resultados de clasificación en el ejercicio de RN radica en que AG no alcanza un mejor resultado en las pruebas de entrenamiento, no obstante, al utilizar AG en el universo de prueba se llega a una mejor clasificación por categoría única.
- Los resultados obtenidos nos permiten ver que utilizando AG no se presenta niveles altos de sobreajuste.

VII. Bibliografía

[1] Charu C. Aggarwal, ChengXiang Zhai, “Mining Text Data”. Springer 2012.

[2] CRISP-DM, SPSS Inc. 2000.

[3] Adriana Pietramala, Veronica L. Policicchio, Pasquale Rullo, Inderbir Sidhu. A Genetic Algorithm for Text Classification Rule Induction. ECML/PKDD (2) 2008: 188-203.

[4] I De Falco, A Della Cioppa, E Tarantino, Discovering interesting classification rules with genetic programming (2002), DOI: https://doi.org/10.1016/S1568-4946(01)00024-2.

[5] Robu, Raul & Stefan, Holban. (2011). A genetic algorithm for classification. Recent Researches in Computers and Computing - International Conference on Computers and Computing, ICCC'11. 52-56.

[6] Mohammed I. Khaleel, Ismail I. Hmeidi, and Hassan M. Najadat. 2016. An Automatic Text Classification System Based on Genetic Algorithm. DOI:https://doi.org/10.1145/2955129.2955174.

[7] Machine Learning and Knowledge Discovery in Databases European Conference, ECML PKDD 2017, Skopje, Macedonia, September 18–22, 2017, Proceedings, Part II.

[8] H. Sulistiani y A. Tjahyanto, «Comparative Analysis of Feature Selection Method to Predict Customer Loyalty», IJOE, vol. 3, n.o 1, p. 1, may 2017, doi: 10.12962/joe.v3i1.2257.

[9]"A Genetic-algorithm for learning rule-based text classifiers - download", Mat.unical.it, 2021. Disponbile en: https://www.mat.unical.it/OlexSuite/OlexGA/OlexGA-overview.htm.