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 diferentes modelos de inteligencia artificial, para luego comparar su desempeño. Puntualmente, el objetivo consiste en utilizar 4 métodos distintos de que asistan en la segmentación de documentos enmarcado en un sistema inteligente basado en agentes racionales. Estas soluciones involucran a los conceptos de Redes Neuronales, Métodos Evolutivos, Lógica Difusa y Aprendizaje de Máquina. Dado el alcance de este documento, se propone una comparación de las soluciones para la categorización de noticias de forma que se pueda observar resultados de cada técnica y el protocolo experimental aplicado con el fin de realizar un análisis comparativo y la interpretación de resultados. Adicionalmente se tendrán procesos de tratamiento de los datos de entrada como *tokenización* y *stop-words,* entre otros. 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, Redes Neuronales, Algoritmos Genéticos, Lógica Difusa, Aprendizaje de Máquina.

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

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], que dado el alcance de este documento, se limitará a Redes Neuronales, Métodos Evolutivos, Lógica Difusa y Aprendizaje de máquina.

Redes Neuronales

Haojin Hu et al. brindan una solución utilizando RNN, específicamente Independently Recurrent Neural Network (IndRNN) combinado con LSTM con modelo de atención, esta combinación favorece la solución del problema de desvanecimiento del gradiente. Para entrenar a su solución utilizaron un set de entrenamiento de 560.000 items de DBpedia y una muestra de prueba de 70.000, que para nuestro caso el dataset es de 2225 noticias de la BBC clasificadas en 5 categorías. Como parte de su experimento compararon con 5 tipos de redes neuronales, a saber, Attention-Based Bidirectional Long Short-Term Memory Networks (AttBLSTM), Hierarchical Attention Network (H-ATT), Adversarial Training Methods for Semi-Supervised Text Classification (ADTM), Convolutional Neural Networks (CNN), Random Multi-model Deep Learning (RMDL), encontrando que los mejores resultados los obtuvo el modelo RMDL (98.82 %), y en segundo lugar se encontraba su solución IndRNN-LSTM (98.51%) [1].

Luego, profundizamos en las redes Bi-LSTM, mediante los resultados de Bingyuan Wang et al. los cuales realizaron un diseño y comparación de mecanismos para la clasificación de comentarios utilizando redes Bi-GRU y Bi-LSTM. Sin embargo, ellos utilizaron un dataset de 50.000 items de IMDB con una separación de 4/5 para entrenamiento y 1/5 para prueba, y hay una diferencia de 2% entre las soluciones [2].

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, siguiendo 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.

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. [4]

Lógica Difusa

Con respecto a Fuzzy C-Means (FCM), se decidió por esta técnica puesto que Thaung Thaung Win y LinMon [3] presentaron cómo utilizar dicha técnica para clasificar documentos de texto con clústeres o grupos predefinidos como lo es en nuestro caso, en el cual el dataset ya cuentan con las categorías definidas de las noticias. Adicionalmente, en el caso de estudio realizado por los autores previamente mencionados afirman que es una técnica eficiente y con la capacidad necesaria para manejar grandes volúmenes de datos. FCM consiste en agrupar las noticias en clusters apoyados con la determinación de si una noticia pertenece a un grupo o a otro a través del grado de pertenencia entre 0 y 1. Así mismo, cada noticia tiene características relevantes y diferenciadores que en este caso son las palabras en donde Xi es una noticia o punto en m-dimensiones y m cada palabra que compone la noticia.

FCM utiliza como fórmula de optimización

$$J_m(U,v)=\sum_{k=1}^{n}\sum_{i=1}^{c}(u_{ik})^m\left\|x_k-v_i\right\|_A^2$$, donde $$U=[u_{ik}]\in M_{fc}$$

Es una matriz de partición difusa de X, uik es la membresía del k-esimo punto de datos de la I-esima clase

$$V=[v_1,v_2,v_3,...,v_c], v_i\in \mathrm{R}^n$$ es el vector de prototipos de clusters que son predeterminados.

$D_{ikA}^2 = \|x_k - v_i\|_A^2 = (x_k - v_i)^T A(x_k - v_i)$ es la distancia Euclidiana entre cada clúster.

Los parámetros en FCM son:

- Número de clústeres: la cantidad de clústeres es representado por c, en este caso es predeterminado por las 5 categorías.
- Parámetro difuso: Representado por m, es el encargado definir qué tan difuso es la partición entre los grupos. Entre menor este valor se dice que es más difuso.
- Número de iteraciones: Cantidad de veces que se repetirá el proceso para encontrar la mejor solución.

Aprendizaje de Máquina

Para esta solución nos inclinamos por el trabajo realizado por Carson Sievert et al. [5], en el que proponen utilizar Latent Dirichlet Allocation (LDA) para visualizar e interpretar tópicos de documentos en base a los términos que mejor representan al documento, determinando el significado del documento obteniendo el término con mayor frecuencia que se adecua al documento en base al cálculo de distancias entre las características de los datos de entrada. Así mismo brinda una idea intuitiva del comportamiento del modelo de LDA aplicado para la clasificación de documentos además de la fácil interpretabilidad que brinda debido a que LDA permite inferir tanto la frecuencia de un término como su exclusividad o pertenencia a un documento en específico de una manera rápida y sencilla.

Adicionalmente se seleccionó LDA como clasificador de noticias puesto que en un artículo de Retno Kusumaningrum et al. [7], comparan dos clasificadores como lo son Naive Bayes y LDA en el cual el último tiene un mejor desempeño con respecto al primero, esto se debe a que Naive Bayes es muy rígido en la clasificación, es decir si está evaluando noticias de política y economía que son temas similares dicha técnica suele inclinarse por la determinante afectando la distribución y por ende disminuyendo la precisión en la clasificación, otra ventaja de esta técnica es que es flexible para parametrizar sin mayor complejidad.

Por otro lado, se tiene otro artículo por parte de Yiqi Bai et al. [8], el cual enfoca su investigación en la clasificación de noticias utilizando LDA afirmando que es un método no supervisado y que por sí solo no puede etiquetar las categorías. Dentro de este artículo se hace uso de una matriz de presencia o ausencia de términos, que se asemeja el Binary Term Occurrences (BTO), indicando si 1 si el termino se encuentra en la noticia y en caso contrario 0. Adicionalmente, se hace uso de categorizar el tópico de la noticia en base a las probabilidades más altas obtenidas por el proceso de LDA teniendo en cuenta la distribución del documento en cada una de las categorías. Para la medida del desempeño de la clasificación se hace uso de las métricas de Recall y Precision que indican los True Positive, True Negative, False Negative y False Positive en base a la matriz de confusión de clasificación.

Métricas de Desempeño

Por otra a partir de la investigación del estado del arte de las diferentes técnicas, una manera común de medir el desempeño del clasificador es utilizando la medida F-Measure, que está dada por la siguiente ecuación.

$$F-Measure_i = \frac{2 \bullet \Pr ecision_i \bullet \operatorname{Re} call_i}{\Pr ecision_i + \operatorname{Re} call_i}$$

Este valor se calcula para cada categoría i, con el fin de determinar el error de clasificación que se puede tener en los grupos de noticias generados.

En donde la Precisión se calcula con la fórmula $\mathbf{Precision_i = CN_i / JN_i}$,

Y el Recall se calcula con la fórmula $\mathbf{Recall_i = CN_i / RN_i}$, definiéndose:

CNi: Número de documentos juzgados de forma correcta en la categoría i,
JNi: Número de documentos juzgados en la categoría i,
RNi: Número de documentos pre-etiquetados en la categoría i
En cuanto al F-Measure General, este se define de la siguiente manera:

$$Overall\ \ F-Measure = \frac{\sum_{i=1}^{C}(RN_i \bullet F-Meausre_i)}{\sum_{i=1}^{C} RN_i}$$

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 [6]:

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 cuatro métodos de IA propuestos. 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, sim embargo, la cantidad de noticias del dataset es mucho menor al tamaño del

vocabulario resultante por lo que se hace necesario aplicar un preprocesamiento al mismo.

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**, **stopwords**, **stemming**, finalmente, 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.
   Así mismo, dada la alta dimensionalidad se aplicó **Information Gain**, para seleccionar las palabreas que pudieran aportar a clasificar mejor las noticias.
4. **Modelado**
   Consiste en la aplicación de la técnica a utilizar, en esta fase se aplicaron los 4 modelos de IA solicitados y para evaluar su efectividad utilizamos como medida el **F-Measure**, el cual consiste en combinar las medidas de *precision* y *recall*. 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. La siguiente sección brinda detalles de los modelos.
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 las diferentes herramientas utilizadas para la implementación de los diferentes clasificadores, los cuales se detallan en la sección V del documento.
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.

IV. Modelo para la aplicación de los conceptos y algoritmo de las técnicas IA

Para la implementación de los modelos se utilizaron varias herramientas y se mantuvo las mismas características de preprocesamiento, en la medida en que la herramienta y la técnica lo permitiera.

Para esto, se implementó una etapa de preprocesamiento de las noticias para pulir la información, y posteriormente alimentar los diferentes algoritmos de IA. A continuación, se detallan los procedimientos y soluciones realizadas.

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 los algoritmos descritos anteriormente es importante realizar un preprocesamiento del texto de las noticias, el cual permita una mejor interpretación de los datos. Para esto, se implementó el siguiente modelo.

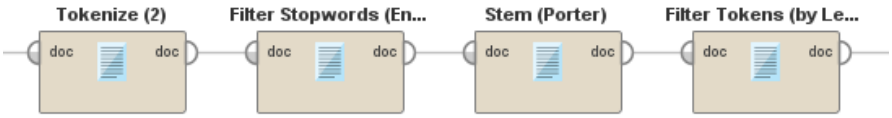


Ilustración 2. Preprocesamiento de noticias

Inicialmente, se **tokenizó** 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, para finalmente **filtrar** aquellas palabras que tengan menos de 2 letras, ya que estas generalmente no aportan para la clasificación.

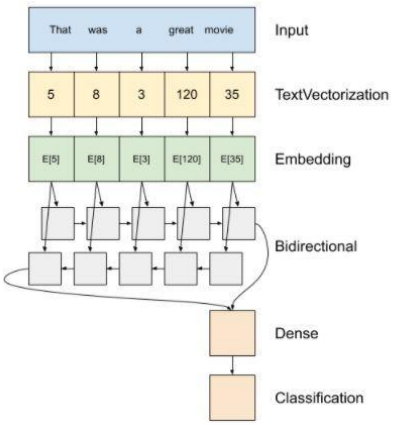
Esto da como resultado un vector de palabras la cual se genera utilizando *"**Binary Term Ocurrences**"*. Este vector de palabras cuenta con valores de 1s y 0s en las columnas de las palabras en cada noticia (filas), donde 1 significa que esa palabra está presente y 0 que no. La dimensión de la matriz en este punto fue de 11.132 features o palabras por 2225 noticias.

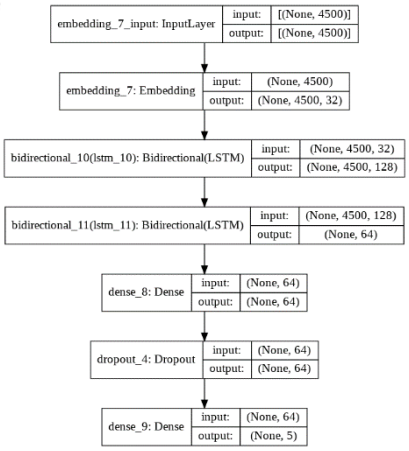
Finalmente, dependiendo del modelo evaluado, seleccionamos una cantidad de palabras que optimizara el desempeño del modelo, para esto, seleccionamos aquellas palabras con el mayor **Information Gain**, este paso fue necesario incluirlo debido a que, dependiendo del modelo evaluado, si utilizábamos todo el universo de palabras la precisión de la clasificación tendía a disminuir. A partir de este punto, obtenemos una matriz BTO (Binary Term Ocurrences), es decir, una matriz de 1 y 0 que indican si una palabra está o no presenta en el corpus de la noticia, respectivamente.

Modelos

Redes Neuronales

En base al estado del arte para las redes neuronales, se decidió implementar una **RNN Bi-LSTM**, para la que se definió el **entrenamiento-prueba con 80-20**, respectivamente utilizando la **disminución del gradiente** con un ***learning rate*** de 0.01 con un número de 5 **epocas**. La RNN Bi-LSTM implementada cuenta con **5 capas**. Dos de las cuales son densas y cuentan con **funciones de activación** de *Tanh* y *Softmax respectivamente,* esta última permite decir si una noticia pertenece a una de las 5 posibles clasificaciones mediante un vector codificado de 1s y 0s. Por otra parte, se aplicó un ***dropout* de 0,5** para las capas densas para disminuir el *overfitting* de la red. A continuación, se presentan unas gráficas que ilustran la RNN Bi-LSTM implementada.





Métodos Evolutivos

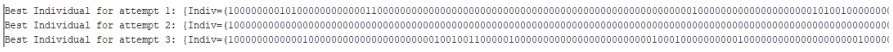
A partir de la revisión del estado del arte, se optó por utilizar la herramienta llamada **Olex-GA**, la cual está diseñada para evolucionar un set de reglas que permitan dar solución a la problemática de clasificación de documentos. Puntualmente, utiliza una representación binaria de varias reglas en 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,…,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}, …, 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, en este caso Information Gain (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.

Lógica Difusa

A partir de la revisión del estado del arte, optamos por utilizar el algoritmo Fuzzy C-Means de la herramienta RapidMiner, la cual permite dar solución a la problemática de clasificación de documentos brindando un grado de pertenencia difuso a los grupos de cada documento y mediante el cálculo del grupo con mayor grado de pertenencia defuzzyfica. Para esto se utilizó el siguiente diagrama de flujo.

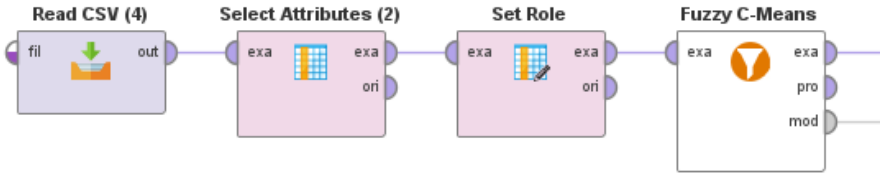


Ilustración 3. Modelo clasificador utilizado

Este flujo inicia con la importación de la matriz de [11.131 x 2225] palabras vectorizadas. Luego, se seleccionan los 100 atributos o palabras con mayor Information Gain, para luego pasar al modelo Fuzzy C-Means, pero antes fue necesario asignar un rol de clasificación a la columna que contiene los labels, esto para mayor facilidad de representación de los resultados.

Luego de revisar el estado del arte se inició una serie de iteraciones para identificar los parámetros que optimizaran la salida del clasificador. Para esto se utilizaron 5 clusters, dado que el dataset de noticias está dividido en 5 categorías; el número de iteraciones se mantuvo por defecto en 50; el parámetro de fuzzyness, indica qué tan difusos van a ser los conjuntos, se manejó el valor por defecto de 2, es decir si este aumenta la mayoría de los documentos quedaban clasificados en un solo grupo; otro parámetro fue MinGain, el cual se varió hasta encontrar aquel que diera mejores resultados, que para nuestro caso fue de 5e-4.

A continuación, una imagen de la configuración del que sería nuestro modelo base.

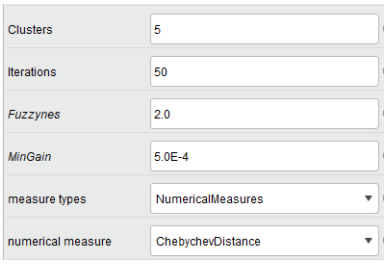


Ilustración 4. Configuración de Fuzzy C-Means

Por otro lado, luego de realizar pruebas con varios tipos de medidas (Measure Types) la que mejores resultados dio para nuestra problemática fue Chebychev Distance.

Aprendizaje de Máquina

LDA es un modelo probabilístico en el que cada documento se representa sobre un conjunto de temas latentes y cada uno de los temas es representado como una distribución sobre un vocabulario, dicha técnica inicia identificando las variables latentes incluida la distribución de palabras para cada temas y proporción de temas para cada documento y con un conjunto de palabras como variables observadas.

De acuerdo al estado del arte decidimos utilizar como herramienta para aplicar esta técnica a Python la cual cuenta con una librería dedicada a ello que es sklearn.decomposition, la cual es fácil de implementar puesto que se requiere como entrada las palabras relevantes para la clasificación del texto y como están distribuidas a lo largo de cada noticia siendo esta cantidad de palabras una de las variables con las cuales se parametriza el modelo, importante resaltar que para identificar estas

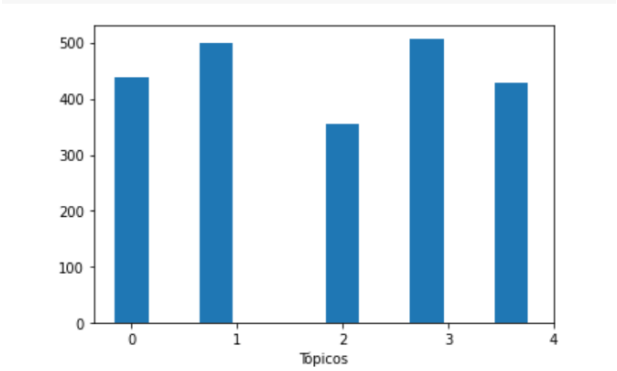
palabras se tuvo que realizar el preprocesamiento mencionado en la sección anterior. Luego de ello y apoyados en el estado del arte se manipularon los diferentes parámetros como iteraciones, cantidad de palabras, distribución de topics por documento, distribución de topics por palabra. En relación con estas dos últimas variables se utilizan para determinar qué tan distantes van a estar las palabras que permiten inferir el tema de la noticia y los temas de las noticias en si respectivamente. Posteriormente de parametrizar y de ejecutar los resultados son probabilidad de ser un tema u otro y de acuerdo con ello el modelo define cual es la clasificación de la noticia efectiva.

	topics0	topics1	topics2	topics3	topics4	Topico_dominante
doc0	0.000139	0.000139	0.026651	0.000139	0.972932	4
doc1	0.000357	0.998574	0.000357	0.000357	0.000357	1
doc2	0.000434	0.000434	0.000434	0.998265	0.000434	3
doc3	0.000434	0.000434	0.000434	0.998265	0.000434	3
doc4	0.000384	0.000384	0.998464	0.000384	0.000384	2
...	...	...	...	...	...	...
doc2220	0.000303	0.894130	0.000303	0.000303	0.104962	1
doc2221	0.998100	0.000475	0.000475	0.000475	0.000475	0
doc2222	0.000344	0.381121	0.617846	0.000344	0.000344	2
doc2223	0.691613	0.101396	0.000285	0.000285	0.206420	0
doc2224	0.000454	0.000454	0.000454	0.998186	0.000454	3

2225 rows × 6 columns

Ilustración 5. ejemplo de matriz de probabilidades por topic y documento

El siguiente resultado es la distribución de las categorías en las que clasifico para culminar en el cálculo de la métrica de desempeño.



V. Análisis de los resultados obtenidos

Redes Neuronales

A continuación, se presenta el modelo base final obtenido durante el desarrollo de la solución al proyecto planeado, detallando los principales parámetros que se utilizaron.

Parámetros	Valor
Epochs	5
Learning Rate	0.01
Neuronas por capa	(64, 32, 64, 5)
Dropout	0.5
Split rate (Train-Test)	80% - 20%
Tiempo de entrenamiento por epoca	Aproximadamente 10 minutos
Accuracy	49.66%

Métodos Evolutivos

Dado que la herramienta solo permitía evolucionar las reglas de un clasificador a la vez, presentamos las características del modelo que brindan un mejor desempeño.

Parámetro	Business	Sport	Entertain-ment	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. Genera-tions	200	200	200	200	200

Parámetro	**Business**	**Sport**	**Entertain-ment**	**Politics**	**Tech**
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
F-Measure Gene-ral	64.26%				

Lógica Difusa

Para este modelo utilizamos la técnica de Fuzzy C-Means en la herramienta de Rapidminer, los parámetros con los cuales configuramos el clasificador fueron:

Parámetro	**Valor**
Número Clusters	5
Fuzzyness	2.0
Min Gain	0.0005
Iteraciones	500
Métrica Distancia	Chebyshev
F-Measure General	38.0%

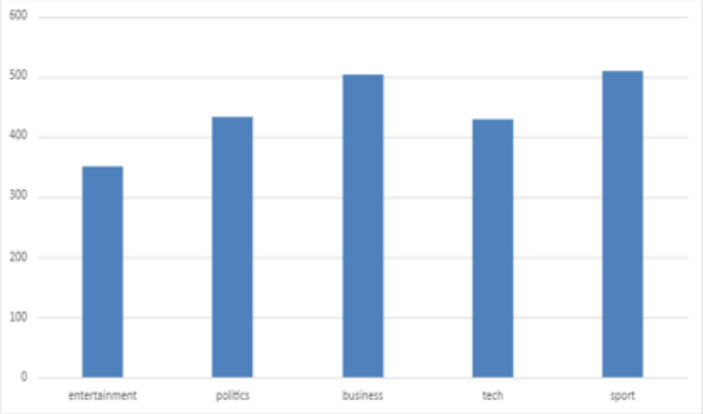
Aprendizaje de Máquina

Para esta solución utilizamos LDA (Latente Dirichlet Allocation) utilizando Python y librerías de scikit-learn, y los parámetros que mejor resultado brindaron fueron los siguientes:

Parámetro	**Valor**
Topics	5
Iteraciones	100
Distribución de Topics por documento	0.1
Distribución de topics por palabra	0.1
Corpus	500 palabras
F-Measure	70.2 %

A continuación, se presentan los resultados y análisis de la aplicación de esta técnica con mayor profundidad

Como primer resultado se obtuvo la gráfica de la distribución de las clasificaciones realizadas por la técnica LDA, de forma que la categoría “sports” fue la que tuvo el mayor número de noticias categorizadas en ese tópico. Por otro lado, “entertainment” fue la que tuvo un menor número de noticias clasificadas.



Adicionalmente, se presenta la matriz de confusión que representa el número de "True Positive", "False Negative", "True Negative" y "False Positive" que se tuvieron durante la clasificación de los distintos tópicos de noticias.

	business	**entertainment**	**politics**	**sport**	**tech**	**Total**
business	**351**	37	39	40	36	503
entertainment	20	**267**	23	16	25	351
politics	391	38	**289**	29	38	433
sport	37	30	36	**3762**	30	509
tech	36	48	34	32	**2795**	429
Total	483	420	421	493	408	2225

Observando detalladamente, se puede interpretar que la clasificación hacia la categoría de noticias de "sports" fue la que tuvo un mejor F-Measure de clasificación con un valor de 75%, y que "tech" obtuvo un error de clasificación de 66% siendo el porcentaje con menor valor entre las otras categorías.

Pruebas experimentales

El proceso se trabajó con las 2225 noticias y se acortaron a únicamente 100 palabras teniendo como base su information gain obtenido de la matriz binaria de términos, que establecía sí el termino textual aparecía en cada noticia y en base a esto

se escogieron los mejores 100 términos que daban mayor significado en la categorización.

Técnica utilizada	Parámetros	Conclusión	Desempeño
Redes neuronales	Épocas: (5,7,12)	A un mayor número de época, la red neuronal tiende a sobreajustarse a los datos de prueba	Accuracy: 49.66%
	Learning Rate: (0.5,0.1,0.01,0.005)	A mayor tasa de aprendizaje, el algoritmo pierde precisión puesto que no converge a una correcta solución	
	Cantidad de neuronas: ([128,64,128], [64,32,64], [24,16,24])	A un mayor número de neuronas el algoritmo gana precisión, sin embargo, aumenta el tiempo de entrenamiento.	
	Stop Words: (Si,No)	Al no aplicar stopwords, la precisión y los valores de perdida disminuyen en gran medida.	
	Split Rate: ([80-20], [70-30])	Al tener un mayor número de datos de validación, la precisión de la red aumenta.	
Métodos Evolutivos	% Split: ([70-30], [80-20], [90-10])	Se debe escoger un valor medio, teniendo una cantidad adecuada de datos de entrenamiento y prueba.	F-Measure: 64.26%
	Num Features: (100,500,1000,1500, 2000)	Altera el tamaño del individuo, a un mayor número el F-measure tiende a aumentar, sin exagerar con altos valores.	

	Xover Ratio (0.8,0.9,1.0)	No se vio alteración en el valor del F-Measure	
	Mutation Ratio (0.01,0.03,0.1)		
	Population Size (250,500,750)		
	Num Generations(100,200,500)		
	Num Runs (3,5,10)		
	Stop Words: (Si, No)	No tiene una mayor afectación en muchas de las ejecuciones con distintas categorías de noticias	
Lógica Difusa	Fuzzyness: (2,2.5,3,5,10)	Un valor alto hace que se difuminen las clases, los elementos tienden a pertenecer a todos los grupos, lo que disminuye el F-Measure	F-Measure: 38%
	Min Gain: (1,0.5,0.0005,0.00005,0.000005)	Con un valor muy pequeño, los valores de las funciones de membresía no se ven afectados puesto que no se tiene una ganancia considerable.	
	Número Iteracciones: (50,100,200,500,1000)	No se vio alteración el valor del F-Measure.	
	Stop Words: (Si,No)	Si afectó en la generación de clusters, disminuyendo el F-Measure general debido a palabras que existían en múltiples noticias.	
	Número Iteraciones: (10,50,100,200,500)	No se vio alteración el valor del F-Measure.	

Aprendizaje de Máquina	Tamaño Corpus: (100,200,500,1000)	Cuando aumenta el tamaño del corpus, se va teniendo un mejor F-Measure, sin embargo, con valores muy grandes disminuye el rendimiento.	F-Measure: 70.2%
	Stop Words: (Si, No)	No se vio alteración el valor del F-Measure.	

VI. Discusión final

- El método que mejor clasificó las noticias fue LDA con un 70.2% de F-Measure, aquel que tuvo el más bajo desempeño fue Lógica Difuza con un 38%
- La utilización de Information Gain para seleccionar las features que alimentaban los modelos ayudo a mejorar el desempeño de los clasificadores utilizando BTO. Sin embargo, utilizar otras soluciones, como TF o IF-IDF brindan un mayor IG en las palabras debido a que se tiene una frecuencia del número de palabras y no solo un dato binario.
- El modelo que más rápido fue entrenado fue AG y el más lento fue RN, sin embargo, una vez implementado puede ser más ágil.
- Las noticias de tipo Sport fueron consistentemente las que mejor desempeño tuvieron en cuanto a su F-Measure.
- El modelo de Redes Neuronales es el más sensible a variaciones de aplicar o no Stop Words.
- Para futuros trabajos, se podría utilizar otras metodologías para vectorizar el vocabulario de las noticias como TF-IDF o Word2vec.
- Así mismo, utilizar una base de datos más grande permitiría entrenar mejor los modelos.

VII. Referencias Bibliográficas

[1] Haojin Hu , Mengfan Liao , Chao Zhang , Yanmei Jing, “Text classification based recurrent neural network”. IEEE Xplore 2020.

[2] Bingyuan Wang, Fang Miao, Xueting Wang, Libiao Jin , “Text Classification Using a Bidirectional Recurrent Neural Network with an Attention Mechanism” IEEE Xplore 2020

[3] Thaung Thaung Win. Mon, Lin. "Document clustering by fuzzy c-mean algorithm," 2010 2nd International Conference on Advanced Computer Control, 2010, pp. 239-242, doi: 10.1109/ICACC.2010.5487022.

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

[5] Carson Sievert, Kenneth E. Shirley. “LDAvis: A method for visualizing and interpreting topic” Proceedings of the Workshop on Interactive Language Learning, Visualization, and Interfaces, pages 63–70,Baltimore, Maryland, USA, June 27, 2014.

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

[7] Retno Kusumaningrum, Ihsan Aji Wiedjayanto, Satriyo Adhy, Suryono. “Classification of Indonesian News Articles based on Latent Dirichlet Allocation”. 2016. IEEE Xplore

[8] Yiqi Bai, Jie Wang. “News classification with Labeled LDA”. 2015.