Clasificación de eventos en bitácoras de perforación de pozos petroleros empleando técnicas de clasificación de textos

Events classification from oil wells drilling logs using text classification techniques

William Feal Delgado[1], Manuel Antonio Chi Chim[2], Claudia Noguerón González[1], Francisco Javier Cartujano Escobar[1]

[1]Instituto Tecnológico de Zacatepec. Calzada Tecnológico No. 27, Zacatepec de Hidalgo, Morelos. C.P. 62780
[2]Instituto Mexicano del Petróleo. Eje Central Lázaro Cárdenas Norte 152, San Bartolo Atepehuacan, Ciudad de México. C.P. 07730

* Correo-e: billyfeal@gmail.com

PALABRAS CLAVE:

Clasificación de textos, aprendizaje supervisado, clasificadores, enfoque de envoltura.

RESUMEN

Uno de los procesos de mayor importancia en la exploración y explotación de hidrocarburos es la perforación de pozos. Los costos asociados al proceso son muy altos por lo que las compañías que desarrollan esta actividad buscan estrategias que les permita disminuir los tiempos de perforación de sus pozos, garantizando de esta forma la reducción de los costos. Una forma de lograr la reducción de los tiempos de perforación es tener la posibilidad de predecir o detectar eventos que ocasionen retrasos. Teniendo en cuenta esta problemática, en este trabajo se aplicaron técnicas de clasificación de texto y aprendizaje automatizado para clasificar los eventos que se registran en las bitácoras del Sistema de Información Operativa de Perforación (SIOP) de PEMEX Exploración y Producción. Para tratar con el problema de la alta dimensionalidad presente en este tipo de proceso de clasificación de texto se empleó un enfoque de envoltura que utiliza un algoritmo genético como herramienta para la selección de características.

KEYWORDS:

Text classification, supervised learning, classifiers, enveloped approach.

ABSTRACT

Wells drilling is one of the main processes of oil exploration and exploitation. This activity allows to know the delimitation and prospectivity of oil deposits, and also permits this resource extraction. The costs associated with oil exploration and exploitation process are high for the companies that develop this activity, for that reason those companies look for strategies that allow them to reduce the wells drilling times. One way to achieve the drilling's time reduction, is by predicting or detecting possible events that may cause delays in the oil drilling process. Attending this situation, text mining and machine learning techniques were applied to classify the events recorded in the Information System Logbooks (SIOP) of PEMEX Exploration and Production. The main goal of the present work was to obtain a high-impact application on the national oil industry. By the autonomous classification of events is possible to reduce time and effort that engineers and technicians traditionally spent in this activity. To deal with the high dimensional problem present in text classification process, was used an envelope approach that uses a genetic algorithm as a tool for the selection of characteristics.

Recibido: 31 de julio 2018• **Aceptado:** 30 de julio de 2019• **Publicado en línea:** 28 de febrero de 2020

1. INTRODUCCIÓN

En los últimos años las técnicas de Minería de Datos han tenido bastante éxito en áreas asociadas a las ciencias y los negocios, pero su aplicación en la Industria del Petróleo y Gas está todavía en una etapa inicial. Esto está dado en gran medida por las características específicas de esta industria, que genera enormes cantidades de diferentes tipos de datos con diferentes niveles de precisión y una gran incertidumbre. Pese a estas dificultades, el potencial de aplicación de estas técnicas es muy prometedor, en este sentido muchas empresas dedicadas a la Exploración y Producción de Hidrocarburos se han visto forzadas a explotar sus bases de conocimientos para poder cumplir con las presiones regulatorias, competitivas y económicas propias de la industria de los hidrocarburos [1].

México se caracteriza por ser un país productor con alta dependencia de los recursos obtenidos a partir de la producción de energía proveniente del petróleo, al grado que casi el 20% de los ingresos presupuestarios son aportados por Petróleos Mexicanos [2]. Es necesario mencionar que actualmente México se encuentra ante un panorama donde la actual producción de crudo ha venido disminuyendo de 3.8 millones de barriles diarios (Mbd) en 2004 a 2.5 Mbd en 2016 [3].

Uno de los procesos de gran importancia en la exploración y explotación de hidrocarburos es la perforación de pozos; esta actividad posibilita la confirmación, delimitación y prospectividad de yacimientos, así como la extracción de los recursos. Los costos asociados al proceso son muy altos; según un reporte de la Cámara de Petróleos de Colombia (CAMPETROL), basado en un estudio que realizó en América Latina, los costos de perforación en la región son elevados comparados con otras áreas del planeta [4]. En el caso de México se han desarrollado estrategias para revertir la situación de los altos costos de perforación. En este sentido PEMEX se ha propuesto impulsar soluciones innovadoras basadas en el empleo de tecnologías de punta y la aplicación de novedosos métodos computacionales como parte de una trasformación digital en la empresa que permitirá generar ahorros y mejorar la eficiencia operativa en todas las líneas de negocio. Un ejemplo palpable de esta estrategia es la organización del primer foro tecnológico "*Drive Oil & Gas*" [5].

El Instituto Mexicano del Petróleo (IMP) también se une a este esfuerzo, con resultados a destacar como el desarrollo de la metodología para el análisis de los tiempos de perforación [6]. A través del análisis de los tiempos empleados en el proceso de perforación se puede evaluar el estado final de un pozo petrolero, identificando las mejores prácticas, métodos y tecnologías que fueron empleadas para transmitirlas a pozos subsecuentes; con el objetivo de minimizar los costos. La metodología compara los tiempos de las actividades programadas con los tiempos de las actividades ejecutadas o reales, colectadas en las bitácoras del Sistema de Información Operativa de Perforación (SIOP) de PEMEX Exploración y Producción, para encontrar las desviaciones en el plan de perforación. De acuerdo a esta metodología, los tiempos reales de perforación se clasifican en: tiempos normales (programados y no

programados) y tiempos no productivos (problemas y esperas). La información generada en estas bitácoras es clasificada de forma manual por los especialistas del IMP. La rápida clasificación de estos eventos tendría un gran impacto en el proceso de prospección petrolero pues permitiría realizar de manera más eficiente la detección de fallas o errores en las operaciones. Atendiendo a esta problemática sería de gran ayuda poder contar con herramientas computacionales capaces de clasificar de forma autónoma los tiempos reales de cada uno de los eventos registrados en las bitácoras de perforación.

2. METODOLOGÍA

Para la realización de este trabajo se decidió emplear el enfoque de clasificación de textos (TC, del inglés Text Classification) basada en aprendizaje de máquina supervisado, ya que se cuenta con un conjunto de documentos clasificados por expertos que se emplearán como base para el entrenamiento de los clasificadores. Este enfoque presenta una gran efectividad y se puede emplear con éxito para remplazar procesos de clasificación manuales como el del presente caso de estudio [7]. La figura 1 muestra un diagrama general del proceso desarrollado.

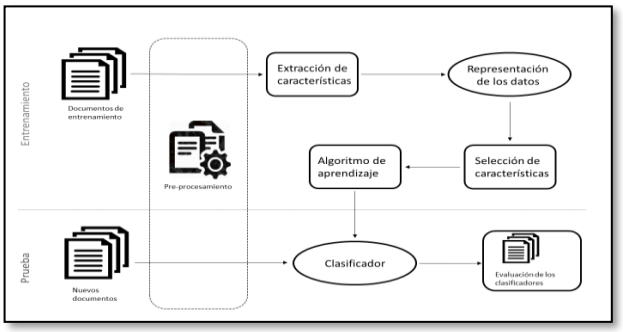

Figura 1. Esquema del proceso de TC empleado.

Los corpus de documentos de entrenamiento y validación provienen de los registros de las bitácoras del Sistema de Información Operativa de Perforación (SIOP) de PEMEX Exploración y Producción. Estos se someten al proceso de extracción de características para su representación vectorial, donde cada documento d_i, de la colección de N documentos se representa con el conjunto de las características de la colección [8]. Estas características pueden tener diversos criterios de formación, aunque ha sido demostrado que los términos o palabras son la mejor unidad tanto para la representación como para la clasificación [9]. La forma de representación del problema será una matriz $A = (w_{ik})$ que representa los valores de las características para la colección de documentos donde los valores de los elementos w_{ik} representan el peso que tiene una característica en cada documento [8]. Para el cálculo de los valores w_{ik} se empleará la función estándar denominada TFIDF (del inglés, *Term Frequency Inverse Document Frequency*)[10]:

$$tfidf(t_k, d_j) = \#(t_k, d_j) * \log \frac{|T_r|}{\#T_r(t_k)}, \quad (1)$$

donde $\#(t_k, d_j)$ denota el número de veces que un término o característica t_k, aparece en el

documento d_j y $\#T_r(t_k)$ representa la cantidad de documentos T_r en que aparece t_k. Para la normalizar los valores de la función TFIDF en el intervalo [0,1] y poder representar todos los documentos con vectores de igual tamaño se emplea la siguiente expresión [7]:

$$w_{kj} = \frac{tfidf(t_k,d_j)}{\sqrt{\sum_{s=1}^{|\tau|}(tfidf(t_k,d_j))^2}}, \quad (2)$$

En TC uno de los aspectos que mayor complejidad adicional es la gran cantidad de términos que se emplean para la clasificación y que está relacionado de forma directa con la cantidad de palabras o términos que contienen los documentos de estudio. La reducción de la dimensionalidad evita lo que en TC se conoce como "sobreajuste", que no es más que el ajuste del clasificador a las características de los datos de entrenamiento [7]. Este proceso se realiza de forma general mediante técnicas como la eliminación de palabras auxiliares (*stopwords*) y la reducción de familias de palabras a su vocablo raíz (*stemming*)[11].

Para la construcción del clasificador se decidió utilizar diversos algoritmos que de forma habitual se citan en la literatura especializada con muy buenos resultados: Naïve Bayes (NB), Máquinas de soporte vectorial (SVM, del inglés *Support Vector Machine*), Redes neuronales (NN, del inglés *Neural Network*), Vecinos próximos (kNN, del inglés *k Nearest Neighbors*) y Bosques aleatorios (RF, del inglés *Random Forest*) [7][12][13]. La evaluación de estos clasificadores se realiza de forma experimental, lo que permite medir la efectividad de los mismos para realizar una correcta clasificación. Para evaluar los resultados se emplea una matriz de confusión, que permite calcular dos indicadores que son ampliamente empleados en las tareas de TC: memoria (ρ) y precisión (π). [7]:

$$\rho = \frac{VP}{VP+FN}, \ \rho \in [0,1], \quad (3)$$

$$\pi = \frac{VP}{VP+FP}, \ \pi \in [0,1], \quad (4)$$

Estos indicadores por si solos no representan una medida apropiada para evaluar un clasificador por lo que es necesario lograr un modelo que maximice ambas métricas simultáneamente. Para lograr este objetivo se empleará la métrica *F-Sore* (F_1) que se puede determinar con la siguiente expresión [14]:

$$F_1 = \frac{2\rho\pi}{\rho+\pi} = \frac{2*VP}{2*VP+FP+FN}, \ F_1 \in [0,1], \quad (5)$$

3. EXPERIMENTACIÓN Y RESULTADOS

Para el proceso de TC se crearon dos archivos de datos que contienen la información relevante para la clasificación. Estos archivos son del tipo separados por coma (CSV) y contienen tres columnas de datos: "Actividad", "Tipo de operación" y "Tiempos no productivos (TNP)". Esta última columna de datos fue empleada como etiqueta para la clasificación pues tiene un valor binario [0,1]; donde el cero (0) representa los eventos de naturaleza productivos (TP) y el uno (1) los eventos de naturaleza no productivos (TNP) (Figura 2).

El primero de los archivos, destinado al entrenamiento del modelo de clasificación, contiene 11642 registros de operaciones proveniente de los archivos SIOPs. Del total de eventos registrados 3990 corresponden a eventos de TNP, producto de operaciones clasificadas como problemas (P) o esperas (E). El resto de los registros (8252) corresponden a operaciones normales por lo que se clasifican como eventos de TP.

La disparidad entre la cantidad de eventos TP y NTP evidencia que en los datos de entrenamiento existe un gran desbalance entre las clases o categorías, añadiéndose complejidad adicional al problema de TC. Para contrarrestar los efectos de un posible sobreajuste de los clasificadores, debido al desbalance de las clases, se creó un archivo para la validación más balanceado que contiene un total de 2044 registros de operaciones. Del total de eventos registrado 964 pertenecen a operaciones con tiempos de naturaleza no productiva (TP) y 1080 a eventos de operaciones productivas (TP)

ACTIVIDAD	TIPO DE OPERACIÓN	TNP
CON BNA A 125 M PROBO HTA DE CIA WEATHERFORD CON 85 EPM/350 GPM/950 PSI	N	0
CON BNA. PDC 6 3/4" Y SARTA ROTATORIA (RSS) CON MWD ; LWD DE CIA WTF PERFORA A	N	0
CON BNA. PDC 63/4" Y SARTA ROTATORIA (RSS) CON MWD ; LWD DE CIA WTF CONTINUAR	N	0
CON BNA. PDC 63/4" Y SARTA ROTATORIA (RSS) CON MWD ; LWD DE CIA WTF PERFORA D	N	0
SACA BNA. PDC 63/4" Y SARTA ROTATORIA (RSS) CON MWD ; LWD DE CIA WTF DE LA PRO	N	0
ALA PROF 2371 MTS SUSPENDE POR FALLA EN BBA NUM1 DESTAPA MODULO PARA CHECA	P	1
ALA PROF 2371 MTS SUSPENDE POR FALLA EN BBA NUM1 DESTAPA MODULO PARA CHECA	P	1

Figura 2. Archivo de datos.

Atendiendo a lo descrito en [9] se escogieron los términos como las características para la representación vectorial de los documentos. Previo a este proceso se realizó la normalización de los textos por lo que se eliminaron los acentos ortográficos, los números y todos los caracteres especiales; además todas las palabras fueron convertidas a minúsculas.

Para evaluar el desempeño de los clasificadores se utilizaron tres configuraciones diferentes. En la primera se utilizó la representación vectorial del corpus de entrenamiento con su texto original normalizado. Para la segunda configuración se eliminaron las palabras auxiliares del lenguaje (*stop words*), ya estos términos tienen un valor de TFIDF alto, lo que implica que su peso en la clasificación no es significativo. En la tercera configuración se aplicó un algoritmo con el fin de reducir familias de palabras a su vocablo raíz y así reducir la dimensionalidad del problema (*stemming*). Siguiendo la metodología de trabajo después de la extracción de características, los documentos de entrenamiento se representaron vectorialmente para entrenar los algoritmos de clasificación seleccionados. Como último paso se emplearon los documentos de validación para comprobar el desempeño de los clasificadores obtenidos. Estas tareas fueron realizadas con ayuda de la biblioteca scikit-learn del lenguaje Python, que incluye una variedad de algoritmos para el aprendizaje de máquina [15].

En la tabla 1 se aprecian los resultados obtenidos aplicando la metodología de trabajo en las tres configuraciones descritas. Se puede apreciar que las técnicas empleadas en las configuraciones dos y tres lograron reducir la cantidad de características. La figura 3 muestra el comportamiento de la métrica F-Score atendiendo

a la cantidad de características empleadas para la representación vectorial de los documentos.

Se puede apreciar que de manera general la precisión de los clasificadores aumentó ligeramente cuando se redujo la cantidad de características empleadas para la representación de los documentos. En el caso de la Red neuronal se empleó su variante más simple, el Perceptrón. Este clasificador se ve limitado cuando no existe una evidente separación lineal entre las clases; por lo que al disminuir la cantidad de características disminuye la dispersión en la matriz de representación y por tanto resulta más difícil para el Perceptrón encontrar una frontera lineal entre las clases. Este resultado contrasta con el obtenido por la Máquina de soporte vectorial, por ejemplo, que a pesar de ser también un clasificador lineal puede emplearse de manera eficiente en casos no lineales como el que es objeto de estudio[7].

Tabla 1. Resumen de resultados para las tres configuraciones establecidas

		Configuración 1				Configuración 2				Configuración 3			
Tipo características:	**Total de características**	15808				15699				11758			
Unigramas	**Eliminación de *stopwords***	no				si				si			
	Algoritmo de *stemming*	no				no				si			
	Filtrado *TFIDF*	si				si				si			
Parameters	**Clasificador**	**Prec(π)**	**Mem(ρ)**	**f-score**	**Soporte**	**Prec(π)**	**Mem(ρ)**	**f-score**	**Soporte**	**Prec(π)**	**Mem(ρ)**	**f-score**	**Soporte**
Perceptron	**Perceptrón**												
Perceptron(alpha=0.00001, max_iter=1000, class_weight='balanced')	0	0,66	0,81	0,73	1080	0,66	0,76	0,70	1080	0,64	0,83	0,72	1080
	1	0,72	0,52	0,61	964	0,67	0,55	0,61	964	0,71	0,48	0,57	964
	avg / total	**0,69**	**0,68**	**0,67**	**2044**	**0,66**	**0,66**	**0,66**	**2044**	**0,68**	**0,66**	**0,65**	**2044**
kNN	**Vecinos Próximos**												
KNeighborsClassifier(n_neighbors = 20)	0	0,58	0,98	0,73	1080	0,57	0,99	0,73	1080	0,59	0,99	0,74	1080
	1	0,90	0,20	0,32	964	0,93	0,18	0,30	964	0,94	0,25	0,39	964
	avg / total	**0,73**	**0,61**	**0,54**	**2044**	**0,74**	**0,61**	**0,53**	**2044**	**0,76**	**0,64**	**0,58**	**2044**
Random Forest	**Random Forest**												
RandomForestClassifier(n_estimators=300, criterion='entropy', min_samples_leaf=1, min_samples_split=9)	0	0,61	0,99	0,76	1080	0,62	0,98	0,76	1080	0,63	0,98	0,77	1080
	1	0,97	0,30	0,45	964	0,95	0,33	0,49	964	0,95	0,34	0,50	964
	avg / total	**0,78**	**0,66**	**0,61**	**2044**	**0,78**	**0,68**	**0,64**	**2044**	**0,78**	**0,68**	**0,64**	**2044**
SVM	**SVM**												
SGDClassifier(alpha=.00001, max_iter=1000, penalty='l2')	0	0,63	0,98	0,76	1080	0,63	0,98	0,76	1080	0,63	0,98	0,77	1080
	1	0,94	0,35	0,51	964	0,93	0,35	0,50	964	0,94	0,36	0,52	964
	avg / total	**0,77**	**0,68**	**0,54**	**2044**	**0,77**	**0,68**	**0,64**	**2044**	**0,78**	**0,69**	**0,65**	**2044**
Naïve Bayes(Multinomial)	**Naive Bayes**												
MultinomialNB(alpha=0.00001)	0	0,57	0,97	0,72	1080	0,58	0,97	0,72	1080	0,57	0,98	0,72	1080
	1	0,85	0,18	0,30	964	0,85	0,21	0,34	964	0,87	0,18	0,29	964
	avg / total	**0,70**	**0,60**	**0,52**	**2044**	**0,71**	**0,61**	**0,54**	**2044**	**0,71**	**0,60**	**0,52**	**2044**

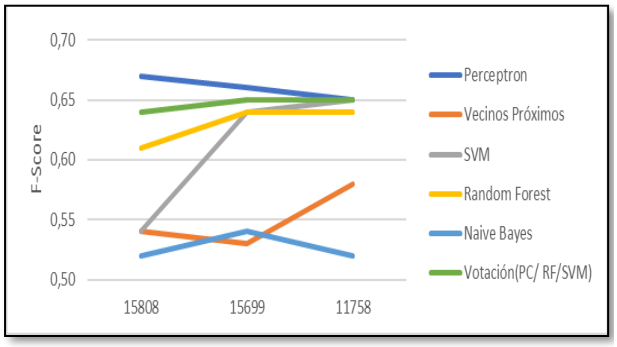

Figura 3. Comportamiento del F-Score vs la cantidad de características de representación

Analizando detalladamente los resultados de la tabla 1 se puede observar que los valores de las métricas obtenidos por los clasificadores en las tres configuraciones son muy similares. Estos resultados se deben en gran medida a la alta dimensionalidad del problema que se traduce en una matriz de representación muy dispersa debido al número limitado de documentos de entrenamiento disponibles (11642). Según [16] es deseable que los conjuntos de datos de entrenamiento contengan más documentos que la cantidad de características empleadas para su representación vectorial. Si bien las técnicas tradicionales empleadas para reducir la dimensionalidad del problema permitieron disminuir el número de características de un total de 15808 (configuración 1) hasta 11758 (configuración 3); este numeró es todavía mayor a lo deseado.

Para dar solución al problema de la alta dimensionalidad se decidió emplear una estrategia de reducción de características basada en el empleo de un enfoque de envoltura. La estrategia empleada fue propuesta en [17] y tiene como objetivo encontrar el subconjunto de las características que maximicen la precisión de la clasificación. Para la selección de este subconjunto se emplea un algoritmo genético que mediante un proceso iterativo evalúa subconjuntos de características empleando un clasificador como función de evaluación. En cada iteración se cambia el espacio de soluciones hasta llegar a la solución óptima.

La función de evaluación o función objetivo definida en [17] tiene como finalidad evaluar cada individuo(subconjunto de características) generado por el algoritmo genético; comparando la precisión y a la vez minimizando el número de características a emplear. La clasificación se realiza empleando el α x 100% de los documentos disponibles para entrenar un SVM y la precisión empleada en la función objetivo se obtiene al realizar la clasificación de los (1- α) x 100% documentos restantes. La siguiente expresión se emplea para evaluar cada individuo:

$$f_e = (1-p) * error + p * nc' \,, \qquad (6)$$

donde p es peso que toma en cuenta tanto la precisión como la cantidad de características de cada individuo y nc' es el número de características normalizado que se obtiene dividiendo el número de característica de cada individuo entre el número total de características.

Para emplear el algoritmo genético propuesto en dicho enfoque es necesario realizar una representación vectorial de cada documento obteniendo una matriz de dimensión *m x n* que representa la colección de documentos; donde *m* representa la cantidad de documentos y *n* representa la cantidad total de características. Esta matriz se denomina matriz de características y el valor de cada elemento M_{ij} es uno (1) si la i-ésima

característica está presente en el j-ésimo documento y cero (0) en caso contrario. En la figura 4 podemos observar un ejemplo de la estructura del vector de características para dos documentos diferentes empleando unigramas y biagramas: d1 y d2.

Unigramas							Biagramas						
0	1	1	1	0	0	0	1	0	0	1	1	0	1

d_1

Unigramas							Biagramas						
0	0	1	0	0	1	1	0	0	0	1	0	0	1

d_2

Figura 4: Vectores de características para los documentos d1 y d2

En la tabla 2 se muestra un resumen de los resultados obtenidos después de aplicar el enfoque de envoltura sobre el corpus de entrenamiento. Se decidió analizar el comportamiento del sistema empleado para tres valores del parámetro de peso de la función objetivo: 0.1, 1.25 y 0.5. La cantidad de iteraciones o generaciones se fijó en 200 ya que se determinó experimentalmente que valores mayores impactan de forma exponencial el costo computacional del algoritmo genético sin garantizar un aumento significativo de los valores de precisión de los clasificadores. Dado el potencial de este método para la reducción del número de características se decidió incluir además de los unigramas, biagramas, generándose un total inicial de 15455 características.

Tabla 2. Resultados después una corrida de 200 generaciones del algoritmo genético para diferentes valores del parámetro p de la función objetivo

,	Población Inicial		Población Final		Peso (*p*)
	Mejor Individuo	**Peor Individuo**	**Mejor Individuo**	**Peor Individuo**	
Características obtenidas	120	160	364	1918	0,1
Precisión de clasificación	0,8220	0,7260	0,8479	0,8461	
Error de clasificación	0,1780	0,2740	0,1521	0,1539	
Función Objetivo	0,1609	0,2476	0,1392	0,1510	
Características obtenidas	120	5078	122	112	0,25
Precisión de clasificación	0,8171	0,8171	0,8504	0,8319	
Error de clasificación	0,1829	0,1829	0,1496	0,1681	
Función Objetivo	0,1391	0,2193	0,1142	0,1279	
Características obtenidas	120	5018	118	111	0,5
Precisión de clasificación	0,8110	0,8042	0,8454	0,8307	
Error de clasificación	0,1890	0,1958	0,1546	0,1693	
Función Objetivo	0,0984	0,2602	0,0811	0,0883	

Tabla 3. Resumen de resultados para los tres valores del parámetro *p*

Pesos (*p*)		**0,1**				**0,25**				**0,5**			
Tipo características: Unigramas + Biagramas	**Total de características**	364				122				118			
Parámetros	**Clasificador**	**Prec (π)**	**Mem (ρ)**	**f-score**	**Soporte**	**Prec (π)**	**Mem (ρ)**	**f-score**	**Soporte**	**Prec (π)**	**Mem (ρ)**	**f-score**	**Soporte**
PC	**Perceptrón**												
Perceptron(alpha=0.00001, max_iter=1000, class_weight='balanced')	0	0,67	0,64	0,65	1080	0,82	0,30	0,44	1080	0,72	0,37	0,49	1080
	1	0,61	0,65	0,63	964	0,54	0,92	0,68	964	0,54	0,84	0,66	964
	avg / total	**0,64**	**0,64**	**0,64**	**2044**	**0,69**	**0,60**	**0,56**	**2044**	**0,64**	**0,59**	**0,57**	**2044**
kNN	**Vecinos Próximo**												
KNeighborsClassifier(n_neighbors=20)	0	0,61	0,98	0,76	1080	0,61	0,99	0,75	1080	0,61	0,98	0,75	1080
	1	0,94	0,31	0,47	964	0,96	0,28	0,44	964	0,94	0,30	0,45	964
	avg / total	**0,77**	**0,66**	**0,62**	**2044**	**0,78**	**0,66**	**0,60**	**2044**	**0,77**	**0,66**	**0,61**	**2044**
RF	**Random Forest**												
RandomForestClassifier(n_estimators=300, criterion='entropy', min_samples_leaf=1, min_samples_split=9)	0	0,65	0,95	0,77	1080	0,66	0,96	0,78	1080	0,65	0,96	0,78	1080
	1	0,89	0,42	0,57	964	0,91	0,44	0,59	964	0,90	0,43	0,59	964
	avg / total	**0,76**	**0,70**	**0,68**	**2044**	**0,77**	**0,71**	**0,69**	**2044**	**0,77**	**0,71**	**0,69**	**2044**
SVM	**SVM**												
SGDClassifier(alpha=.00001, max_iter=1000, penalty='l2')	0	0,64	0,96	0,77	1080	0,64	0,96	0,77	1080	0,65	0,97	0,78	1080
	1	0,91	0,39	0,55	964	0,91	0,40	0,56	964	0,94	0,40	0,56	964
	avg / total	**0,77**	**0,69**	**0,66**	**2044**	**0,77**	**0,70**	**0,67**	**2044**	**0,78**	**0,71**	**0,68**	**2044**
NB	**Naïve Bayes**												
MultinomialNB(alpha=0.00001)	0	0,63	0,95	0,76	1080	0,64	0,95	0,77	1080	0,64	0,96	0,77	1080
	1	0,86	0,38	0,53	964	0,87	0,41	0,56	964	0,89	0,40	0,56	964
	avg / total	**0,74**	**0,68**	**0,65**	**2044**	**0,75**	**0,69**	**0,67**	**2044**	**0,76**	**0,70**	**0,67**	**2044**

Los nuevos subconjuntos de características obtenidas se emplearon para entrenan nuevamente los clasificadores objeto de estudio. La tabla 3 muestra un resumen de los resultados obtenidos al someter los documentos de prueba a un nuevo proceso de clasificación. La figura 5 muestra el comportamiento de la métrica F-Score con la disminución del número de características. Esta gráfica muestra un comportamiento similar al observado en la figura 3 donde aumenta la precisión de la clasificación al disminuir el número de características empleado para la representación vectorial de los documentos.

Los clasificadores que obtuvieron mayor precisión en los tres casos son las Máquinas de soporte vectorial y los Bosques aleatorios. Estos resultados coinciden con los obtenidos en los anteriores experimentos, hecho que demuestra el potencial de estos dos algoritmos de clasificación para esta tarea. En el caso del Perceptrón se comprobó que su precisión disminuyó notablemente con la disminución del número de características de representación. Este comportamiento apoya la idea de que los resultados del primer experimento estuvieron influenciados por la alta dimensionalidad obtenida inicialmente.

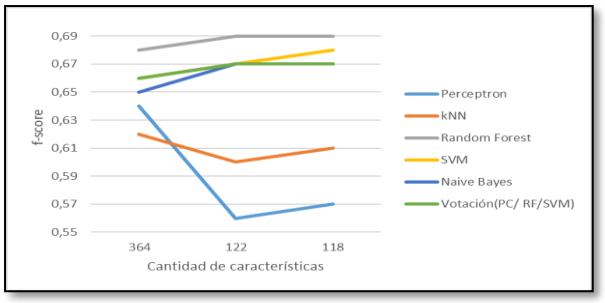

Figura 5. Comportamiento del F-Score vs la cantidad de características de representación

Los resultados muestran el potencial del enfoque de envoltura para reducir la dimensionalidad del problema; partiendo de un valor inicial de 15455 características (unigramas y biagramas presentes en el corpus) se obtiene un conjunto de solo 118 características para una corrida con 200 generaciones del algoritmo genético. Estos resultados contrastan con los obtenidos inicialmente donde se redujo la dimensionalidad desde un valor de 15808 características hasta 11758 empleando solamente técnicas de eliminación de palabras vacías (*stopwords*) y un algoritmo de *stemming* para encontrar la raíz de los vocablos empleados como características. Aunque el enfoque de envoltura empleado demostró ser un método efectivo para tratar con el problema de la alta dimensionalidad, es de señalar como punto de atención su alto costo computacional.

A pesar de los resultados obtenidos en el presente trabajo todavía existen limitaciones que no pueden ser ignoradas. El conjunto de datos empleados para el entrenamiento es un poco limitado en cuanto a su tamaño (sólo 16231 registros), aun cuando se empleó validación cruzada es recomendable contar con un corpus de entrenamientos más grandes. Otro de los

aspectos a tener en cuenta es la naturaleza del texto empleado. En las bitácoras del SIOP encontramos textos cortos formados por una mezcla de palabras en idioma inglés y español, términos propios de jerga empleada en la industria petrolera local, unidades de medidas y abreviaturas que son generadas en un ambiente operativo lo que propicia la introducción de errores ortográficos, tipográficos y gramaticales. La combinación de todas estas características complejiza de manera notable las tareas de procesamiento del texto impactando la precisión en el proceso de clasificación

4. CONCLUSIONES

Este trabajo es la continuación de los esfuerzos que inició el Instituto Mexicano del Petróleo con su Metodología de Análisis de Tiempos de Perforación diseñada para la reducción de los tiempos no productivos (TNP) y también se inserta en la estrategia trasformación digital que está impulsando PEMEX para generar ahorros y mejorar la eficiencia operativa en todas sus líneas de negocio.

Los resultados obtenidos permiten concluir que fue posible realizar exitosamente un proceso de TC para la clasificación autónoma de los eventos registrados en las bitácoras del Sistema de Información Operativa de Perforación (SIOP) de PEMEX Exploración y Producción. Los mejores resultados de clasificación fueron obtenidos empleando el algoritmo de Bosques aleatorios (RF) el cual obtuvo el mayor valor en la métrica de desempeño F-Score que combina el desempeño de otras dos métricas: precisión y memoria. El puntaje del clasificador basado en RF fue de 0.69 para documentos desconocidos, superando levemente el valor de 0.68 alcanzado por las Máquinas de Soporte Vectorial.

Estos resultados fueron obtenidos luego de realizar un proceso de reducción de dimensionalidad que permitió reducir de forma notable el número de características empleadas en la representación de los documentos. Esta tarea fue realizada empleando un enfoque de envoltura que usa un algoritmo genético para obtener un buen conjunto de características utilizando un clasificador como función objetivo.
Resulta notable señalar que, el grupo de las características empleadas en la obtención de los mejores resultados está formada sólo por unigramas. Este hecho reafirma la idea planteada en la bibliografía especializada, donde se afirma que los términos son la mejor unidad o característica tanto para la representación como para la clasificación de textos.

Como línea de trabajo futuro se propone analizar el impacto del empleo de otros modelos de representación de textos; en particular pasar de una representación vectorial de documentos a una representación vectorial de palabras en el espacio continuo. Este modelo de representación conocido como word2vect (del inglés, *words to vectors*) [18] ha ganado mucha relevancia en los últimos años. Entre las ventajas más notables de su empleo podemos citar: disminución de la dimensionalidad al obtener representaciones de las palabras con vectores pequeños, procesos de preprocesamiento más simples ya que palabras

semánticamente similares generan representaciones similares o cercanas; y además es posible obtener relaciones semánticas mediante operaciones vectoriales simples.

REFERENCIAS

[1] Abou-Sayed, A. Data Mining Applications in the Oil and Gas Industry, *Journal of Petroleum Technology*. 2012, 40(10), 88–95.

[2] SHCP, Estimación de Gasto Público para 2016. Fuentes de los recursos públicos, Ciudad de Mexico, 2016.

[3] IEA, “Energy policies beyond IEA countries. Mexico 2017”, 2017.

[4] Langer, J. Costos de perforación de pozos en Latinoamérica, Campetrol, Cámara Colombiana de Bienes y Servicios Petroleros. Bogota, Colombia, 2015.

[5] Pemex, 1er foro tecnológico Drive Oil & Gas. Recuperado el 22 de febrero de 2018, de http://www.pemex.com/saladeprensa/boletines_nacionales/Paginas/2018-014-nacional.aspx, 2018.

[6] IMP, Nueva Metodología de Análisis de Tiempos de Perforación, Instituto Mexicano del Petróleo, Ciudad de Mexico, 2015.

[7] F. Sebastiani, Machine Learning in Automated Text Categorization, *ACM Computing Surveys*. 2002, 34(1), 2002.

[8] Harish, B., Guru, D. y Manjunath, S. Representation and Classification of Text Documents: A Brief Review, *IJCA,Special Issue on RTIPPR*. 2010, 2, 110–119.

[9] Song, F., Liu, S. y Yang, J. A comparative study on text representation schemes in text categorization, *Journal of Pattern Analysis Application*. 2005, 8, 199–209.

[10] Salton, G. y Buckley, C. Term-weighting approaches in automatic text retrieval, *Information processing & management*. 1998, 24(5), 513–523.

[11] Vijayarani, S., Ilamathi, J. y Nithya, M. Preprocessing Techniques for Text Mining - An Overview, *International Journal of Computer Science & Communication Networks*. 2015, 5(1), 7–16.

[12] Hotho, A., Nürnberger, A. y Paaß, G. A brief survey of text mining, *Ldv Forum*. 2005, 20(1).

[13] Mandowara, J. y Jain, A. Text Classification by Combining Text Classifiers to Improve the Efficiency of Classification, *International Journal of Computer Application*. 2016, 6(2), 126–129.

[14] Li, W. Automatic Log Analysis using Machine Learning, Uppsala Universitet, 2013.

[15] Pedregosa, F. *et al.*, Scikit-learn: Machine Learning in Python, *Journal of Machine Learning Researc*h. 2011, 12, 2825–2830.

[16] Casasola, E. y Marín, G. Evaluación de Modelos de Representación del Texto con Vectores de Dimensión Reducida para Análisis de Sentimiento, en *TASS 2016: Workshop on Sentiment Analysis at SEPLN*. 2016, pp. 23–28.

[17] Ortega, R. Diseño de algoritmos bioinspirados para la selección de características en el análisis de sentimientos de documentos en español, 2015.

[18] Mikolov, T. y Le, Q. Distributed representations of sentences and documents, en *International Conference on Machine 2014*. 2014, 32.

Acerca de los autores

William Feal Delgado graduado como Ingeniero Informático en la Universidad de Cienfuegos, Cuba. Actualmente es alumno de la División de Posgrado e Investigación del Instituto Tecnológico de Zacatepec donde cursa la Maestría en Ciencias de la Ingeniería. Sus intereses de investigación se centran en la minería de datos, la ingeniería de software y los sistemas distribuidos de bases de datos.

Manuel Antonio Chi Chim, investigador del Instituto Mexicano del Petróleo. Graduado en la Universidad Autónoma de Yucatán, cuenta con una maestría y un doctorado en Ciencias de la Computación, por parte del Instituto Politécnico Nacional y del Instituto Tecnológico y de Estudios Superiores de Monterrey, respectivamente. Sus líneas de investigación se orientan a las aplicaciones de modelos de inteligencia artificial, minería de datos, ingeniería de software y de optimización combinatoria para la toma de decisiones en explotación de campos maduros, yacimientos naturalmente fracturados, pozos depresionados e ingeniería concurrente.

Claudia Noguerón González, es maestra en Ciencias de la Computación egresada del Centro nacional de Investigación y Desarrollo Tecnológico (CENIDET) obtuvo el grado en diciembre de 1996, obtuvo su Licenciatura en Informática en el Instituto Tecnológico de Zacatepec en septiembre de 1993, ha participado en varios proyectos de investigación, así como en artículos científicos de índole nacional e internacional. Ha sido miembro del Sistema Estatal de Investigadores.

Francisco Javier Cartujano, Doctor en Administración con Especialidad en Sistemas de Información egresado del Tecnológico de Monterrey. Se ha desempeñado en el sector privado como gerente de sistemas y como profesor investigador del Departamento de Computación del Tecnológico de Monterrey, Campus Ciudad de México y Campus Cuernavaca. Actualmente está adscrito al departamento de Sistemas Computacionales del Instituto Tecnológico de Zacatepec. Ha formado parte del Sistema Nacional de Investigadores del CONACYT y ha sido líder de varios proyectos de investigación computacional. Sus áreas de interés son todo lo relacionado a bases de datos y minería de datos.