Técnicas de aprendizaje de máquina utilizadas para la minería de texto

Ángel Freddy Godoy Viera*

Artículo recibido:
30 de octubre de 2014.
Artículo aceptado:
5 de noviembre de 2015.

Resumen

Las técnicas de aprendizaje de máquina continúan siendo muy utilizadas para la minería de texto. Para este artículo se realizó una revisión de literatura en periódicos científicos publicados en los años de 2010 y 2011, con el objetivo de identificar las principales formas de aprendizaje de máquina empleadas para la minería de texto. Se utilizó estadística descriptiva para organizar, resumir y analizar los datos encontrados, y se presentó una descripción resumida de las principales encontradas. En los artículos analizados se hallaron 13 aplicadas para la minería de texto, el 83% de los artículos mencionaban de 1 a 3 técnicas de aprendizaje de máquina, las princi-

* Líder del grupo de investigación de ¨Recuperación de la Información y Tecnología Avanzadas¨ (RITA) de la Universidad Federal de Santa Catarina. a.godoy@ufsc.br

pales usadas por los autores en los artículos estudiados fueron *support vector machine* (SVM), *k-means* (K-M), *k-nearest neighbors* (K-NN), *naive bayes* (NB), *self-organizing maps* (SOM). Los pares que aparecen con mayor frecuencia son SVM/NB, SVM/K-NN, SVM/decission tree.

Palabras clave: Aprendizaje de máquina; Minería de texto; Técnicas de aprendizaje de máquina.

Abstract

Machine Learning Techniques Used for Text Mining
Angel Freddy Godoy Viera

The machine learning techniques are still extensively used for text mining. In this paper, there was a literature review of scientific journals published in the years 2010 and 2011, with the aim of identifying the main machine learning techniques utilized for text mining. It was possible to use descriptive and statistical techniques to organize, summarize and analyze the found data, and also, there is a shortened description of the main techniques found. Thirteen learning techniques applied to text mining appeared in the articles analyzed and 83% of the papers mentioned from 1 to 3 machine learning techniques. The main techniques used by the authors in the papers studied were support vector machine (SVM), k-means (K-M), k-nearest neighbors (K-NN), naive bayes (NB), self-organizing maps (SOM). Pairs of techniques that appear most frequently are SVM/NB, SVM/K-NN, SVM/Decision Tree and NB/K-NN.

Keywords: Machine Learning; Text Mining; Machine Learning Techniques.

Introducción

El área de inteligencia artificial con el transcurso de los años desarrolló diversas técnicas de aprendizaje de máquina para múltiples aplicaciones en el mundo real. En la actualidad, con el gran volumen de documentos producidos en la web por individuos, organizaciones no gubernamentales e instituciones públicas y privadas, existe una gran necesidad de disponer de instrumentos tecnológicos que ayuden a procesar, extraer y resumir las informaciones útiles. Uno de los métodos que surgió para intentar resolver ese problema es la minería de texto que, entre otras técnicas, utiliza el aprendizaje de máquina para la minería de grandes colecciones documentales en formato digital con el fin de extraer informaciones relevantes para los destinatarios de la información. La popularidad del uso de la minería de texto se debe a la capacidad de trabajar con documentos no estructurados para procesarlos y obtener informaciones significativas. En las secciones siguientes de este artículo, presentamos los principales conceptos sobre aprendizaje de máquina y minería de texto, el método utilizado en esta investigación, los resultados hallados y una descripción de las principales técnicas encontradas.

Aprendizaje de máquina

Tom M. Mitchell afirma que el aprendizaje de máquina es un área que estudia cómo construir programas de computadoras que mejoren su desempeño en alguna tarea gracias a la experiencia.[1] Ésta se basa en las ideas de diversas disciplinas, como inteligencia artificial, estadística y probabilidad, teoría de la información, psicología y neurobiología, teoría de control y complejidad computacional.[2] El mismo autor afirma que, para utilizar el abordaje de aprendizaje, se deben considerar una serie de decisiones que incluyen la selección del tipo de entrenamiento, la función objetiva a ser aprendida, su representación y el algoritmo para aprender esa función a partir de ejemplos de entrenamiento.[3]

De acuerdo con Mitchell, los algoritmos de aprendizaje han demostrado ser útiles en diversos dominios de aplicación, como en la minería de datos en grandes bases de datos que contienen regularidades implícitas, las cuales pueden ser descubiertas en forma automatizada, en dominios que son poco

1 T. M. Mitchell, *Machine Learning* (Nueva York: McGraw Hill, 1997), 15.
2 N. J. Nilsson, "Introduction to Machine Learning" (1998), 3-4, http://robotics.stanford.edu/people/nilsson/MLBOOK.pdf, consultado el 12 de noviembre de 2013; Mitchell, *Machine Learning*, 15.
3 Nilsson, "Introduction to Machine Learning", 5-11.

entendidos y donde los humanos no poseen el conocimiento necesario para desarrollar algoritmos efectivos, y en dominios donde los programas se deben adaptar en forma dinámica para responder a condiciones cambiantes en el ambiente.[4]

Por su parte, Fang Lu y Qingyuan Bai mencionan que las técnicas de aprendizaje son el abordaje predominante para la categorización de textos. Estos autores las definen las técnicas como un proceso inductivo general que automáticamente construye un clasificador, aprendiendo a partir de un conjunto de documentos preclasificados.[5]

Las técnicas de aprendizaje se clasifican en supervisadas, no supervisadas y semisupervisadas. En las supervisadas, la meta es aprender el mapeo de las respuestas correctas para los datos de entrada que le son proporcionados; para esto, se utiliza un conjunto de datos de entrenamiento constituidos por pares que consisten en patrones de entrada y salida correcta. De esta forma, el sistema aprende el mapeo de la salida correcta para cada patrón de entrada que se le presenta.[6]

Yu Wanjun y Song Xiaoguang afirman que las etapas del aprendizaje supervisado aplicadas en la categorización de texto son inicialmente un conjunto de ejemplos ya clasificados que son presentados al algoritmo con lo que se construye un clasificador. Posteriormente, se presentan ejemplos no clasificados al clasificador para que los ordene. Finalmente, se debe tomar medidas para evaluar el desempeño del clasificador.[7]

Algunos ejemplos de técnicas de aprendizaje de máquina supervisado son árboles de decisión, máxima entropía, *naive bayes*, *support vector machines*, etcétera.[8]

En las técnicas de aprendizaje no supervisado, no se requiere la intervención de humanos para elaborar un conjunto de datos previamente categorizados para ser presentado al algoritmo de aprendizaje. La meta del aprendizaje no supervisado es encontrar patrones interesantes considerando la distribución y composición de los datos que le son presentados.[9] Ejemplos de las técnicas de aprendizaje no supervisado son las técnicas de *clustering*.

4 *Ibid.,* 17.
5 F. Lu y Q. Bai, "A Refined Weighted K-Nearest Neighbors Algorithm for Text Categorization", *IEEE* (2010): 326.
6 O. Chapelle, B. Schölkopf y A. Zien, *Semi-Supervised Learning* (Cambridge: MIT Press, 2006), 3.
7 Y. Wanjun y S. Xiaoguang, "Research on Text Categorization Based on Machine Learning," Advanced Management Science (ICAMS), 2010 *IEEE* International Conference 2, *IEEE* (2010): 253.
8 Lu y Bai, "A Refined Weighted K-Nearest Neighbors Algorithm…", 326; D. Torunoglu, E. Çakırman, M. Can Ganiz, S. Akyoku y M. Zahid Gürbüz, "Analysis of Preprocessing Methods on Classification of Turkish Texts," *IEEE* (2011): 114.
9 Chapelle, Schölkopf y Zien, *Semi-Supervised Learning*, 349.

Las técnicas de aprendizaje semisupervisado son un término medio entre el aprendizaje supervisado y el no supervisado. En este tipo de aprendizaje, los datos se dividen en dos partes: un grupo de datos clasificados y otro de no clasificados. A esto se denomina aprendizaje semisupervisado estándar.

Para Olivier Chapelle, Bernhard Schölkopf y Alexander Zien, el aprendizaje semisupervisado es más útil cuando existe una mayor cantidad de datos no clasificados que de datos clasificados. Especialmente, cuando para obtener los clasificados se requiere mucho esfuerzo, demora mucho o es muy costoso; ya que, además, la obtención no clasificados generalmente es menos costosa.[10] Ejemplos de técnicas de aprendizaje semisupervisado son las técnicas de *transductive support vector machines*, *expectation maximization* (EM), etc. Algunas aplicaciones del aprendizaje semisupervisado, mencionadas por los autores son el reconocimiento del habla, clasificación de páginas web y en la secuencia de proteínas.

Hsinchun Chen y Michael Chau mencionan los principales paradigmas del aprendizaje de máquina, que son el modelo probabilístico, el aprendizaje simbólico y la inducción de reglas, las redes neuronales, los algoritmos basados en evolución, el aprendizaje analítico y los métodos híbridos.

El modelo probabilístico es uno de los métodos más antiguos de aprendizaje que es utilizado frecuentemente para clasificar diferentes objetos en clases definidas previamente con base en un conjunto de características. Ejemplos de éste son el modelo Bayesiano y el modelo *naive Bayes*.[11]

El *aprendizaje simbólico y la inducción de reglas* puede clasificarse de acuerdo con la estrategia de aprendizaje subyacente en aprendizaje de rutina (*rote*), por instrucciones, por analogía, a partir de ejemplos y por descubrimiento.[12] Ejemplo de técnicas de este tipo son el algoritmo de árbol de decisión ID3 y su variación el algoritmo C4.5. Éstos presentan el resultado de la clasificación en forma de árboles de decisión o un conjunto de reglas de producción.

Las redes neuronales artificiales imitan a las neuronas humanas. Aquí los conocimientos son representados por descripciones simbólicas; el conocimiento es aprendido y recordado por redes de neuronas artificiales interconectadas por sinapsis con pesos y unidades de umbral lógicas (Lippmann; Rumelhart, Hinton y McClelland).[13] Existen diferentes modelos de redes neuronales,

10 Chapelle, Schölkopf y Zien, *Semi-Supervised Learning,* 11.
11 H. Chen y M. Chau, "Web Mining: Machine Learning for Web Applications", *Annual Review of Information Science and Technology*, editado por Blaise Cronin (2004): 293.
12 *Ibid.,* 293.
13 *Ibid.,* 294.

algunos ejemplos son *feedforward/backpropagation*, los mapas autoorganizados de Kohonen (SOM) y el modelo de red neuronal de Hopfield.[14]

Los *algoritmos evolutivos* imitan el proceso de evolución en la naturaleza. Fogel identifica tres categorías de éstos: los genéticos, las estrategias evolutivas y la programación evolutiva.[15] Los algoritmos genéticos imitan los principios de los genes y utilizan operadores de mutación y *crossover* en la población para seleccionar los individuos más adaptados y repiten esa operación en varias generaciones hasta obtener el mejor individuo. En las estrategias evolutivas, se hace evolucionar una población de números reales que codifican las posibles soluciones de un problema numérico y los tamaños de salto; en las estrategias evolutivas la selección es implícita. En la programación evolutiva, se hace evolucionar a una población de máquinas de estados finitos, sometiéndolas a transformaciones unitarias.[16]

El *aprendizaje analítico* representa los conocimientos como reglas lógicas y realiza un raciocinio sobre ellas para buscar por pruebas, las cuales son compiladas en reglas más complejas para resolver problemas con un número pequeño de búsquedas.[17]

Se debe destacar que, en la práctica, estos paradigmas de aprendizaje de máquina muchas veces se utilizan combinando varias técnicas para aprovechar mejor las ventajas que cada una presenta y también subsanar las fragilidades que tienen si fuesen utilizados en forma individual. Eso es conocido como métodos híbridos. En la siguiente sección presentamos algunas consideraciones sobre minería de texto.

Minería de texto

La minería de texto busca encontrar información relevante para un propósito, en particular lo hace a partir de textos en lenguaje natural. Ian H. Witten, Frank Eibe y A. Mark la definen como aquélla que busca hallar patrones en el texto. Para alcanzar dicho objetivo, analiza textos para extraer informaciones que sean útiles para un propósito particular.[18]

Por su parte, Ronen Feldman, y James Sanger la describen como un proceso intensivo de conocimiento en el que un usuario interactúa con una colección de documentos en el tiempo, utilizando un conjunto de herramientas

14 *Ibid.,* 294-295.
15 *Ibid.,* 295.
16 R. Gómez Pino Díez, A. Gómez y N. de Abajo Martínez, *Introducción a la inteligencia artificial: sistemas expertos, redes neuronales artificiales y computación evolutiva* (Oviedo: Servicio de Publicaciones Universidad de Oviedo, 2011), 88.
17 Chen y Chau, "Web Mining: Machine Learning for Web Applications", 295.
18 I. H. Witten, F. Eibe, Mark A., *Data Mining: Practical Machine Learning Tools and Techniques* (San Francisco: Morgan Kaufmann, 2011), 386.

de análisis; además, busca extraer información útil de una fuente de datos a través de la identificación y exploración de patrones interesantes. En ella las fuentes de datos son las colecciones de documentos y los patrones interesantes se encuentran en textos generalmente no estructurados.[19]

La minería de texto proviene en gran parte de las investigaciones en minería de datos y, por lo tanto, tienen similitudes en su arquitectura de alto nivel; por ejemplo, ambos sistemas se basan en rutinas de preprocesamiento, algoritmos para descubrir patrones y la capa de elementos de presentación que contienen herramientas de visualización para mejorar la navegación en los conjuntos de respuestas.

Considerando que la minería de texto se centra en análisis de textos en lenguaje natural, se fundamenta en otras disciplinas de las ciencias de la computación que trabajan con el manejo de lenguaje natural. Asimismo, utiliza técnicas y metodologías de las áreas de recuperación de la información, extracción de la información y lingüística computacional, principalmente.[20]

Podemos diferenciar la minería de datos de la de texto con lo que señalan los autores Sholom M. Weiss y colaboradores, quienes afirman que en la primera se organizan los datos en forma de planillas, mientras que, en la segunda, se ven los formatos de documentos, y para el aprendizaje parten de un patrón usado en el mundo de los documentos, que es una variación del formato xml.[21]

Feldman y Sanger también señalan las diferencias entre minería de datos y minería de texto. Para ellos, en la primera, los datos se guardan en formatos estructurados, y gran parte de su preprocesamiento se centra en la depuración y normalización de los datos, así como en crear un gran número de uniones de tablas. En contraste, en la minería de texto, el preprocesamiento se enfoca en reconocer y extraer características representativas para documentos en lenguaje natural. Tales características pueden ser palabras clave relevantes, identificación de nombres de personas, organizaciones, etcétera. El objetivo del preprocesamiento es transformar datos no estructurados que se encuentran en la colección de documentos en un formato intermediario estructurado más explícito.[22]

Witten *et al.* explican la diferencia entre la minería de datos y la minería de texto, indicando que, en la primera, se procura extraer información implícita, previamente desconocida y potencialmente útil a partir de un gran

19 R. Feldman y J. Sanger, *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data* (Cambridge: Cambridge University Press, 2007), 1.
20 *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data,* 1.
21 S. M. Weiss, N. Indurkhya, T. Zhang y F. Damerau, *Text Mining: Predictive Methods for Analyzing Unstructured Information* (Nueva York: Springer, 2004), 3.
22 Feldman y Sanger, *The Text Mining Handbook*, 1.

volumen de datos. En la segunda, la información a ser extraída se encuentra escrita en el texto en forma clara y explícita. Sin embargo, el problema principal para el usuario es la dificultad de tener que acceder y leer grandes volúmenes de documentos textuales en formato digital que están disponibles en la actualidad para fines informativos, estratégicos o de recreación.[23]

Por lo tanto, la minería de texto estudia diversas formas de representar los documentos textuales para que puedan ser utilizados por sistemas informáticos y por personas que no tienen tiempo para leer los grandes volúmenes disponibles en el medio digital.[24]

Desde el punto de vista de los sistemas de procesamiento automatizado de texto, el problema consiste en que los documentos textuales, generalmente no son estructurados, al contrario, son amorfos y difíciles de ser procesados, además, los documentos en su mayoría no están representados de una forma en la que puedan ser aprovechados directamente por los sistemas de minería de texto.[25]

Michael W. Berry y Jacob Kogan, a su vez, afirman que los mayores temas estudiados en la minería de texto son la extracción de palabras clave, clasificación, agrupamiento, extracción de nombres y entidades, detección de anomalías y tendencias y flujos de texto.[26] Cada uno de esos temas forma parte de un subárea de la minería de texto.

En la subárea de resumen de texto, la salida del sistema de minería de texto es el resumen de características que se destacan (trechos relevantes dentro de los documentos) en un gran acervo textual. Otra subárea de la minería de texto es la clasificación, en la que cada instancia representa un documento y las clases son los asuntos tratados. Así, los documentos se clasifican según las palabras que aparecen dentro de los mismos utilizando diferentes técnicas de la minería de texto.[27]

El agrupamiento de documentos es otra subárea más de la minería de texto, donde son reunidos de acuerdo con criterios de similitud entre las palabras que se encuentran. La principal característica de esta técnica es que no existen categorías predefinidas, sin embargo, esto permite que sea definido el número de grupos que se desea crear para el conjunto de documentos que se procesarán.

Weiss *et al.* citan algunas áreas en las que la minería de texto es aplicada, tales como la clasificación de documentos, recuperación de la información,

23 *Ibid.,* 386.
24 *Ibid.,* 386.
25 *Ibid.,* 386.
26 M. W. Berry y J. Kogan, *Text Mining: Applications and Theory* (Chicester: Wiley, 2010), 13.
27 Witten, Eibe y Mark, Data Mining: *Practical Machine Learning Tools and Techniques*, 387.

agrupamiento y organización de documentos, extracción de información.[28] También se aplica para análisis de sentimientos, análisis de Internet, textos en los medios sociales, etcétera.

Otras aplicaciones de minería de texto son la identificación automatizada de idiomas de los documentos en colecciones internacionales, atribución de autoría de documentos a partir del análisis de los mismos, extracción de metadatos y extracción de entidades.

En la próxima sección de este artículo se exponen la metodología utilizada en la investigación, los resultados y una breve descripción de algunas de las principales técnicas identificadas.

Método

Se realizó una investigación bibliográfica sobre minería de texto en las bases de datos Science Direct,[29] Scopus,[30] IEEEXplore Digital Library[31] y Web of Knowledge.[32] Para esto, fueron seleccionados artículos de texto completo en idioma inglés, publicados entre los años 2010 y 2011, disponibles por suscripción en el portal de periódicos de la Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Capes).[33]

Para recuperar los artículos se utilizó la estrategia de búsqueda "text mining" en los títulos de los documentos y "OR (*text mining OR machine learning*) and *information retrieval*" en el resumen de los artículos. Se usó el término "*information retrieval*" para limitar los resultados de búsqueda y contextualizar las aplicaciones de minería de texto que utilicen técnicas de aprendizaje de máquina, con la recuperación de la información.

Inicialmente fueron recuperados 94 documentos de textos completos después de pasar por un proceso de verificación, en el que se eliminaron los documentos duplicados que no utilizaban técnicas de aprendizaje de máquina en la minería de texto. Concluida esa verificación le restaron 56 documentos. Los resultados del análisis de ésos son presentados en la siguiente sección.

Resultados

En los 56 documentos analizados se identificó a 162 autores, de los cuales 6 publicaron dos artículos en los años de 2010 y 2011, respectivamente.

28 *Ibid.,* 5-10.
29 *Science Direct*, http://www.sciencedirect.com.
30 *Scopus*, http://www.scopus.com.
31 *IEEEXplore Digital Library*, http://ieeexplore.ieee.org.
32 *Web of Knowledge*, http://webofknowledge.com.
33 Portal de Periódicos CAPES/MEC, http://www.periodicos.capes.gov.br.

Los 156 autores restantes publicaron una única vez en los dos años estudiados. La media de autores por artículo es de 3.0 con un mínimo de 1 autor y máximo de 9 autores.

Fueron identificadas 13 técnicas de aprendizaje de máquina aplicadas en la minería de texto que aparecían con mayor frecuencia en los documentos estudiados. En medida, fueron identificadas 2.1 técnicas de aprendizaje de máquina por documento con un mínimo de 1 técnica por documento y un máximo de 6 técnicas por documento.

El 41.07% de los documentos presentaba una sola técnica de aprendizaje de máquina, el 30.36% mencionaba dos, el 12.50% 3, el 10.71% mencionaba 4, el 3.57% 5 y el 1.79% señalaba 6 técnicas. La *Tabla 1* muestra el número de técnicas de aprendizaje de máquina identificadas por documento, indicando su frecuencia y porcentaje.

Tabla 1. Número de técnicas identificadas por documento expresadas en frecuencia y porcentaje

No. De técnicas mencionadas por documento	Cantidad de documentos (Frecuencia)	Porcentaje
1	23	41.07%
2	17	30.36%
3	7	12.50%
4	6	10.71%
5	2	3.57%
6	1	1.79%
TOTAL	56	100.00%

Fuente: Elaboración propia.

La *Tabla 2* muestra las técnicas de aprendizaje de máquina identificadas en los artículos estudiados, agrupando esas técnicas según tres criterios: el primero si la técnica fue utilizada directamente en alguna etapa del proceso de minería de texto, el segundo criterio si la técnica fue utilizada con el objetivo de comparar el desempeño de otra técnica principal y el tercer criterio si la técnica fue mencionada en la revisión de literatura de los documentos analizados. Los datos están puestos por orden decreciente de importancia considerando si la técnica fue utilizada directamente en alguna etapa del proceso.

Es importante resaltar que el conteo de cada técnica se hizo de forma individual para cada documento, por lo tanto, los que poseían más de una técnica fueron contabilizados varias veces.

Tabla 2. Técnicas de aprendizaje de máquina aplicadas en la minería de texto identificadas en las publicaciones estudiadas, separadas por técnica principal, técnica comparativa y mención en la revisión de literatura

TÉCNICA	Frecuencia de utilización como técnica principal	Frecuencia de utilización de la técnica para comparación de desempeño	Frecuencia de mención de la técnica en la revisión de literatura	TOTAL
Support vector machine (SVM)	15 (22.72%)	7 (20.59%)	13 (16.88%)	35
K-means (K-M)	9 (13.63%)	5 (14.71%)	8 (10.39%)	22
K-nearest neighbors (K-NN)	8 (12.12%)	5 (14.71%)	11 (14.29%)	24
Naive Bayes (NB)	8 (12.12%)	5 (14.71%)	10 (12.99%)	23
Self-organizing maps (SOM)	6 (9.09%)	1 (2.94%)	3 (3.90%)	10
Latent semantic indexing (LSI)	4 (6.06%)	1 (2.94%)	7 (9.09%)	12
Hierarchical agglomerative clustering (HAC)	3 (4.55%)	3 (8.82%)	6 (7.80%)	12
Decision trees (DT)	3 (4.55%)	3 (8.82%)	3 (3.90%)	9
Artificial neural network (ANN)	3 (4.55%)	2 (5.88%)	2 (2.60%)	7
Association rules (AR)	3 (4.55%)	0	5 (6.49%)	8
Case-based reasoning (CBR)	2 (3.03%)	0	4 (5.19%)	6
Maximum entropy classifier (MEC)	2 (3.03%)	0	4 (5.19%)	6
Multinomial naive Bayes (MNB)	0 (0.00%)	2 (5.88%)	1 (1.29%)	3
TOTAL	66 (100.00%)	34 (100.00%)	77 (100.00%)	177

Fuente: Elaboración propia.

La *Tabla 2* muestra que la técnica más utilizada es el *support vector machine* (SVM) o máquina de vectores de soporte con el 22.72% de incidencia, seguida de *k-means* (algoritmo k-media) con el 13.63%, *k-nearest neighbors* (el clasificador k vecinos más próximos) y *naive Bayes* con el 12.12% cada una; *self-organizing maps* (mapas autoorganizados) con el 9.09% de incidencia y *latent semantic indexing* (indexación semántica latente) con el 6.06%. Todas estas suman el 75.74% de las técnicas utilizadas como principales en los documentos estudiados.

Hierarchical agglomerative clustering (agrupamiento jerárquico aglomerativo)*, decision tres* (árboles de decisión)*, artificial neural network* (redes neuronales artificiales) y *association rules* (reglas de asociación) presentaron el 4.55% de incidencia cada una; *case-based reasoning* (razonamiento basado en casos) y *maximum entropy classifier* (clasificador de entropía máxima) tuvieron el 3.03% cada una, lo que da un total del 24.26% de las técnicas principales utilizadas en algunas de las etapas de minería de texto.

Entre las técnicas utilizadas para comparar el desempeño en relación con la técnica principal, *support vector machine* apareció en el 20.59% de los documentos, *k-means, k-nearest neighbors* y *naive Bayes* el 14.71% cada una; *hierarchical agglomerative clustering* y *decision trees el* 8.82% cada una. Esas técnicas dan un total del 82.36%.

El 17.64% restante está constituido por *artificial neural network* y *multinomial naive Bayes* con el 5.88% cada una; *self-organizing maps* y *latent semantic indexing* con el 2.94% cada una.

Considerando el criterio "mención de la técnica" en la revisión de literatura de los documentos estudiados, las principales técnicas identificadas presentadas en orden decreciente de incidencia fueron *support vector machine* con el 16.88%, *k-nearest neighbors* con el 14.29%, *naive Bayes* con 12.99%, *k-means* conel 10.39% y *latent Semantic Indexing* con el 9.09%, lo que da un total del 63.64% de las técnicas identificadas. La *Tabla 3* muestra la combinación en pares de las formas de aprendizaje de máquina encontradas. Los números en las celdas detallan la cantidad de documentos en los que las combinaciones de técnicas fueron mencionadas.

Tabla 3. Frecuencia de la combinación en pares de las técnicas de aprendizaje de máquina en los documentos

	SVM	NB	KNN	K-M	AR	DT	LSI	MEC	ANN	MNB	CBR	HAC	SOM
SVM	0	12	8	2	2	6	2	5	4	3	2	0	0
NB	12	0	6	1	1	3	1	5	2	3	1	0	0
KNN	8	6	0	3	2	1	1	1	1	2	1	1	0
K-M	2	1	3	0	4	0	1	0	1	0	1	2	1
AR	2	1	2	4	0	1	2	1	0	0	1	0	0
DT	6	3	1	0	1	0	1	2	2	1	0	0	0
ISI	2	1	1	1	2	1	0	1	0	0	1	0	1
MEC	5	5	1	0	1	2	1	0	1	1	1	0	0
ANN	4	2	1	1	0	2	0	1	0	0	0	0	0

MNB	3	3	2	0	0	1	0	1	0	0	0	0	0
CBR	2	1	1	1	1	0	1	1	0	0	0	0	0
HAC	0	0	1	2	0	0	0	0	0	0	0	0	0
SOM	0	0	0	1	0	0	1	0	0	0	0	0	0

Fuente: Elaboración propia.

La *Tabla 3* muestra que los pares de técnicas que aparecen con mayor frecuencia en los documentos analizados son *support vector machine / naive Bayes* con 12; *support vector machine / k-nearest neighbors* con 8; *support vector machine / decision trees* con 6, *k-nearest neighbors / naive Bayes* con 6; *support vector machine / maximum entropy classifier* con 5, y *naive Bayes / maximum entropy classifier* con 5.

Otras técnicas encontradas en las publicaciones que tienen menor incidencia son *decision rule, logistic regression, latent dirichlet allocation*, *fuzzy association rule, fuzzy clustering, hierarchical fuzzy clustering, rought set* y *clustering by committee* (CBC).

En la siguiente sección serán descritas en forma sucinta las principales técnicas encontradas en esta investigación.

Descripción de las técnicas identificadas

A continuación se hace una revisión de las principales técnicas de aprendizaje de máquina para minería de texto que fueron identificadas en la investigación.

Support vector machine (SVM), máquina de vectores de soporte

El SVM es un método que alcanza altos grados de clasificación correcta en diversos tipos de aplicaciones. Es un algoritmo de aprendizaje supervisado, que funciona como un clasificador lineal que separa los datos en dos clases. Básicamente, el algoritmo encuentra entre los diversos hiperplanos que separan esas dos clases el hiperplano ideal que maximiza el margen entre los vectores de soporte. Los vectores de soporte son un subconjunto de los datos, que son los puntos más próximos de las clases, que define la posición del hiperplano ideal. Los otros puntos de datos no son considerados al determinar ese hiperplano. Cuando dos clases no son linealmente separables, el método de SVM utiliza la función de kernel para proyectar los puntos de los datos en un espacio dimensional superior, de modo que los puntos de los datos se tornen linealmente separables.[34]

34 Wanjun y Xiaoguang, "Research on Text Categorization Based on Machine Learning", 114.

Diversas técnicas de kernel pueden ser utilizadas con el algoritmo SVM. Algunas de las funciones de kernel más comunes son el polinomio, la base radial y la función sigmoidal. Algunos autores afirman que, según su experiencia, observaron que el kernel lineal posee mejor desempeño que las otras.[35]

Weng Zhang, Taketoshi Yoshida y Xijin Tang explican que para realizar la categorización de múltiples clases en el método svm, puede acudirse al método uno-contra-resto, el árbol de decisión, el SVM, etcétera.[36]

Ying Liu y Han Tong Loh señalan que el SVM es la técnica que lidera en desempeño para aplicaciones de clasificación de textos por su excelente capacidad para trabajar con datos, con muchas dimensiones como entrada y conseguir producir salidas (clasificaciones) consistentes.[37]

Mukherjee indica que dos de las principales ventajas de utilizar svm para la clasificación supervisada es que

a) el desempeño de SVM es bueno cuando se tiene un gran número de características; y
b) el SVM ayuda mucho cuando se tiene pocos ejemplos de entrenamiento en tareas de clasificación de múltiples clases.[38]

Naive Bayes (NB)

El NB se basa en el modelo de probabilidad posterior. El término ingenuo (*naive*) se refiere a la suposición de la condición de independencia de los atributos. La meta de la clasificación NB es encontrar la clase óptima para un determinado documento, calculando la clase que da la probabilidad posterior máxima.[39] NB asume que la probabilidad de cada palabra que existe en un documento es independiente de los otros términos que existen en el documento.[40]

35 Wanjun y Xiaoguang, "Research on Text Categorization Based on Machine Learning", 114; W. Zhang, T. Yoshida y X. Tang, "A Comparative Study of TF*IDF, LSI and Multi-words for Text Classification", *Expert Systems with Applications* 38 (2011): 2762.

36 Zhang, Yoshida y Tang, "A Comparative Study of TF*IDF, LSI and Multi-words for Text Classifiation", 2760.

37 Y. Liu y H. Tong Loh, "Domain Concept Handling in Automated Text Categorization", 5th IEEE Conference on Industrial Electronics and Applications, *IEEE*, 2010, 1545.

38 Citado por Bader Aljaber, David Martinez, Nicola Stokes y James Bailey, "Improving MeSH Classification of Biomedical Articles Using Citation Contexts", *Journal of Biomedical Informatics – Elsevier* 44 (2011): 892.

39 S. Pitigala, C. Li y S. Seo, "A Comparative Study of Text Classification Approaches for Personalized Retrieval in PubMed", 2011 IEEE International Conference on Bioinformatics and Biomedicine Workshops, IEEE (2011): 919.

40 K. Sang-Bum, H. Kyoung-Soo, R. Hae-Chang y Sung Hyion Myaeng, "Some Effective techniques for Naïve Bayes Text Classification", *IEEE Transactions on Knowledge and Data Engineering* 18, no. 11 (2006): 1458.

El NB es ampliamente utilizado para la clasificación de texto, debido a su simplicidad, aprendizaje rápido, desempeño razonable con grande cantidad de datos y su capacidad de aprender en forma incremental.[41]

Naive Bayes multinomial (NBM)

El NBM combina el uso de la ley de probabilidad de Bayes y la ley multinomial. El clasificador *naive Bayes* usa la regla de Bayes como su ecuación principal, asumiendo la condición de independencia de los atributos; cada característica individual se presume que sea una indicación de la clase a la cual está asignada, independiente una de otra.[42] El modelo NBM asume que las palabras en un documento son delineadas por una distribución multinomial subyacente que es independiente una de otra. El documento es representado por el número de ocurrencias de los términos en el documento.[43]

Maximum entropy (ME)

El método de entropía máxima (ME) intenta preservar la mayor cantidad de incertidumbre posible. Se calculan una cantidad de modelos y cada característica corresponde a una restricción del modelo. Éste se selecciona para la clasificación sobre todos los modelos que satisfacen esa restricción. De esta forma, todas las presunciones se justifican con las evidencias empíricas disponibles.[44]

Weiss *et al.* afirman que el me históricamente ha sido usado en la literatura de procesamiento de lenguaje natural y es importante en la literatura de extracción de la información. Está relacionado con el RRM (*Robust Risk Minimization*) y el *naive Bayes*, pero tiene un costo computacional mayor.[45]

La ventaja principal del modelo de máxima entropía sobre el *naive Bayes* es que no se necesita presuponer la independencia para cada componente de las características. Esto indica que en el modelo me tenemos que enfocarnos únicamente en seleccionar características que son más útiles para el problema, sin preocuparnos si existen características que son redundantes.[46]

41 K.A. Vidhya y G. Aghila, "Hybrid Text Mining Model for Document Classification," *IEEE* 1 (2010): 210.
42 D. Bing, S. Peiji y D. Zhao, "E-Commerce Reviews Management System Based on Online Customer Reviews Mining", 2010 International Conference on Innovative Computing and Communication and 2010 Asia-Pacific Conference on Information Technology and Ocean Engineering, IEEE *Computer Society* (2010): 376.
43 Torunoglu *et al.*, "Analysis of Preprocessing Methods…", 114
44 Bing, Peiji y Zhao, "E-Commerce Reviews Management…", 376.
45 Weiss *et al.*, *Text Mining*, 118.
46 *Ibid.,* 119-120.

K-nearest neighbor (K-NN)

El clasificador *K*, vecinos más próximos (K-NN), es un método de aprendizaje perezoso basado en ejemplos. En la clasificación basada en ejemplos cuando una nueva instancia necesita ser clasificada, se le compara con los ejemplos existentes usando una métrica de distancia y los ejemplos más próximos son utilizados para asignar la clase a la nueva instancia.[47]

El algoritmo de K-NN para la clasificación de texto se basa en el modelo de espacio vectorial, el cual representa los documentos en forma de vector con la frecuencia de los términos existentes en el documento. Una vez generado tal vector de características para un determinado texto desconocido, el algoritmo K-NN procura por todos los ejemplos de entrenamiento comparar la similitud entre sus vectores de características, para encontrar los *k* ejemplos de entrenamiento más próximos y el documento desconocido es designado a los *k* vecinos más próximos con mayor valor de clasificación.[48]

La principal ventaja del algoritmo K-NN es su facilidad de implementación, pero su costo computacional es alto cuando el tamaño de los documentos usados en el entrenamiento crece. Torunoglu *et al.* afirman que para especificar cuál es el mejor valor de *k* se depende del conjunto de datos y del dominio de la aplicación.[49]

Decision Trees (DT)

El árbol de decisión (DT) es una estructura en árbol, donde cada nodo representa un atributo a ser probado; las ramas representan la salida de la prueba y los nodos finales (hojas) representan la clasificación.

El algoritmo de DT posee dos fases principales: en la primera llamada fase de crecimiento del árbol, el algoritmo inicia con todo el conjunto de datos como nodos raíz. Los datos son divididos en subconjuntos utilizando algún criterio de división. En la segunda fase, etapa de poda del árbol, el árbol total formado se poda para prevenir el exceso de ajuste (*over-fitting*) del árbol a los datos de entrenamiento.[50]

Existen diversos algoritmos para construir árboles de decisión entre ellos ID3, C4.5, SPRINT, SLIQ y PUBLIC. El algoritmo ID3 utiliza esencialmente el criterio de ganancia de información como medida de pureza para cada

47 C. D. Manning, P. Raghavan, y H. Schutze, *Introduction to Information Retrieval* (Cambridge: Cambridge University Press, 2008), 297; Torunoglu *et al.*, "Analysis of Preprocessing Methods…", 114.
48 Wanjun y Xiaoguang, "Research on Text Categorization Based on Machine Learning", 254.
49 Torunoglu *et al.,* "Analysis of Preprocessing Methods…",114.
50 Wanjun y Xiaoguang, "Research on Text Categorization Based on Machine Learning", 52-53.

nodo. El algoritmo C4.5 incluye diversos métodos para trabajar con atributos numéricos, valores ausentes, datos con ruidos y para generar reglas a partir de árboles de decisión. El algoritmo J48 es la implementación del algoritmo C4.5 en Java del sistema Weka.[51]

Una de las ventajas de los árboles de decisión es que representan el modelo aprendido con una estructura que puede ser interpretada fácilmente por los usuarios, eso facilita la realización de ajustes al árbol de decisión en caso de necesidad.[52]

K-means (K-M)

El algoritmo k-media (K-M) es uno de los algoritmos más importantes de agrupamiento *flat*. Eso se corroboró en este estudio en la *Tabla 2*. Manning *et al.* afirman que el objetivo del algoritmo k-media es minimizar el error cuadrático medio de la distancia Euclidiana de los documentos con respecto al centroide del agrupamiento al cual pertenecen. El centro de un agrupamiento es definido como la media o centroide de los documentos que están en un agrupamiento.[53]

La medida para ver si un centroide representa bien a sus miembros del grupo, es la suma residual cuadrática (SRC), que es la sumatoria de la distancia cuadrática de todos los vectores con relación al centroide. El src es la función objetiva del k-media y el objetivo es minimizarla.[54]

El algoritmo de k-media no garantiza que será alcanzado el mínimo global al utilizar la función objetiva. Por ejemplo, si la muestra posee muchos valores atípicos no se ajustará bien a ninguno de los agrupamientos.[55] Otro problema del k-media es que debido a la selección aleatoria inicial de los centroides, los agrupamientos resultantes pueden variar mucho en calidad y finalmente el método k-media no sirve para descubrir agrupamientos con formas no convexas o agrupamientos con diferentes tamaños.[56]

Self-Organizing Maps (SOM)

Valluru B. Rao y Hayagriva Rao explican que la red neuronal mapas autoorganizados (SOM) es un tipo de aprendizaje de máquina competitivo no

51 Wanjun y Xiaoguang, *"Research on Text Categorization Based on Machine Learning", 254*, 254; Witten, Eibe y Mark, *Data Mining…*, 195.
52 *Ibid*., 8.
53 Manning, Raghavan y Schutze, *Introduction to Information Retrieval*, 360.
54 *Ibid*., 360-362.
55 *Ibid*., 363; Witten, Eibe y Mark, *Data Mining…*, 139.
56 Z. Wang, Z. Liu, D. Chen y K. Tang, "A New Partitioning Based Algorithm For Document Clustering", Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), *IEEE* (2011): 1741.

supervisado. Esta red, debido a su capacidad de autoorganizarse, tiene la habilidad de inferir relaciones y aprender más cuando más ejemplos le son presentados. La red neuronal SOM utiliza el aprendizaje competitivo. Esto significa que las neuronas compiten unas con otras y usa la estrategia "el vencedor se queda con todo".[57]

La red neuronal de mapas autoorganizados posee dos capas, la capa de entrada y la capa de salida. El tamaño de la capa entrada es determinado por el número de atributos seleccionados por el usuario para el aprendizaje. La capa de salida generalmente se organiza en forma bidimensional y es la encargada de procesar la información y formar el mapa de rasgos.

Los patrones de entradas diferentes activarán neuronas diferentes, pero los patrones de entrada similares van a activar las mismas neuronas y se obtiene como resultado un agrupamiento de los ejemplos que fueron presentados en las entradas.

En el SOM, en la capa de salida, se establecen conexiones laterales con las otras neuronas del mismo nivel. Las conexiones a favor son llamadas "conexiones de excitación", mientras las negativas son llamadas "de inhibición".[58]

El SOM es utilizado para clasificar entradas en diferentes categorías en diversas aplicaciones como el reconocimiento de voz y control robótico. El vector binario es frecuentemente utilizado en SOM para la clasificación de texto y procesos de agrupamientos, esto se debe a que los vectores binarios permite el tratamiento de vectores dispersos durante el procesamiento de mapas y procesamiento de texto.[59] Por su parte Feldman y Sanger afirman que SOM tiene la ventaja en el tratamiento y organización de conjuntos de datos extremamente grandes en volumen y en conectar relaciones.[60]

Association Rules (AR)

Las reglas de asociación (AR) permiten encontrar relaciones de asociación o correlación en un conjunto extenso de datos. Las ar funcionan con reglas que ligan fuertemente cualquier atributo y no solamente las clases de un conjunto de datos.[61]

57 Rao, Valluru B. y Hayagriva V. Rao, *C++* Neural *Network and Fuzzy Logic*, 2ª ed. (Nueva York: mis Press, 1995), 271.
58 Valluru B. y Hayagriva V., *C++ Neural Network and Fuzzy Logic*, 272-275.
59 Nicole Lang Beebe *et al.*, "Post-retrieval Search Hit Clustering to Improve Information Retrieval Effectiveness: Two Digital Forensics Case Studies", *Decision Support Systems –Elsevier* 51 (2011): 736.
60 Feldman y Sanger, *The Text Mining Handbook*, 213.
61 Witten, Eibe y Mark, *Data Mining: Practical Machine Learning Tools and Techniques,*, 72.

Hay dos etapas para crear las reglas de asociación: la primera para generar el conjunto de ítem con el valor mínimo de cobertura especificado y la segunda para cada conjunto de ítems determinar las reglas que tienen el valor mínimo de precisión especificado. La cobertura de las reglas de asociación es el número de casos que la regla consigue predecir correctamente. La precisión es el número de instancias que son predichos correctamente, expresadas como una proporción de todos los casos al cual se aplican.[62]

Uno de los problemas de las reglas de asociación es que las reglas generadas deben ser evaluadas manualmente para determinar si tienen sentido o no. Para evitar tener un número excesivo de reglas se restringe el número de reglas a aquellas que se aplican a un gran número de casos. Otros problemas son que las reglas de asociación envuelven generalmente atributos no numéricos y que el costo computacional para generar reglas de asociación dependen del grado de cobertura especificado.[63]

Latent Semantic Indexing (LSI)

La indexación semántica latente (LSI) es una técnica de agrupamiento suave (*soft clustering*) donde cada dimensión del espacio reducido es un agrupamiento y los valores que los documentos tienen en esas dimensiones es su fracción de pertenencia a ese agrupamiento.[64]
La indexación semántica latente es el proceso por el cual se aproxima la matriz término-documento C por una matriz de rango menor de esa matriz Ck para un valor *k* que es mucho menor que el rango de la matriz C. Esto produce una nueva representación para cada documento en la colección.[65]

La meta del LSI es encontrar el mejor subespacio de aproximación del espacio de los documentos, para minimizar el error de reconstrucción global (la diferencia de la norma de Frobenius entre la matriz original y la matriz aproximada).[66]

Las ventajas del LSI es que el vector disperso de consulta se torna vector denso de consulta en el espacio dimensional reducido al reducir el espacio dimensional k. También la exhaustividad aumenta, para valores adecuados de k la LSI consigue superar algunos de los desafíos de la sinonimia aumentando la precisión y funciona mejor en las aplicaciones donde existen poca superposición entre las consultas y los documentos.[67]

62 Witten, Eibe y Mark, Data Mining: *Practical Machine Learning Tools and Techniques*, 41.
63 *Ibid*., 123.
64 Manning, Raghavan y Schutze, *Introduction to Information Retrieval*, 417.
65 *Ibid*., 413.
66 Zhang, Yoshida y Tang, "A Comparative Study of TF*IDF, LSI…", 2760.
67 Manning, Raghavan y Schutze, *Introduction to Information Retrieval*, 417.

Algunas deficiencias de la indexación semántica latente es que surgen algunos valores negativos en la matriz de aproximación que no tiene explicación plausible y también el alto costo computacional de esta técnica.[68] Manning *et al.* afirman que el LSI no es bueno para expresar negación o para ejecutar condiciones Booleanas.[69]

Hierarchical Agglomerative Clustering (HAC)

El agrupamiento jerárquico aglomerativo (HAC) es un método de aprendizaje no supervisado. El HAC es un algoritmo de agrupamiento jerárquico de abajo para arriba que considera inicialmente cada documento como un agrupamiento individual y luego sucesivamente va fusionando los agrupamientos en pares, teniendo en cuenta la similitud entre ellos, hasta que todos los agrupamientos se aglomeren en un único agrupamiento que contiene todos los documentos presentados al algoritmo.[70]

Para el cálculo de similitud se utilizan diferentes técnicas de medidas de similitud como el agrupamiento de *link* simple, el agrupamiento de *link* completo, agrupamiento aglomerativo de media del grupo y agrupamiento por centroide.[71]

Para la visualización de los HAC es utilizado generalmente un dendrograma, donde cada unión (et al aglomeración) es representada por una línea horizontal entre los dos agrupamientos y el valor de la similitud entre esos dos agrupamientos esta especificado en el eje Y. El HAC tiene la suposición fundamental que es la monotonicidad en las operaciones de aglomeración (*merge operation*).[72]

Artificial Neural Network (ANN)

Las Redes Neuronales Artificiales (ANN) son redes que utilizan aprendizaje supervisado. Para Witten *et al.* las redes neuronales artificiales imitan las neuronas humanas que forman estructuras neuronales altamente interconectadas para realizar tareas complejas de clasificación.[73] Esas conexiones de varias neuronas en estructuras jerárquicas permiten representar límites no lineales de decisiones.

68 Zhang, Yoshida y Tang, "A Comparative Study of TF*IDF, LSI…", 2760.
69 Manning, Raghavan y Schutze, *Introduction to Information Retrieval*, 417.
70 *Ibid*., 378; S. C. Suh, N. B. Pabbisetty y S. G. Anaparthi, "Extraction of Meaningful Rules in a Medical Database", en Zhang, Yagang, ed., *Machine Learning* (Vukovar: InTech, 2010), 416-417.
71 Manning, Raghavan y Schutze, *Introduction to Information Retrieval*, 380-392; Witten, Eibe y Mark, *Data Mining: Practical Machine Learning Tools and Techniques,* 275-276.
72 *Ibid*., 378.
73 Witten, Eibe y Mark, *Data Mining: Practical Machine Learning Tools and Techniques*, 232.

Las redes neuronales son una representación simplificada en forma de gráfico de una malla de neuronas en el cerebro humano. Los nodos son las unidades de procesamiento, y los vínculos representan las conexiones sinápticas. Para simular la fuerza de las conexiones sinápticas, un peso es asociado con cada conexión entre los nodos de la red neuronal. A cada instante, el estado de un nodo es definido por su nivel de activación. Dependiendo de ese nivel de activación el nodo envía una señal al nodo vecino; la fuerza de esa señal dependerá del peso asociado con esa conexión.[74] Si los valores de los pesos de una red neuronal convergen con los datos del entrenamiento se dice que el problema puede ser representado por una red neuronal.[75]

Los clasificadores de texto basados ANN están formados por unidades de entrada que representan los términos existentes en los documentos, la capa oculta y las unidades de salida que representan las categorías de interés. La cantidad de neuronas en la unidad de salida dependerá de la cantidad de categorías en las que los patrones de entrada deben ser clasificados.[76]

De acuerdo con Peter Brevern Sivarao *et al*., las ANN poseen gran capacidad en tareas de clasificación y reconocimiento de patrones, además son adecuadas en soluciones que requieran conocimiento que son difíciles de especificar, pero existen suficientes datos registrados.[77] Las ANN tienen la capacidad de modelar sistemas no lineales, presentan robustez en los datos con ruidos y poseen capacidad de modelaje genérico.

Algunos problemas de las redes neuronales multicapa con entrenamiento de retropropagación (principalmente si la estructura de la red neuronal está superdimensionada) son que puede ocurrir un sobreajuste (*overfitting*) con los datos de ejemplo utilizados para el entrenamiento y también que la capa oculta de la red neuronal actúa como una caja negra que dificulta la extracción de reglas de la red neuronal una vez entrenada.[78]

Conclusión

Esta investigación permitió identificar en los años estudiados (2010 y 2011) las principales técnicas de aprendizaje de máquina aplicadas a la minería de texto. Se observó un predominio de la técnica *support vector machine* en las publicaciones que fueron evaluadas tanto como técnica principal (el 22.72%)

74 R. Baeza-Yates y B. Ribeiro-Neto, *Modern Information Retrieval* (Nueva York: Addison-Wesley, 1999), 46-47.
75 B. Yegnanarayana, *Artificial Neural Networks* (Nueva Delhi: Prentice-Hall India, 2005), 28.
76 Feldman y Sanger, *The Text Mining Handbook*, 75.
77 P. Brevern Sivarao, El-Tayeb N.S.M. y Vengkatesh V.C., "Neural Network Multi Layer Perceptron Modeling for Surface Quality Prediction in Laser Machining" , en Yagang Zhang, ed., *Application of Machine Learning* (Vukovar, Croatia: InTech, 2010), 52.
78 Witten, Eibe y Mark, *Data Mining: Practical Machine Learning Tools and Techniques,* 240.

como comparativa (el 20.59%) y también fue la más mencionada en la sección de revisión de literatura de los artículos analizados (16.88%).

Otras técnicas que se destacan son *k-means*, *k-nearest neighbors* y *naive Bayes* que aparecen con porcentajes similares en la categoría de técnica principal (aproximadamente el 12.12%); como técnica comparativa las tres técnicas aparecen con el mismo porcentaje de 14.71% y en la revisión de literatura K-NN tiene el 14.29%, NB 12.99% y K-M 10.39%.

Al realizar la comparación en pares de las técnicas de aprendizaje de máquina, los pares más frecuentes encontrados en los artículos son el par de *support vector machine* / *naive Bayes* es el de mayor predominio con frecuencia 12, seguido del par *support vector machine* / *k-nearest neighbors* con frecuencia 8 y de los pares *support vector machine* / *decision trees* y *k-nearest neighbors* / *naive Bayes*, ambos con frecuencia 6.

Concluimos, a través de este estudio, que la técnica de *support vector machine* fue la más destacada en los artículos estudiados. Eso se podría explicar debido a que el SVM tiene gran versatilidad para diversos tipos de aplicaciones, puede ser empleada cuando se tienen pocos ejemplos etiquetados de entrenamiento, permite utilizar diversas técnicas de kernel y presenta la capacidad de trabajar con datos de alta dimensión como entrada y conseguir dar respuestas consistentes.

Por su parte, *k-mean* es una técnica de aprendizaje no supervisado, considerada como la de agrupamiento más simple para el problema de agrupamiento de los datos que le son presentados. Ésta permite que el investigador especifique la cantidad de agrupamientos que quiere obtener.

La técnica de *k-nearest neighbors* es un algoritmo con gran facilidad de implementación que permite clasificar documentos desconocidos llevando en consideración sus *k* vecinos más próximos, sin embargo el costo computacional se eleva a medida que el tamaño de los datos de ejemplo utilizados en el entrenamiento crece.

Finalmente, *naive Bayes* es una técnica simple, con semántica clara, para representar, usar y aprender conocimiento probabilístico. NB es una técnica popular para clasificación de documentos por ser una técnica simple, rápida y segura. Esta técnica puede presentar desvíos en el proceso de aprendizaje, por considerar que los atributos son independientes para una determinada clase.

Referencias

Aljaber, B., D. Martinez, N. Stokes y J. Bailey. "Improving MeSH Classification of Biomedical Articles Using Citation Contexts". *Journal of Biomedical Informatics–Elsevier* 44 (2011): 881-896.

Baeza-Yates, R. y B. Ribeiro-Neto. *Modern Information Retrieval*. Nueva York: Addison-Wesley, 1999.

Beebe, N. L., J. Guynes Clark, G. B. Dietrich, M. S. Ko y D. Ko. "Post-retrieval Search Hit Clustering to Improve Information Retrieval Effectiveness: Two Digital Forensics Case Studies". *Decision Support Systems–Elsevier* 51 (2011): 732-744.

Berry, M. W. y J. Kogan. *Text Mining: Applications and Theory*. Chicester, GB: Wiley, 2010.

Bin, D., S. Peiji y Z. Dan. "E-Commerce Reviews Management System Based on Online Customer Reviews Mining". International Conference on Innovative Computing and Communication and 2010 Asia-Pacific Conference on Information Technology and Ocean Engineering. *IEEE Computer Society* (2010): 374-377.

Chapelle, O., B. Schölkopf y A. Zien. *Semi-Supervised Learning*. Cambridge, MA: MIT Press, 2006.

Chen, H. y M. Chau. "Web Mining: Machine Learning for Web Applications". En Blaise Cronin, ed., *Annual Review of Information Science and Technology* 38 (2004): 293.

Feldman, R. y J. Sanger. *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge: Cambridge University Press, 2007.

Kim, Sang-Bum, H. Kyoung-Soo, H. C. Rim y S. Hyon Myaeng. "Some Effective techniques for Naïve Bayes Text Classification". IEEE *Transactions on Knowledge and Data Engineering* 18, no. 11 (2006): 1457-1466.

Liu, Y. y H. Tong Loh. "Domain Concept Handling in Automated Text Categorization", 5th IEEE Conference on Industrial Electronics and Applications, IEEE, 2010, 1543-1549.

Lu, F. y Q. Bai. "A Refined Weighted K-Nearest Neighbors Algorithm for Text Categorization", *IEEE* (2010): 326-330.

Manning, C. D., P. Raghavan y Hinrich Schutze. *Introduction to Information Retrieval*. Cambridge: Cambridge University Press, 2008.

Mitchell, T. M. *Machine Learning*. New York: McGraw Hill, 1997.

Nilsson, N. J. *Introduction to Machine Learning, an Early Draft of a Proposed Textbook*, Consultado el 12 de noviembre de 2013. http://robotics.stanford.edu/people/nilsson/MLBOOK.pdf.

Pino Díez, R., A. Gómez Gómez y N. de Abajo Martínez. *Introducción a la inteligencia artificial: sistemas expertos, redes neuronales artificiales y computación evolutiva*. Oviedo: Servicio de Publicaciones Universidad de Oviedo, 2011.

Pitigala, S., C. Li y S. Seo. "A Comparative Study of Text Classification Approaches for Personalized Retrieval in PubMed". IEEE International Conference on Bioinformatics and Biomedicine Workshops, IEEE, 2011, 919-921.

Rao, V. B. y H. V. Rao. *C++ Neural Network and Fuzzy Logic*. 2ª ed. New York: MIT Press, 1995.

Sivarao, P. Brevern, El-Tayeb N.S.M. y Vengkatesh V.C. "Neural Network Multi Layer Perceptron Modeling for Surface Quality Prediction in Laser Machining". En *Application of Machine Learning*, ed. Yagang Zhang, 51-61. Vukovar, Croatia: InTech, 2010.

Suh, S. C., N. B. Pabbisetty y S. G. Anaparthi. "Extraction of Meaningful Rules in a Medical Database", en Yagang Zhang, ed., *Machine Learning*, 411-426. Vukovar, Croatia: InTech, 2010.

Torunoglu, D., E. Çakırman, M. Can Ganiz, S. Akyokus y M. Zahid Gürbüz. "Analysis of Preprocessing Methods on Classification of Turkish Texts", *IEEE* (2011): 112-117.

Vidhya, K.A y G. Aghila. "Hybrid Text Mining Model for Document Classification", *IEEE* 1 (2010): 210-214.

Wang, Z., L. Zhijing, C. Donghui y T. Kai. "A New Partitioning Based Algorithm For Document Clustering". Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), *IEEE* (2011): 1741-1745.

Wanjun, Y. y S. Xiaoguang. "Research on Text Categorization Based on Machine Learning", Advanced Management Science (ICAMS). IEEE International Conference 2, IEEE, 2010, 253-255.

Weiss, S. M., N. Indurkhya, T. Zhang y F. Damerau. *Text Mining: Predictive Methods for Analyzing Unstructured Information. Texts in Computer Science*. USA: Springer, 2004, 118.

Witten, I. H., F. Eibe y M. A. Hall. *Data Mining: Practical Machine Learning Tools and Techniques*. San Francisco: Morgan Kaufmann, 2011.

Yegnanarayana, B. *Artificial Neural Networks*. New Delhi: Prentice-Hall of India, 2005.

Zhang, W., T. Yoshida y X. Tang. "A Comparative Study of TF*IDF, LSI and Multi-words for Text Classification", *Expert Systems with Applications* 38 (2011): 2758-2765.

Para citar este artículo:

Godoy Viera, Angel Freddy. "Técnicas de aprendizaje de máquina utilizadas para la minería de texto". *Investigación Bibliotecológica* 31, no. 71 (enero-abril 2017): 103-126.