Revista Signos. Estudios de Lingüística ISSN 0718-0934
© 2014 PUCV, Chile · DOI: 10.4067/S0718-09342014000300001 · 47(86) 346-364

Clasificación automática de textos usando redes de palabras*

Automatic text classification using words networks

Juan Pablo Cárdenas
Pontificia Universidad Católica de Valparaíso
Chile
juanpablo@analitic.cl

Gastón Olivares
Pontificia Universidad Católica de Valparaíso
Chile
gastonolivares@gmail.com

Rodrigo Alfaro
Pontificia Universidad Católica de Valparaíso
Chile
rodrigo.alfaro@ucv.cl

Recibido: 24-V-2013 / **Aceptado**: 13-XII-2013

Resumen

El objetivo de este trabajo es proponer un algoritmo para la clasificación automática de textos, como una alternativa a los tradicionalmente utilizados en esta tarea. El clasificador propuesto considera la dependencia entre las variables predictoras (palabras o términos), algo que los clasificadores de texto comúnmente utilizados no hacen. La dependencia entre estas variables queda plasmada en forma de enlaces en grafos de palabras co-ocurrentes, objetos utilizados para entrenar el clasificador y además estimar la categoría de un texto desconocido. Los resultados obtenidos al clasificar automáticamente el sentido positivo, negativo o neutral de más de 1.000 mensajes de *Twitter* escritos en español, en distintos contextos (temas), muestran que el algoritmo, además de ser una propuesta novedosa para la clasificación automática de textos, tiene un desempeño, al menos, similar al de otros tradicionalmente utilizados en este tipo de problemas, como las Máquinas de Soporte Vectorial o algoritmos de estadística Bayesiana.

Palabras Clave: Clasificación automática de textos, redes de palabras, algoritmo, inteligencia artificial, inteligencia computacional.

Abstract

The purpose of this paper is to propose an algorithm for automatic text classification, as an alternative for those traditionally used for this task. The proposed classifier considers dependence between predictor variables (words or terms), an approach ignored by traditional classifiers. The dependence between predictor variables is captured as links of co-ocurrent words networks, objects that are used for training the classifier and also estimate the category of an unknown text. The results obtained from the automatic sentiment classification of more than 1,000 Twitter messages in positive, negative or neutral categories, and considering different context (topics), show that the proposed classifier, besides being a novel proposal, performs well compared to other algorithms traditionally used in automatic text classification such as Support Vector Machines or algorithms based in Bayesian statistic.

Key Words: Automatic text classification, networks, algorithm, artificial intelligence, computational intelligence.

INTRODUCCIÓN

La clasificación automática puede definirse como la acción ejecutada por un sistema artificial sobre un conjunto de elementos para ordenarlos en clases o categorías. Si bien los elementos a clasificar pueden ser de cualquier tipo, es la clasificación automática de textos una de las áreas de investigación que ha cobrado mayor importancia en los últimos años debido, en parte, a los grandes volúmenes de textos digitales que se almacenan en bases de datos empresariales, páginas *web* y redes sociales.

La clasificación automática de textos ha estado ligada históricamente al desarrollo de Máquinas de Aprendizaje, una línea de la Inteligencia Artificial y la Inteligencia Computacional que se basa en el desarrollo de algoritmos que 'aprenden' o reconocen patrones recurrentes en cada clase a partir de un gran volumen textos de entrada, previamente clasificados por humanos.

La clasificación de textos tiene entre sus particularidades una alta dimensionalidad y desbalance entre categorías, lo que la distingue y la hace ser más complicada que la de otro tipo de datos, como por ejemplo imágenes. Por esta razón, es un problema abierto de gran desarrollo en las Ciencias de la Informática. De hecho, en las últimas dos décadas, el manejo automático de documentos electrónicos se ha transformado en el mayor campo de investigación en esta área de la ciencia. Es precisamente dentro de este escenario, el de grandes volúmenes de información digital vinculada a la Inteligencia Artificial, en el que se inserta este trabajo que tiene como objetivo proponer un algoritmo alternativo a los ya existentes, basado en grafos de palabras, para la clasificación de textos digitales.

El trabajo se estructura de la siguiente forma. En la siguiente sección se introduce brevemente la teoría tras la clasificación automática, las Máquinas de Aprendizaje y las redes de palabras. La propuesta de un algoritmo para clasificar textos, así como los resultados de su desempeño en la clasificación de textos digitales, se presentan en las siguientes secciones. Finalmente, se presentan las conclusiones más relevantes del trabajo desarrollado.

1. Marco teórico

La clasificación automática de textos es por lo general un proceso supervisado (Baeza-Yates & Ribeiro-Neto, 1996). Esto significa que requiere de un conjunto de documentos previamente clasificados por expertos humanos que funcionan como entrenamiento para el sistema. Así, un conjunto de documentos clasificado por un humano en cierta categoría sirven para que el clasificador automático genere una clasificación propia frente a un documento desconocido. El desempeño del clasificador automático dependerá de qué tan similar sea esta clasificación respecto a la humana, lo que se evalúa con matrices de confusión y otras métricas (Sebastiani, 2002).

1.1. Clasificación automática de textos como Máquinas de Aprendizaje

Bajo el marco formal de Máquinas de Aprendizaje se define como D el espacio de entrada que representa los documentos de textos y C el espacio de salida de un proceso de clasificación realizado por un humano, donde cada par $\{d_i,c_j\}$ corresponden al documento y la categoría en la que este fue clasificado, respectivamente. Asimismo, se define Z como el conjunto de todos estos pares de documentos y sus clasificaciones, $z_i=(d_i,c_j)$. Los valores de Z corresponden a los datos que sirven de ejemplo para que la Máquina de Aprendizaje utilice en su proceso de entrenamiento y prueba.

El principal desafío del proceso de aprendizaje automático es obtener una máquina que posea buena capacidad de generalización, es decir, que no solo aprenda a clasificar los ejemplos Z, conocidos utilizados en su proceso de entrenamiento, sino también, que sea capaz de construir un modelo general que permita clasificar bien nuevos ejemplos desconocidos. Para asegurar una buena capacidad de generalización, la Teoría del Aprendizaje Estadístico (Vapnik, 1989) define una Función Riesgo $R(g(z))$ como aquella que agrupa funciones $g(z)$ del clasificador automático representando la distancia de estas funciones respecto a las reales. Cada conjunto de funciones g opera como clasificador con otro conjunto de parámetros. Al conjunto de funciones g y sus parámetros se les denomina Funciones Admisibles, de forma que $R(g(z))=R(\alpha)$, donde a es el conjunto de parámetros del clasificador automático. R evalúa la distancia acumulada entre las funciones del clasificador respecto a la función real subyacente, por lo que $R(\alpha)=\int Q(z,\alpha)dz$, entonces $Q(z,\alpha)=L(z,g(z,\alpha))$, siendo L la Función de Pérdida que relaciona los datos ya clasificados con la clasificación automática.

En problemas de clasificación automática, cuando $L \approx 1$ se dice que el resultado de esta clasificación no es bueno, ya que es distinto a la realidad z, usando una determinada función con ciertos parámetros, ambos pertenecientes a $g(z,\alpha)$. Por el contrario, cuando $L \approx 0$ se dice que la clasificación automática es similar a la real y, por lo tanto, el resultado es bueno (Mitchell, 1997). Entonces, los algoritmos para entrenar las Máquinas de Aprendizaje buscan encontrar el conjunto de parámetros α para la función g que minimice L. Estos algoritmos pueden estar basados en procesos de optimización, o también, en heurísticas.

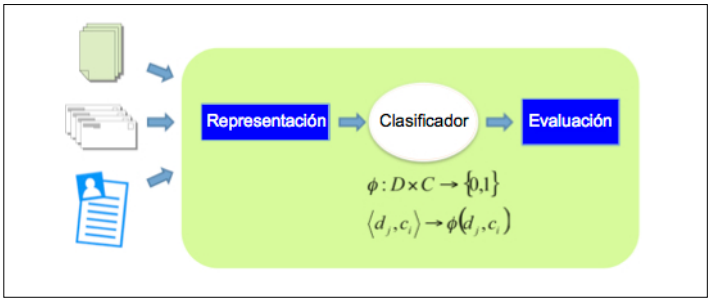

Figura 1. Esquema de la clasificación automática de textos. Los documentos son representados para entrar al algoritmo clasificador cuyos resultados son evaluados.

La Figura 1 muestra un esquema del proceso automático de clasificación de un texto. Como se aprecia, la representación del documento, que no es otra cosa que una estructuración particular de este, es la entrada que recibe el clasificador definido formalmente como una función binaria del tipo $\varphi: D \times C \rightarrow \{0,1\}$. En esta función, D corresponde al conjunto de documentos y C al conjunto de categorías, de forma que si la función $\varphi=1$ el documento (texto) pertenece a la categoría, mientras que si $\varphi=0$, no pertenece (Sebastiani, 2002).

Varios han sido los tipos de representación propuestos a la fecha (Harish, Guru & Manjunath, 2010). Una de las más extendidas, sobre todo por su simpleza, es la llamada 'Bolsa de Palabras' (*bag of words*, BoW). La representación BoW se usa para formar un vector con la frecuencia de los términos, o variables predictoras presentes en el documento. Dentro de sus limitaciones está el carecer de información sobre relaciones entre estos términos (por ejemplo, relaciones entre palabras), así como de relaciones semánticas que existen entre éstos a lo largo del documento.

Frente a esta simplificación, otras representaciones que consideran el peso estadístico en las frecuencias de los términos (Lan, Tan, Su & Lu, 2009), o que mediante ontología (Hotho, Maedche & Staab, 2001) rescatan las relaciones semánticas entre estos, también han sido propuestas. Es necesario remarcar que si bien cada una de estas supone un aporte, también presentan sus propias limitaciones, ya sean de implementación, de complejidad computacional o de requerimientos de memoria (Harish et al., 2010).

Con un enfoque totalmente distinto al de las representaciones antes mencionadas, Choudhary y Bhattacharyya (2003) proponen una que transforma el documento en un grafo dirigido, en donde los nodos corresponden a palabras y los enlaces a sus relaciones. Para los investigadores ese grafo es una representación de un documento escrito en un lenguaje natural en un lenguaje formal, llamado *Universal Networking Language* (UNL) (Uchida, Zhu & Della Senta, 1995). Esta representación, junto a otras propuestas más actuales (Jin & Srihari, 2007; Zhou, Zhang & Yang, 2010), se asemejan en su esencia, tal como se verá más adelante, a la propuesta de representación planteada en este trabajo.

Así como sucede con las representaciones, muchos clasificadores de textos también han sido propuestos. El más simple de estos es el llamado clasificador *Naïve Bayes,* correspondiente a un modelo estructural y a un conjunto de probabilidades condicionales. La estructura del modelo es la de un grafo dirigido en el que cada nodo representa un atributo y cada enlace representa la dependencia entre atributos expresada por una probabilidad condicional por cada nodo. Si se asume que todos los atributos son independientes dada la categoría, el clasificador es de este tipo (Lewis, 1998).

A pesar que Zhang (2004) aportó con fundamentos teóricos que avalan a este 'ingenuo' método en la clasificación de problemas reales, más tarde, Caruana y Niculescu-mizil (2006), demostraron que su desempeño es inferior a otros métodos. A pesar de esto, está ampliamente extendido, posiblemente por la ventaja que significa necesitar un pequeño entrenamiento para obtener buenos resultados.

Otro tipo de algoritmo utilizado comúnmente en clasificación automática de textos son las Máquinas de Soporte Vectorial (Joachims, 2002). Estas corresponden a máquinas de aprendizaje que toman distintas características de los elementos que se quieren clasificar y los llevan a un espacio vectorial multidimensional. Es en este espacio, donde el algoritmo identifica, de forma óptima, un hiperplano que separa a los vectores de una clase del resto. Es en ese concepto de 'separación óptima' donde reside la característica fundamental de estos algoritmos. Las SVMs son bastante populares porque tienen un funcionamiento relativamente sencillo, asociado a una serie de propiedades teóricas y prácticas atractivas (Vapnik, 1995).

Otros clasificadores de texto destacados, son el clasificador de vecinos próximos, los árboles de decisión y las redes neuronales (Sebastiani, 2002). Todos estos, al igual que *Naïve Bayes* y las SVMs, utilizan por lo general representaciones vectoriales de documentos (Harish et al., 2010), tales como BoW.

A pesar que el supuesto de independencia entre variables predictoras, adoptado tanto por las representaciones y clasificadores tradicionales, ha permitido obtener buenos desempeños de clasificación, la naturaleza relacional de los datos reales parece chocar, al menos intuitivamente, con este enfoque. Por ejemplo, si se quiere clasificar

un computador en una determinada marca, no es razonable pensar que su diseño, valor de mercado, color, capacidad, etc., son características que deban ser tratadas por separado por un clasificador. Esta es la motivación tras la propuesta de este trabajo, una representación y clasificación de textos basada en grafos, llamados redes de palabras, que se hace cargo de esta simplificación.

1.2. Redes de palabras

En términos generales, una red de palabras es la transformación de un texto, escrito en un determinado lenguaje, en un grafo *G(N,E)*, donde *G* es el grafo, o red, compuesto por *N* palabras (o términos) distintas y *E* enlaces que las relacionan en un texto (Cárdenas, Losada, Moreira, Torre & Benito, 2011). Muchas redes de palabras pueden extraerse desde un texto, sin embargo, este trabajo se enfoca en aquellas que buscan rescatar la gramática de la lengua. De esta forma, si la palabra *i* y la palabra *j* aparecen adyacentes en un texto, en el grafo *G* son palabras enlazadas porque su co-ocurrencia en el texto depende de la gramática de una lengua particular.

Es necesario remarcar que en este caso no se está haciendo referencia a la direccionalidad del enlace, es decir, a la posición de la palabra *i* respecto a *j*. Ahora, para el grafo *G* se define una matriz de adyacencia *A(G)* como una representación bidimensional de las relaciones entre todas las palabras distintas presentes en el texto. En esta matriz, cuando A_{ij}=1 existe un enlace entre las palabras *i* y *j*, mientras que si A_{ij}=0 ese enlace no existe (*i.e.*, ambas palabras nunca aparecen juntas en el mismo texto).

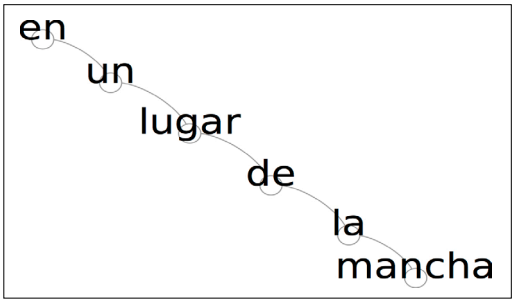

Figura 2. Red de palabras co-ocurrentes construida a partir de la frase "en un lugar de la mancha".

La Figura 2 muestra una red de palabras simple, construida a partir de la frase extraída de la obra de Miguel de Cervantes Saavedra, El ingenioso hidalgo don Quijote de la Mancha. La frase contiene seis palabras distintas. Dentro de estas, la palabra p3='lugar' tiene como entrada la palabra p2='un', y como salida la palabra p4='de'. Ahora, en el grafo *G*, ya que $A_{2,3}$=1 y $A_{3,4}$=1, la conectividad de p3 es k_3=1+1=2.

Si el texto está compuesto solo por palabras distintas, como en el ejemplo, la red es una simple cadena como la observada en la Figura 1. En este tipo de redes, a

excepción de las palabras ubicadas en los extremos, la conectividad modal es k=2. Sin embargo, este escenario es muy improbable en textos largos ya que, por lo general, en los lenguajes la frecuencia en el uso de las palabras es distinta (Zipf, 1965). Así, si se introduce una nueva sentencia al texto anterior, por ejemplo "y en la casa de don quijote", tres palabras repetidas y cuatro diferentes son añadidas. En un proceso de (re)generación de la red de palabras, aquellas repetidas mantienen la misma identidad (número p) que tomaron en su primera aparición en el texto, aunque pueden estar conectadas a palabras distintas o muchas veces a la misma (imaginar el caso de un nombre propio compuesto por dos palabras). La nueva red generada en este caso se muestra en la Figura 3. Notar que las palabras 'en', 'la' y 'de' tienen una conectividad, k=3, superior al resto.

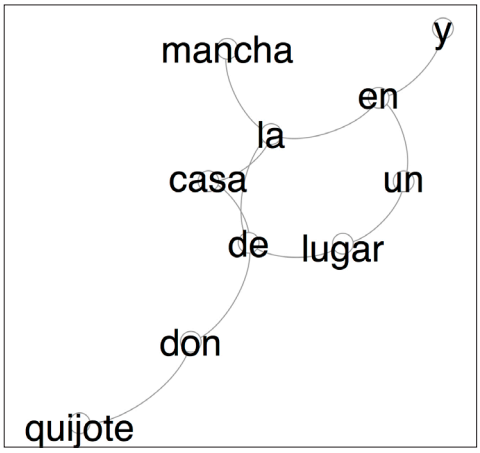

Figura 3. Red de palabras co-ocurrentes construida a partir de las frases: "en un lugar de la mancha" y "y en la casa de don quijote".

El método para obtener redes de palabras, antes descrito, no considera diferencias entre palabras escritas con letras mayúsculas o minúsculas, eliminando el efecto de la puntuación en el texto. Esta simplificación no permite distinguir entre nombres propios y lugares usados en diferente contexto. Por otro lado, el método asume que la puntuación impide el enlace entre palabras adyacentes. Por esta razón, si las palabras i y j están separadas por cualquier tipo de puntuación entonces A_{ij}=0.

2. Marco metodológico

Con el objetivo evaluar el desempeño del clasificador de textos basado en redes de palabras, se utilizaron, para entrenar y clasificar, más de 1.000 mensajes del microblog *Twitter*, escritos en español y correspondientes a distintos contextos. Gracias a su API (*Application Programming Interface)*, que permite descargar los mensajes emitidos por los usuarios de la plataforma, *Twitter* se ha transformado en la fuente de datos más popular para investigaciones sociales (Leetaru, Wang, Cao, Padmanabhan & Shook,

2013). Es por esta razón, sumada a sus particularidades en el uso del lenguaje (Bryden, Funk & Jansen, 2013), que los datos obtenidos desde esta 'red social' se presentan como un buen desafío para evaluar el clasificador propuesto y compararlo con otros tradicionalmente utilizados en estas tareas.

2.1. Clasificador basado en redes de palabras

En este trabajo se propone un algoritmo de clasificación y una representación de textos basados en redes de palabras, tales como aquellas descritas en la sección anterior. De esta forma, se define un grafo G_i como aquella red de palabras co-ocurrentes construida a partir de un conjunto de textos clasificados en la categoría *i* por un humano. Dicho grafo *G* corresponde al entrenamiento para dicha categoría. Se desprende de lo anterior que habrá tantos grafos *G* como categorías de textos hayan sido clasificadas por personas. Por otro lado, se define otro tipo de grafo, *h*, correspondiente a la red de palabras que representa al texto a clasificar.

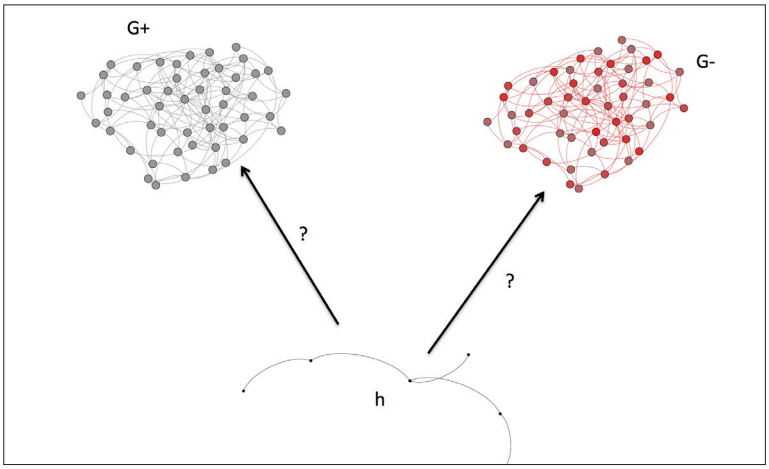

Figura 4. Red de palabras *h*, que representa al texto sin clasificar, evaluando su reconstrucción en dos redes de entrenamiento *G+* y *G-*, construidas con textos clasificados por un experto humano en la categoría '+' y la categoría '-'.

Antes de comenzar el proceso de clasificación, una serie de procesos de 'limpieza' son aplicados a las representaciones *G* y *h*. Esta limpieza se traduce en el colapso de ciertos nodos, antes distintos, en uno solo. Una aproximación similar a la planteada por Choudhary y Bhattacharyya (2003) pero más tolerante a la diversidad de términos. El primero de estos procesos es llevar distintas variantes de una misma palabra a su raíz, esto para reducir la variabilidad del lenguaje producto de, por ejemplo, la conjugación verbal. Para realizar lo anterior, se utilizó el corpus de la Real Academia de la Lengua Española (2013), compuesto por más de 7 mil palabras junto a su frecuencia de uso. En primer lugar, este corpus se dividió en dos partes de acuerdo a la frecuencia: palabras muy utilizadas y menos utilizadas. El primer grupo, compuesto por cerca de

50 palabras, se les consideró inútiles para la clasificación de sentimiento, debido a su gran frecuencia de aparición. Se excluyeron de esta lista las palabras 'si', 'no', 'años' y 'está'. En el modelo propuesto, se eliminaron de las redes *G* y *h* todas aquellas relaciones entre dos palabras de este tipo. El otro grupo de palabras, con mayor frecuencia (>1), fue usado para el proceso antes descrito de llevarlas a una raíz común, de acuerdo al siguiente procedimiento. Inicialmente, todas las palabras de los grafo *G y h* cuyo sufijo fuese 'ar', 'er' o 'ir' (verbos) son extraídas para su colapso a una raíz común. A modo de ejemplo, a la palabra 'saltando' se le saca la raíz 'salt', posteriormente se compara si es igual al verbo sin su sufijo (saltar sin contar 'ar'), al ser igual, la palabra 'saltando' es cambiada automáticamente por 'saltar'. Cabe destacar que este método funciona bien para los verbos regulares, sin embargo varias excepciones se consideran para verbos irregulares.

El segundo paso considera las palabras plurales (terminadas con 's' o 'es'). A estas se les elimina el sufijo para compararlas con su equivalente singular, realizando el cambio de manera automática. También se consideran casos especiales, donde la antepenúltima letra es 'n', 'l', 'r' o 'd'.

Finalmente, una vez que las palabras han sido llevadas a su raíz común, se procede a otro tipo de limpieza. Así, todas aquellas direcciones de *Internet* (URLs) pasan a ser el término 'URL', todos los dígitos pasan a ser el término 'NÚMERO', entre otras normalizaciones.

Ahora, una vez limpiados ambos grafos comienza el proceso de clasificación propiamente tal (Figura 4). De acuerdo a este, el texto con categoría desconocida será clasificado según el siguiente criterio: corresponderá a una determinada categoría si el grafo *h*, que lo representa, puede ser reconstruido[1] en el grafo *G* de dicha categoría.

Ya que existe la posibilidad que *h* pueda ser reconstruido en más de una categoría de grafo *G*, se define un 'costo de construcción' como una medida de decisión. Así, la categoría en la que el costo de construcción sea menor será la que determine la categoría del texto clasificado.

El costo de construcción *C* para un texto *h* en una determinada categoría se define como,

$$C_h \approx (l^{\alpha} / P^{\beta}) + v,$$

donde *l* es el promedio de las distancias más cortas entre todos los pares de palabras co-ocurrentes de *h* presentes en *G* (*i.e.*, número de nodos mínimo que debe recorrerse en *G* para unir cada par), *P* la cantidad de palabras de *h* presentes en *G*, α y β las ponderaciones de ambas medidas y *v* un valor (negativo) obtenido a partir de una valoración de las palabras y símbolos presentes en *h*. De esta forma mientras más palabras de *h* estén en *G,* y menor sea la distancia entre estas en *G*, menor será el costo

de (re)construcción de *h* en *G*. El método de mapeo de un grafo en otro es similar al propuesto por Namata y Getoor (2009), sin embargo, el de este trabajo considera el mapeo de nodos y enlaces al mismo tiempo, en la misma ecuación, y no como una secuencia de comprobaciones.

3. Resultados

Se realizaron una serie de experimentos para clasificar los textos digitales del microblog *Twitter*. Todas las evaluaciones que se presentan a continuación corresponden al modelo del clasificador con parámetros $\alpha=1$ y $\beta=2$ según la ecuación de costo presentada en la sección anterior. Para evaluar el desempeño del clasificador se utilizaron las tradicionales medidas de precisión (exactitud) y cobertura (Sebastiani, 2002) respecto a la clasificación realizada por un humano.

Un primer grupo de experimentos estuvo enfocado en el estudio de un proceso 'tiempo-real' de entrenamiento y clasificación que siguiese la dinámica de *Twitter*. Para esto se 'recolectaron' y clasificaron (manual y automáticamente) mensajes o *tweets* diariamente para diversos temas. La diferencia entre la clasificación manual y automática dio como resultado un desempeño diario del clasificador. El Gráfico 1 muestra la dinámica de este proceso para tres categorías de clasificación de *tweets* (positivos, negativos y neutrales) en un contexto particular (tema y momento) durante cerca de un mes.

El entrenamiento utilizado por el clasificador fue aumentando con el tiempo ya que correspondió al conjunto de *tweets* clasificados por un humano desde t_0 (inicio del entrenamiento) hasta un día antes de cada evaluación, t-1. Debido a esto el desempeño diario del clasificador está desfasado un día en la gráfica.

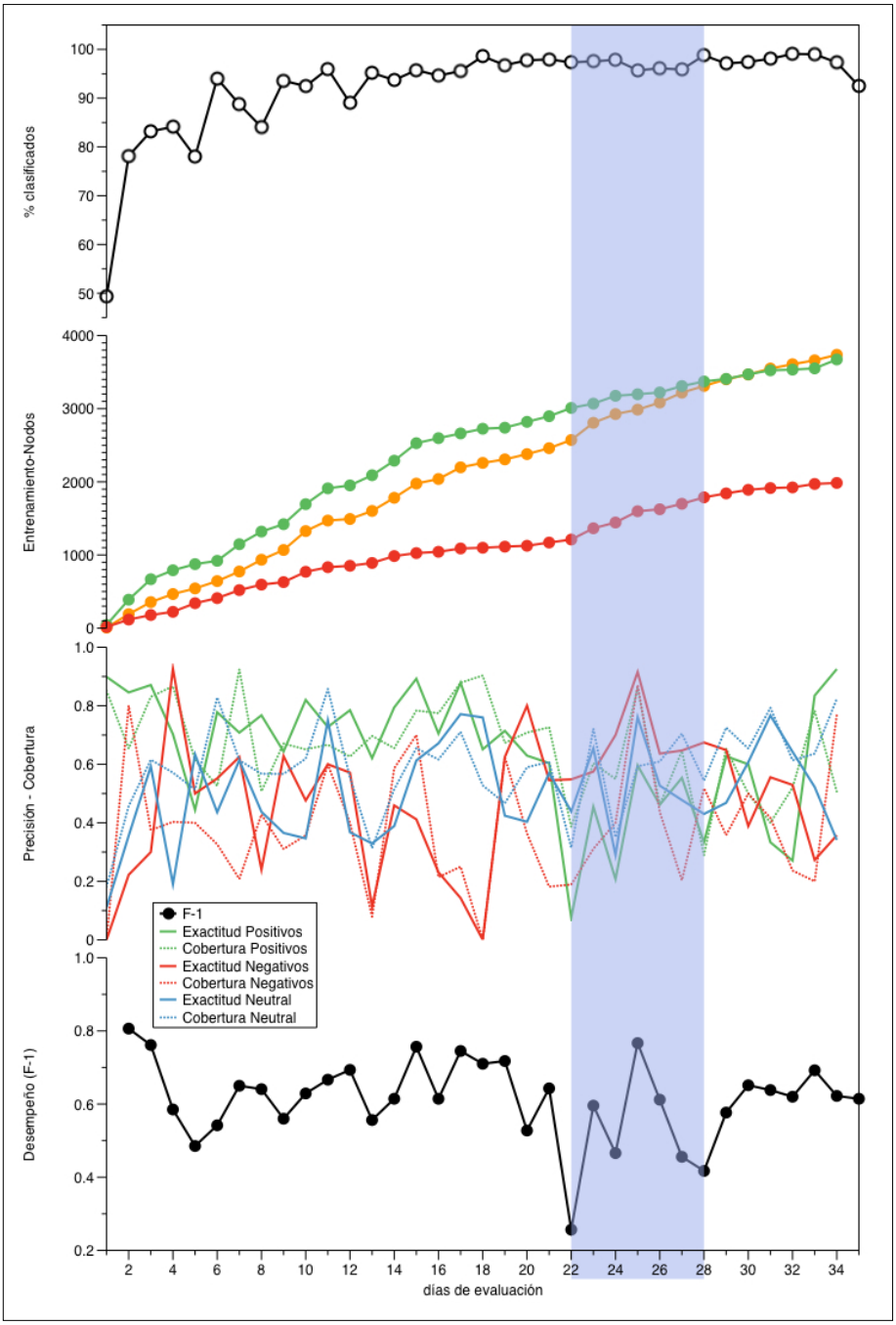

Gráfico 1. Dinámica del proceso de clasificación automática. Micro F1 (gráfica inferior), 'precisión' y 'cobertura' de cada categoría (gráfico medio inferior), tamaño (nodos) del entrenamiento (gráfica media superior) y porcentaje de clasificación (gráfico superior).

Cuando se observa la precisión y la cobertura de cada categoría (gráfica media inferior), se evidencia que la clasificación automática de textos no es una tarea fácil, más aún en *Twitter* donde las particularidades en el uso del lenguaje son significativas (Manley, 2012; Bryden et al., 2013). Sin embargo, para la clasificación de *tweets* positivos (línea verde), la precisión y la cobertura son, por lo general, altas y estables, seguidas por la categoría neutral (línea azul) y más abajo la categoría negativa (línea roja). A pesar de estas diferencias, el desempeño global[2] del clasificador (Micro F1, gráfica inferior), obtenido como la media armónica entre la precisión y la cobertura,

$$F1=2\cdot(precisión\cdot cobertura)/(precisión + cobertura)\ ,$$

es estable y cercano al 70% durante gran parte del proceso. Solo el día 22 el desempeño del clasificador cae drásticamente.

La razón de esta caída podría estar en la entrada de (muchas) nuevas palabras neutrales y negativas el día 23 (ver gráfica media superior donde la clasificación se hace con los datos del día anterior). Aquellas nuevas palabras pasan a formar parte del *pool* de entrenamiento y es por eso que el sistema tardaría en 'aprender' el nuevo vocabulario y sus relaciones (región demarcada en azul en los gráficos). Una vez finalizado este proceso de aprendizaje, el clasificador volvería a estabilizarse con valores de exactitud y cobertura más parejos entre categorías. Es interesante notar el porcentaje de clasificación se estabiliza en valores altos tempranamente, aunque esto no se traduce siempre en un buen desempeño.

La hipótesis anterior, referida al proceso de aprendizaje, podría ser válida para este caso, sin embargo, un segundo análisis, muestra que la razón de las caídas/subidas del clasificador podrían ser otras, vinculadas a la actividad propia de *Twitter*. En el Gráfico 2 se presenta la dinámica del desempeño para otro tema de acuerdo a la cantidad de *tweets* evaluados y la cantidad de palabras nuevas que estos aportan al entrenamiento. Es posible observar que una mayor cantidad de *tweets* de entrada diaria (gráfica inferior) se traduce, evidentemente, en un número mayor de palabras nuevas (gráfica media), aunque no se distribuyen uniformemente entre categorías. Entre los días 15 y 18 un gran volumen de mensajes entra al sistema, pero estos parecen ser en su gran mayoría contenedores de palabras negativas. Lo interesante es que estos aumentos en el tamaño del entrenamiento no siempre se traducen en caídas en el desempeño. De hecho, parece ser que la dinámica del entrenamiento no tiene una correlación inmediata con éste.

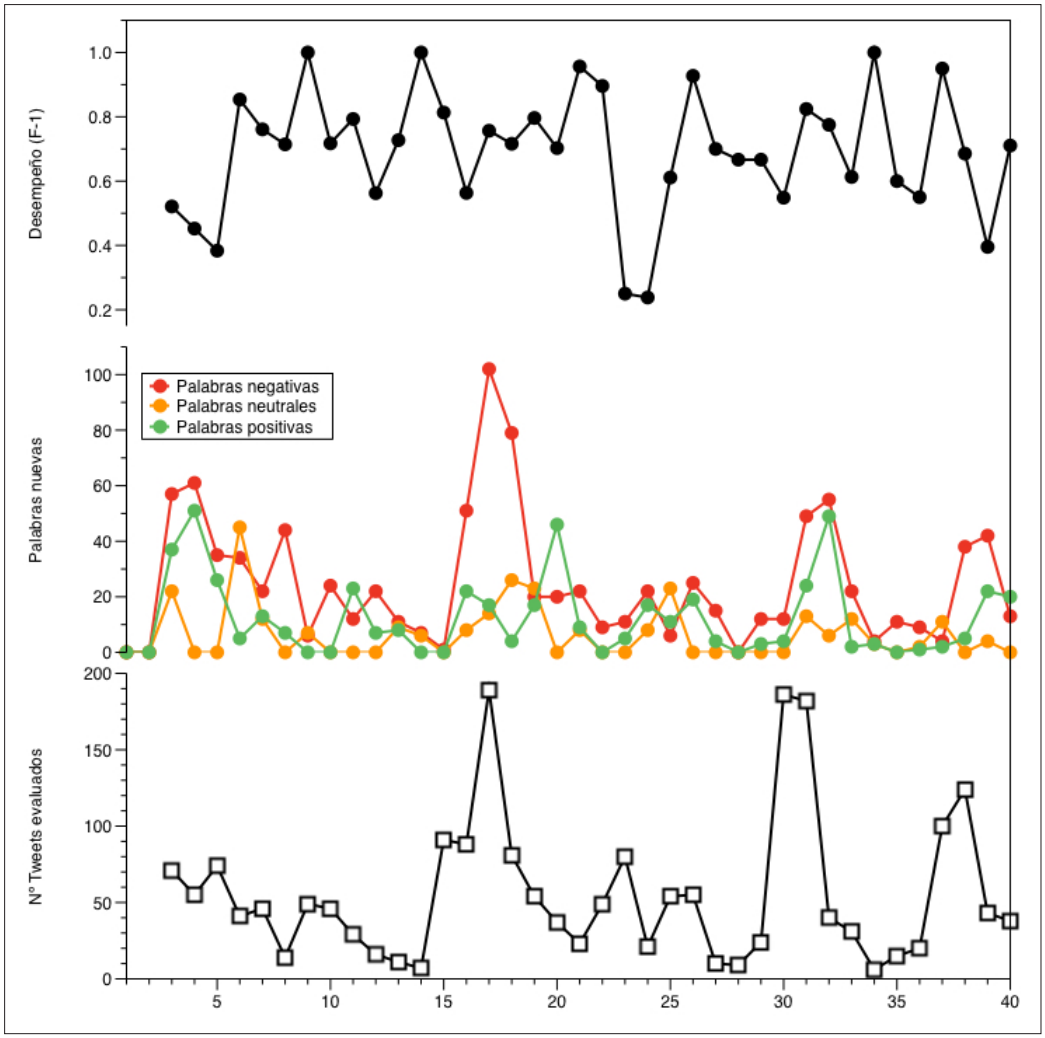

Gráfico 2. Desempeño (Micro F1) del clasificador para otro tema (gráfica superior) en función del número de *tweets* evaluados (gráfica inferior) y cantidad de palabras nuevas incorporadas en cada categoría (gráfica media).

Una de las funcionalidades más utilizadas en la red social *Twitter* es el redirigir mensajes emitidos por otros usuarios. Estos mensajes se denominan *retweets* (RTs) y, en términos de contenido, son copias exactas del mensaje original. Del corpus total de mensajes de esta red social, compuesto por 170 billones de mensajes, cerca de un 24% corresponden a RTs (Leetaru et al., 2013), y en días de actividad particular este porcentaje puede ser aún mayor. Sería la presencia de un gran número de estos mensajes (similares) en ciertos días los que 'amplificarían' los errores/aciertos de clasificador. Si el clasificador falla/acierta en la clasificación de uno de estos mensajes

en un día, se podría esperar que en promedio cerca del 24% de estos esté mal/bien clasificado.

Una forma de eliminar el problema de los RTs es evaluar el desempeño del clasificador en conjuntos de *tweets* escogidos al azar. Aunque es de esperar que la proporción de tipos de mensajes se mantenga, la probabilidad de que aparezcan mensajes iguales (RTs de un mismo día) disminuye. La Tabla 1 muestra los resultados para este análisis en los mismos dos temas de los gráficos 1 y 2. Para cada tema se realizaron una serie de repeticiones en las cuales del conjunto total de *tweets* clasificados por un humano, un 70% se utilizó como entrenamiento y el 30% restante como conjunto de evaluación.

Los resultados muestran que el desempeño del clasificador es superior al escenario 'real' de clasificación de los gráficos 1 y 2. Para dos temas totalmente distintos, que no comparten *tweets*, los resultados de desempeño son prácticamente iguales, a pesar de tener un nivel distinto de entrenamiento, sugiriendo que las redes de palabras utilizadas en estos experimentos tienen un desempeño para clasificar mensajes de *Twitter* cercano al 80%. Analizando el desempeño en cada categoría, se aprecian otras similitudes. En ambos casos, son los mensajes positivos aquellos mejor detectados por el algoritmo, seguidos por los negativos y finalmente por los neutrales.

Tabla 1. Desempeño de tres clasificadores automáticos: redes de palabras, Máquina de Soporte Vectorial (SVM-SMO) y *Naïve Bayes* en la clasificación de *tweets* en un escenario aleatorio. Promedios obtenidos de la evaluación de 1543 *tweets* en 5 repeticiones para el Tema 1 y 453 *tweets* evaluados en 3 repeticiones para el Tema 2. El desempeño se muestra como F1 Macro y Micro, y como medidas de Precisión (P) y Cobertura (C) para cada categoría en relación tamaño de entrenamiento medido como nodos (N) y enlaces (E) para las redes de palabras.

Tema	Micro F1	Macro F1	P / C positivos	P / C negativos	P / C neutrales	N / E positivos (Miles)	N / E negativos (Miles)	N / E neutrales (Miles)
Redes de Palabras								
1	0.80 ± 0.01	0.78 ± 0.00	0.84 / 0.88	0.84 / 0.67	0.71 / 0.72	3.3 / 8.1	1.7 / 4.2	3.2 / 8.5
2	0.82 ± 0.00	0.79 ± 0.00	0.84 / 0.91	0.80 / 0.71	0.79 / 0.69	1.3 / 3.3	0.9 / 2.0	0.7 / 1.4
SVM								
1	0.84 ± 0.00	0.82 ± 0.01	0.89 / 0.89	0.88 / 0.76	0.73 / 0.78	-	-	-
2	0.84 ± 0.01	0.80 ± 0.01	0.84 / 0.93	0.86 / 0.71	0.78 / 0.70	-	-	-
Naïve Bayes								
1	0.71 ± 0.02	0.72 ± 0.02	0.86/ 0.77	0.65 / 0.68	0.71 / 0.67	-	-	-
2	0.67 ± 0.01	0.70 ± 0.01	0.94 / 0.65	0.66 / 0.71	0.43 / 0.85	-	-	-

A pesar que es necesario remarcar que esta situación aleatoria no es del todo 'real', por la alteración de la representatividad diaria de ciertos *tweets*, es un escenario que permite establecer el desempeño del clasificador propuesto y compararlo con aquellos tradicionalmente utilizados en clasificación supervisada automática de textos. En la

misma tabla se muestra el desempeño de una Máquina de Soporte Vectorial y un algoritmo tipo *Naïve Bayes* en la resolución de los mismos problemas de clasificación (mismos *tweets* para entrenamiento y evaluación). Para este análisis se utilizó el algoritmo SVM que usa la Optimización Secuencial Mínima (SMO) (para mayores detalles ver Platt, 1998; Keerthi, Shevade, Bhattacharyya & Murthy, 2001). Para implementar la clasificación con estos algoritmos se utilizó el *software* para minería de datos WEKA (2013).

Al igual que en las redes de palabras, para los clasificadores SVM y *Naïve Bayes* son los mensajes positivos los mejor detectados, luego los negativos, y finalmente los neutrales. Esta similitud entre clasificadores sugiere que es una propiedad independiente del clasificador y reflejaría particularidades de los temas a evaluados. No obstante la exactitud y cobertura con la que los algoritmos detectan las categorías difiere bastante en algunos casos.

Los resultados sitúan al clasificador *Naïve Bayes* como el con peor desempeño. En el análisis, su desempeño fue aproximadamente un 10% inferior a SVM, diferencia mayor a la obtenida en otros experimentos (Harish et al., 2010) y que pondría en evidencia que para la clasificación de este tipo de documentos digitales se requiere de algoritmos más sofisticados, dado las particularidades de los mensajes. Esto explicaría el por qué el desempeño de la SVM es también inferior al obtenido en la clasificación de textos noticiosos donde alcanzan en promedio un desempeño cercano al 88% (Harish et al., 2010). De hecho, clasificando mensajes de *Twitter*, el desempeño de la SVM disminuye cerca de 8%, en el caso de la Micro F1, siendo ligeramente superior al obtenido por las redes de palabras propuestas en este trabajo. Para la otra media armónica, la Macro F1, los desempeños entre ambos algoritmos son estadísticamente similares en uno de los temas.

CONCLUSIONES

En este artículo se describe y evalúa un algoritmo para clasificar textos automáticamente, basado en una representación y clasificación distinta a la utilizada tradicionalmente por algoritmos de clasificación supervisada. Para representar el texto a clasificar, el método propuesto considera las relaciones de cercanía de las palabras que lo componen transformándolo en una red de palabras co-ocurrentes. Dicha representación opera también como un objeto clasificador, de forma que la categoría de un texto sin clasificar dependerá de la probabilidad de reconstruir su representación como red de palabras en otra red de palabras construida por entrenamiento humano.

Los resultados muestran que el clasificador presenta niveles de acierto buenos en la clasificación de mensajes de *Twitter*, cercanos al 80% respecto a la clasificación realizada por una persona. Estos niveles de desempeño son superiores en cerca de 10% a los obtenidos con el tradicional algoritmo de clasificación de textos *Naïve Bayes*, sugiriendo que la 'ingenuidad' de este método afecta negativamente su desempeño

para este tipo de datos. En comparación a algoritmos más sofisticados, como las Máquinas de Soporte Vectorial, las redes de palabras presentan un desempeño prácticamente similar en la clasificación de este tipo de mensajes digitales.

El trabajo aquí presentado muestra además la factibilidad para la implementación del clasificador en un sistema automático de clasificación diario para mensajes de *Twitter*, con un desempeño general promedio cercano al 70% respecto a la clasificación realizada por personas una vez alcanzado un nivel de entrenamiento satisfactorio. La diferencia respecto al desempeño de 80% comentado en el párrafo anterior tendría que ver con la clasificación errónea de mensajes similares, como los *retweets*, sobrerrepresentados respecto a los mensajes distintos en días particulares.

Uno de los aspectos a destacar del método propuesto es que se inserta en la línea de algoritmos de la inteligencia artificial basada en una computación descentralizada. Esta aproximación es la que Turing (1992) como uno de los paradigmas para el desarrollo de máquinas inteligentes que emulan el comportamiento de los procesos cognitivos en seres vivos. En el algoritmo propuesto operan mecanismos tipo abajo-arriba (*bottom-up*) para la clasificación siendo esta dependiente de detalles de bajo nivel tales como las relaciones entre palabras. De esta forma la clasificación no es un fenómeno que se 'enseña' a alto nivel sino que emerge de la abstracción topológica de la gramática que significa la red de palabras. Por esta razón, el algoritmo que se propone en este trabajo tiene la potencialidad de operar en distintos contextos (temas, idiomas, etc.). Además, debido a que la representación matricial del grafo es una abstracción relacional para cualquier tipo de entidades, el algoritmo puede ser implementado fácilmente en clasificaciones de otro tipo y no solo de textos.

Si bien las evaluaciones preliminares del método sugieren que este es una buena alternativa a los algoritmos tradicionalmente utilizados en clasificación automática de textos, aún es necesaria una mayor exploración del mismo. Mediante algoritmos clásicos de optimización, debería ser posible optimizar la ecuación de costo y encontrar los parámetros α y β indicados para cada tema en particular. Además, utilizando otras definiciones de representación en grafos y la determinación de los parámetros topológicos de G_i que signifiquen un entrenamiento suficiente para una determinada categoría, el desempeño del clasificador propuesto debería mejorar.

REFERENCIAS BIBLIOGRÁFICAS

Baeza-Yates, R. & Ribeiro-Neto, B. (1996). *Modern information retrieval: The concepts and technology behind search.* Reading, M.A.: Addison-Wesley.

Bryden, J., Funk, S. & Jansen, V. (2013). Word usage mirrors community structure in the online social network Twitter. *EPJ Data Science, 2*(1), 3.

Cárdenas, J. P., Losada, J. C., Moreira, A., Torre, I. G. & Benito, R. M. (2011). Topological complexity in natural and formal languages. *Int. J. Complex Systems in Science, 1*(2), 221-225.

Caruana, R. & Niculescu-mizil, A. (2006). *An empirical comparison of supervised learning algorithms.* Ponencia presentada en el International Conference on Machine learning, Pittsburgh, Estados Unidos.

Choudhary, B. & Bhattacharyya, P. (2003). *Text clustering using universal networking language representation.* Ponencia presentada en el 11th International World Wide Web Conference, Honolulu, Hawaii, Estados Unidos.

Harish, B. S., Guru, D. S. & Manjunath, S. (2010). Representation and classification of text documents: A brief review. *IJCA Special Issue on Recent Trends in Image Processing and Pattern Recognition*, 2,110-119.

Hotho, A., Maedche, A. & Staab, S. (2001). *Ontology-based text clustering.* Ponencia presentada en el International Joint Conference on Artificial Intelligence, Seattle, Estados Unidos.

Jin, W. & Srihari, R. K. (2007). *Graph-based text representation and knowledge discovery.* Ponencia presentada en ACM symposium on Applied computing, Nueva York, Estados Unidos.

Joachims, T. (2002). *Learning to classify text using Support Vector Machines.* Dordrecht: Kluwer Academic Publishers.

John, G. & Langley, P. (1995). *Estimating continuous distributions in bayesian classifiers.* Ponencia presentada en el 11th Conference on Uncertainty in Artificial Intelligence, Montreal, Canadá.

Keerthi, S., Shevade, S., Bhattacharyya, C. & Murthy, K. (2001). Improvements to Platt's SMO Algorithm for SVM Classifier Design. *Neural Computation, 13*(3), 637-649.

Lan, M., Tan, C. L., Su, J. & Lu, Y. (2009). Supervised and traditional term weighting methods for automatic text categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 31*(4), 721-735.

Leetaru, K., Wang, S., Cao, G., Padmanabhan, A. & Shook, E. (2013). *Mapping the global Twitter heartbeat: The geography of Twitter* [en línea]. Disponible en: http://firstmonday.org/ojs/index.php/fm/article/view/4366/3654

Lewis, D.D. (1998). *Naive (Bayes) at forty: The independence assumption in information retrieval*. Ponencia presentada en European Conference on Machine Learning, Chemnitz, Alemania.

Manley, E. (2012). *Urban movements. Flows, behaviour and networks in the City* [en línea]. Disponible en: http://urbanmovements.posterous.com/detecting-languages-in-londons-twittersphere

Mitchell, T. (1997). *Machine learning*. Nueva York: WCB/Mcgraw-Hill.

Namata, G. M. & Getoor, L. (2009). *A pipeline approach to graph identification*. Ponencia presentada en el International Workshop on Mining and Learning with Graphs, Leuven, Bélgica.

Platt, J. (1998). *Fast training of Support Vector Machines using sequential minimal optimization. Advances in Kernel Methods - Support Vector Learning*. Cambridge: MIT Press.

RAE (2013). *Corpus de Referencia del Español Actual (CREA)* [en línea]. Disponible en: http://corpus.rae.es/lfrecuencias.html

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys, 34*(1), 1-47.

Sen, P., Namata, G., Bilgic, M., Getoor, L., Gallagher, B. & Eliassi-Rad, T. (2008). Collective classification in network data. *AI Magazine, 29*(3), 93-106.

Turing, A. (1992). *Intelligent machinery. Collected Works of A. M. Turing: Mechanical Intelligence*. Amsterdam: Elsevier Science Publishers.

Uchida, H., Zhu, M. & Della Senta, T. (1995). *UNL: A gift for a millennium*. Tokyo: UNU/IAS.

Vapnik, V. N. (1989). *Statistical learning theory*. Nueva York: Wiley-Interscience.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. Berlin: Springer-Verlag.

WEKA. (2013). Weka 3: *Data Mining Software in Java* [en línea]. Disponible en: http://www.cs.waikato.ac.nz/ml/weka/.

Zhang, H. (2004). *The optimality of Naive Bayes* [en línea]. Disponible en: http://www.citeulike.org/user/JoSeK/article/370404

Zhou F., Zhang, F. & Yang, B. (2010). Graph-based text representation model and its realization. Ponencia presentada en el *International Conference on Natural Language Processing and Knowledge Engineering,* Beijing, China.

Zipf, G. L. (1965). *Human behavior and the principle of least effort*. Reading, M.A.: Addison-Wesley.

NOTAS

1 El concepto de reconstrucción es similar al mapeo entre grafos propuesto por Sen, Namata, Bilgic, Getoor, Gallagher y Eliassi-Rad (2008) y Namata y Getoor (2009) que busca correspondencia entre un par de estos respecto a sus nodos y sus enlaces, esto con la finalidad de identificar una red particular a partir de otra que la contiene pero con mucho ruido (i.e., con información inútil para la red particular a detectar).

2 Para este cálculo existen dos alternativas, promediar los valores de 'precisión' y 'cobertura' obtenidos localmente en cada categoría (Macro-F1) u obtener los valores de 'precisión' y 'cobertura' sumando todas las decisiones individuales de cada categoría (Micro-F1). Según Sebastiani (2002) dichos cálculos pueden resultar diferentes, en el caso en que las categorías estén representadas de forma desuniforme en la muestra total a clasificar.

* AGRADECIMIENTOS

Este trabajo contó con el financiamiento de los proyectos FONDECYT Nº 11121292, *"Microblogs Texts Classification Using Word Networks"*, CONICYT Nº 78110202, "Análisis de redes sociales y textos digitales de la Internet mediante técnicas basadas en Grafos" y CONICYT SOC-1101, "Anillo en Complejidad Social".