PAIDEIA XXI

Vol. 4, Nº 5, Lima, agosto 2014, pp. 113-123

CATEGORIZACIÓN DE TEXTOS UTILIZANDO ANÁLISIS STRINGKERNEL E INDEXACIÓN SEMÁNTICA LATENTE: APLICACIÓN DE TEXTOS DE PROPIEDADES MEDICINALES DE PLANTAS

Augusto Cortez Vásquez

Resumen

Debido a la abundante información existente se hace necesario organizar, mantener y procesar toda información disponible a partir de un conocimiento más profundo del lenguaje. Un clasificador de textos (CT) consiste en etiquetar un texto o documento con una o varias categorías temáticas predefinidas. El enfoque de clasificación considera que dado un conjunto de documentos D y un conjunto de categorías C, encontrar una función haga corresponder a un documento *d* tomado de D, una categoría determinada *c en C.* Para ello realiza un análisis léxico que identifique las subsecuencias de lexemas de d; luego, mediante un análisis stringkernel encuentre el grado de similitud entre dos textos. Dos textos son más similares mientras tengan más subsecuencias en común.

Palabras clave: Categorización de textos, clasificación de textos, análisis lexicográfico.

Abstract

Due to the abundant information is necessary to organize, maintain and process all information available from a deeper understanding of language. A text classifier (CT) consists of a text label or document to one or more predefined subject categories. The classification approach considers that given a set of documents D and a set of categories C, find a function then correspond to a document d taken from D, a category c in C. It performs a lexical analysis to identify the subsequences lexemes of d, and then by analyzing string kernel find

the degree of similarity between two texts. Two texts are more similar while you have more in common subsequences.

Keywords: text categorization, text classification, lexical analysis.

INTRODUCCIÓN

Lo que la ciencia y la tecnología ha conseguido hasta ahora ha sido realmente espectacular. Solo tenemos que mirar a nuestro alrededor para atestiguar lo que el extraordinario poder de nuestra comprensión de la naturaleza y del conocimiento nos ha ayudado a obtener. La categorización de textos consiste en etiquetar un texto o documento con una o varias categorías temáticas preestablecidas [4,7]. El volumen de información en las diferentes áreas de conocimiento crece en un grado exponencial, de esto se deriva que su tratamiento así como su almacenamiento sea más complejo. A causa de ello, se ha hecho necesario desarrollar nuevos instrumentos y herramientas que faciliten la realización de procesos de búsqueda de forma eficiente y efectiva, así como la administración de estos recursos. Para categorizar textos se establece el grado de similitud entre un texto y una clase de textos, para ello se utiliza el método de StringKernel (SK). El método SK establece que dos textos son similares mientras tengan más subsecuenticas en común. En [4,Cortez] 2013, se presentó un estudio de Categorización de textos mediante Máquinas de Soporte Vectorial (MSV), en la que se utilizó la técnica StringKernel (SK) para detectar similitud de textos. La técnica indexación semántica latente utiliza un método numérico llamado *descomposición en valores singulares,* que permite identificar patrones en las relaciones entre los términos contenidos en una colección de textos no estructurados [Hernández, 2009]. El principio de ISL es que muchas palabras utilizadas en textos pueden tener significados similares. El presente trabajo tiene como objetivo construir un modelo que permita etiquetar un texto con una o varias categorías temáticas predefinidas. Para ello se construyó un conjunto de clases de textos, se definió una función que asigne a cada documento una clase, se eligió una técnica para representación de documentos y se construyó un prototipo que implemente la especificación.

MARCO TEÓRICO

Clasificación de textos

La clasificación de textos es un tema que forma parte de Recuperación de Información RI (*Information Retrieval*) y se enmarca dentro de la disciplina de lenguaje de procesamiento natural. Tiene como propósito etiquetar, es decir, asignar etiquetas que indican a qué categoría o categorías corresponde el documento. La mayor parte de los autores, entre ellos Hernández, clasifican la categorización de texto como un cruce entre Máquinas de Aprendizaje (*Machine Learning* - ML) y RI. Varios investigadores en el área

MSV se refieren a esta área de estudio como una instancia de la Minería de Textos (*Text Mining* - TM) [6,8]. En el contexto del presente trabajo, definiremos clasificar como distinguir las características propias de un objeto y establecer las diferencias con otros objetos. En este contexto, clasificar textos significa relacionar un texto con clases [Russell, 2003; Palma, 2008].

Estrategias de clasificación

Existen dos estrategias para la clasificación de textos: el primero consiste en incorporar información semántica a la representación de textos. Es conveniente destacar que, en general, estos estudios están enfocados en documentos donde es factible, en la mayoría de los casos, disponer de una colección de entrenamiento para la tarea de desambiguación del sentido de las palabras (*WSD* las siglas en inglés para *Word Sense Disambiguation*). La segunda estrategia consiste en el uso de métodos de WSD basados en conocimiento que obtienen información desde recursos léxicos externos. Estudios realizados muestran que si bien este tipo de métodos suelen mostrar resultados de menor calidad que los obtenidos con métodos basados en corpus, constituyen en muchos casos la única alternativa realista, si se desea hacer uso de información semántica en la representación de documentos [14].

Indexación semántica latente (ISL):

Para comprender más claramente la similitud de dos textos, se utiliza un método numérico llamado descomposición en valores singulares (SVD por sus siglas en inglés), cuya función primordial es identificar patrones en las relaciones entre los términos contenidos en una colección de textos no estructurados. El principio de ISL es que muchas palabras utilizadas en textos pueden tener significados similares. La idea principal es emparejar por conceptos en lugar de por términos, o sea, un documento podría ser recuperado si comparte conceptos con otro que es relevante para la consulta dada.

MÉTODO Y TÉCNICAS UTILIZADAS

Metodología

Para modelar el problema P de clasificación de textos, se seguirá la siguiente metodología:

1. Definición del dominio de todos los documentos (D) y el dominio de todas las clases predefinidas (C).
2. Construcción de un analizador léxico que detecte los lexemas y las subsecuencias que componen al texto.
3. Para cada clase aprenderemos una función que decidirá si cada documento d pertenece o no a la clase asociada.

El objetivo es aprender una función:

$\emptyset : D \rightarrow C$, tal que $\emptyset (d_i)=c_i$;

d_i es un documento cualquiera y c_i es el vector de las categorías a las que pertenece el documento d_i.$\emptyset (d_i) \subseteq C \rightarrow$

4. Para clasificar un documento d en D encontraremos el grado de similitud entre dos textos. Para esto utilizaremos la técnica de StringKernel

5. Construcción de un prototipo para hallar la similitud de un texto con alguna clase ya creada. Cuando se ingresa un texto se compara con cada una de los textos de cada clase. El nuevo texto se clasificará con la clase que tenga mayor similitud. Si no existe similitud, se creará una nueva clase para el texto analizado. Se utilizará la técnica de indexación semántica latente, que permita recuperar textos a partir de conceptos y no solo de términos

Técnica utilizada

StringKernel: Cuando se quiere clasificar un documentos d_i, se compara cada subcadena del documento con un referente. Cuanto más subcadenas tengan en común dos subcadenas más similares se consideran. Cada categoría corresponde a una clase que tiene un referente c_i.un documento d_i será similar a una clase c_i si tiene más subcadenas en común.

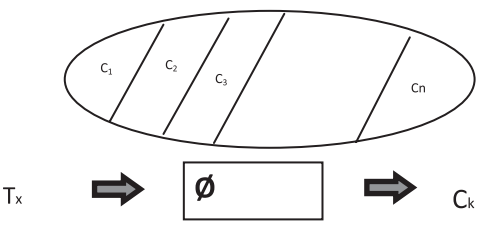


Figura 1. Clases de textos

Ø verifica a qué clase pertenece el texto Tx

Especificación formal

Precondición: el texto Tx se encuentra depurado.

FunCategoriza(Tx: secuencia de texto) dev (C_k: clase)

Postcondición: Existe K tal que $T_k$esta en C_k y similar(Tx,Tk)

Tx debe encontrarse limpio, es decir se han abstraído los lexemas irrelevantes

Función de similitud

Similar (T_x,T_k) establece que para un g: grado de similitud, el número de subsecuencias en común entre Tx y Tk es mayor o igual a g

Ø:D →F,

La función Ø transforma un ejemplo n-dimensional en un vector decaracterística N-dimensional

Ø (d) = ($Ø_1$(d), $Ø_2$(d) …, $Ø_n$ (d)) = ($Ø_i$ (d)) para todo i =1,…, N

Kernel para secuencia de textos

Un núcleo para secuencias de texto de dos documentos de texto, permite comparar los textos por medio de las subcadenas que contienen: las subcadenas más en común, las más similares [9]. Un aspecto importante es que dichas subcadenas no necesitan ser contiguas, y el grado de contigüidad de una subcadena en un documento determina cuál será el peso que se le asignara en la comparación. Por ejemplo: la subcadena ‹r-a› está presente tanto en la palabra «área» y en la palabra pera, pero con diferente ponderación. Con el fin de hacer frente a subcadenas no contiguas, es necesario introducir un factor de decaimiento $\gamma \in (0, 1)$ que se puede utilizar como ponderar la presencia de una determinada característica en un texto.

Definición: Sean Σ un alfabeto finito, Σ^n el conjunto de todas las ca-

denas de longitud *n*, y el conjunto de todas las cadenas finitas. La longitud de una cadena $s \in \Sigma^*$ es $|s|$, y sus elementos son $s(1)$, $s(2)$,..., $s(|s|)$; la concatenación de dos cadenas s y $t \in \Sigma^*$ se escribe st.[9].

$$\Sigma^* = \bigcup_{\infty} \Sigma^n$$

Dado un índice de secuencia $i = (i_1, i_2, \ldots, i_{|u|})$ con $1 \leq i_1 < \ldots < i_{|u|} \leq |s|$
La subsecuencia se define como $u = s(i) = s(i_1),\ldots, s(i_{|u|})$.
La longitud de la subsecuencia en s se define como $i_{|u|} - i_1 + 1$, si i no es contigua entonces l(i) es mayor que la longitud de u ($|u|$).
El espacio de rasgos generado a partir de cadenas de longitud n se define como $H_n = R_{(\Sigma^*)}$, esto significa que dicho espacio tiene una dimensión o coordenada por cada uno de los elementos de Σ^n. La proyección de todas las coordenadas en el espacio de rasgos para cada subsecuencia $u \in \Sigma^n$ se describe como $[\emptyset_n(s)]_u = \Sigma \gamma^{l(i)}$

Ejemplo

Considere las palabras ***cima, imán y tina***. Si tenemos en cuenta solo k =2, obtenemos un espacio de características 13-dimensional, donde las palabras se asignan de la siguiente manera:

	Ci	cm	ca	im	ia	ma	in	mn	an	ti	tn	ta	na
cima	γ^2	γ^3	γ^4	γ^2	γ^3	γ^2	0	0	0	0	0	0	0
imán	0	0	0	γ^2	γ^3	γ^2	γ^4	γ^3	γ^2	0	0	0	0
tina	0	0	0	0	γ^3	0	γ^2	0	0	γ^2	γ^3	γ^4	γ^2

Lo sorprendente de este ejemplo es que se identifican las subsecuencias en un espacio de características amplio.

Diseño del clasificador de texto basado en StringKernel:

Kernel utilizado

Se utilizara un Kernel SSK (Kernel de subsecuencia de cadenas).

Sea un alfabeto Σ, entonces definimos

$$\Sigma^* = \bigcup_{\infty} \Sigma^n$$

$\Sigma^0 = \{ \lambda \}$ define el conjunto de cadenas de longitud 0
$\lambda =$" " es la única cadena de longitud cero
Σ^1 define el conjunto de cadenas de longitud 1
Σ^2 define el conjunto de cadenas de longitud 2
En general Σ^n define el conjunto de cadenas de longitud n
Cada cadena $w \in \Sigma^n$ le corresponde Ø (w), donde los elementos de w son w_1 w_2 w_3... w_n
Si s, $t \in \Sigma^n$, entonces s.t $\in \Sigma^n$

Prototipo para clasificar un documento en función a su similitud con los textos de las clases ya creadas.

Se definió 15 clases de propiedades de plantas medicinales, como se muestra en la siguiente figura:

Clase	Representante de clase
C1	hepática: que afecta al hígado; que contribuye a curar enfermedades del hígado
C2	antiséptica: que se opone a la putrefacción
C3	astringente: que contrae o detiene hemorragias
C4	Mucilaginosa: que alivia las partes inflamadas
C5	Carminativo:ayuda a expulsar la ventosidad de las entrañas
C6	Aromatica:estimulante, sabrosa
C7	Diurética: que incrementa la secreción y el flujo de orina
C8	Febrífuga: que rebaja y elimina la fiebre
C9	Litotriptica: que disuelve los cálculos de los órganos urinarios
C10	Antihelmíntico: que expulsa las lombrices
C11	Expectorante: que facilita la expectoración
C12	Antiespasmodica: que alivia y previene los espasmos
C13	Rubefaciente: que estimula la circulación, provocando la rojez de la piel
C14	Sedante: que calma los nervios
C15	Emetica: provoca vómitos

Figura 2. Muestra de textos de propiedades medicinales

Precondiciones: Se eliminaron los caracteres que no ofrecen ningún tipo de información y aumentan la dimensión del espacio de rasgos:

- dos puntos (:),
- coma (,),
- punto y coma (;),
- punto (.),
- comillas simples (‘) y dobles (“),
- guión (-)

No se han considerado las tildes, por lo que se sustituyeron las vocales acentuadas por vocales no acentuadas, con la finalidad de reducir la dimensión del espacio de rasgos.

Se considerará un conjunto de palabras relevantes:

```
FuncionClasificar()
    Inicio
        Leer T              //Lee Texto
        V=Vectorizar(T) // explora el texto T y almacena en un
        vector V todas las subsecuencias de T
        K = max{            Similar(V , Ci) para todo i:1..N}  //
                            devuelve la clase de mayor similitud
        si K < grado
                crear nueva clase C(N+1)
        Sino
                clasificar T en clase K
        FinSi
Fin
```

```
    FuncionVectorizar(T)
    Inicio
       Lexema =Lexico(T)  // retorna el siguiente lexema
        i=0
        Mientras(existan lexemas)
                Si ¬ desechable(lexema)
                        i=i+1
                        V[i]=lexema
                Finsi
                Lexema =Lexico(T)  // retorna el siguiete lexema
        FinMientras
        Retornar V
    Fin
```

```
    Funcion Similar(X,Y)
    Inicio
        contador = 0
        Para cada secuencia w de X
                Si w es secuencia de Y
                        contador = contador +1
                FinSi
                // realiza un análisis de indexación semántica latente
                Exploracion_Semanticalatente(w,Y)
        FinPara
        Retornar contador
Fin
```

Sea el alfabeto $\sum$ y el texto

$T_x = a_1\ a_2 \ldots a_n \quad T_x \in \sum^*$

Sea la secuencia $w = a_1\ a_2 \ldots a_k$ subsecuencia de T_x $w \in T_x^*$

Sea $X = \{\alpha$ / Si E $w \in \sum^*$ y $w \in T_x \alpha = w\}$ clase de texto

$| |X = n$ cardinalidad de la clase X

Sea $\quad \emptyset : T \rightarrow \sum$

$T_x \quad \sum_i \rightarrow$

Para todo texto T_x , Ø le hace corresponder la clase $\sum_i$

Sea grad(u,v) = g_{uv} grado de similitud de T_x y T_x

Si $T_u \in T$ elegir v tal que g_{uv} = *max { g_{ui} para todo i: 1..n}*

Cuando se vectoriza el texto, se consideran solo los lexemas significativos. Para ello se creó el archivo de lexemas no relevantes (poco significativos): *el, las, en, que, por..* etc. Cuando se explora el texto, por cada lexema se busca en el archivo no relevantes, si se encuentra, se desecha el lexema, en otro caso se almacena en el vector. Esto reducirá la complejidad espacial del algoritmo. La indexación semántica latente permite que la búsqueda no sea solo por términos sino por conceptos.

TX= pelotas de futbol, se asociará con textos que incluyan balones de fútbol o bolas de futbol.

Concepto : Pelota	Textos
Conceptos relacionados: Balón bola	Los muchachos lanzaron la pelota. Los muchachos lanzaron el balón. Los muchachos lanzaron la bola

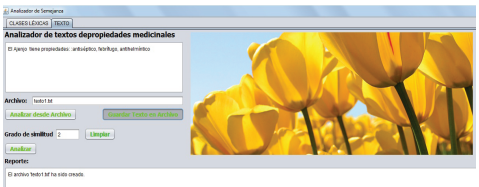
Concepto : muchacho	Textos
Conceptos relacionados: chiquillo joven	El muchacho es muy inteligente El joven es muy inteligente. El chiquillo es muy inteligente

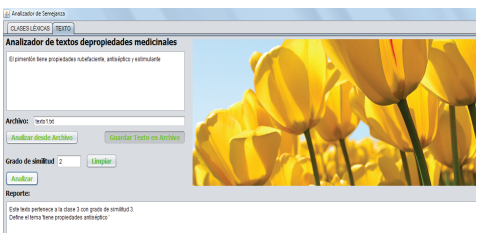
El siguiente prototipo ingresa un texto, halla el grado de similitud con las clases existentes:

Clase	Textos de la misma clase
C1	Ajenjo: antiséptico, febrífugo, antihelmíntico El diente de león tiene propiedades hepáticas y es diurético
C2	Ajenjo: antiséptico, febrífugo, antihelmintico Ajo: diaforético,diurético, expectorante, antiséptico, antihelmíntico Manzanilla: antiséptica, antiespasmódica, emética, aromatica El pimentón tiene propiedades rubefaciente, antiséptico y estimulante
C3	Fresa: diurético, astringente, laxante La verbena es astringente y expectorante
C4	Cataplasma de semillas molidas de fenogreco(Trigonellafoenumgraecum) tiene efecto mucilaginoso, especialmente contra los abscesos, ulceras e hinchazones. Una cataplasma de hojas de Malva mojadas en agua caliente es bueno para las llagas e inflamaciones
C5	La infusión de salvia tiene efecto carminativo La valeriana previene la aparición de gases y piedras en la vejiga de la orina
C6	La manzanilla es recomendable para la dispepsia, estomago débil y desordenes o trastornos nerviosos La hierbabuena es aromática y se recomienda para cólicos, gases, dispepsia, espasmos y náuseas
C7	Ajo: diaforético, diurético, expectorante, antiséptico, antihelmíntico Fresa: diurético, astringente, laxante El perejil es diurético y expectorante El diente de león tiene propiedades hepáticas y es diurético
C8	Ajenjo: antiséptico, febrífugo, antihelmíntico
C9	La infusión de hojas de chancapiedra alivia los cálculos renales El jugo de piña con cucharada de aceite de oliva expulsa los cálculos
C10	Ajo: diaforético, diurético, expectorante, antiséptico, antihelmíntico
C11	Ajo: diaforético, diurético, expectorante, antiséptico, antihelmíntico El perejil es diurético y expectorante La verbena es astringente y expectorante
C12	Manzanilla: antiséptica, antiespasmódica, emética, aromática La valeriana tiene propiedades antiespasmódica
C13	El pimentón tiene propiedades rubefaciente, antiséptico y estimulante El jengibre es rubefaciente
C14	
C15	La manzanilla tiene propiedad estimulante y a la vez emetica

Figura 3. Tabla de textos analizados con sus respectivas categorías

Se ingresa el texto1: Ajenjo: antiséptico, febrífugo, antihelmíntico





CONCLUSIONES

a) La clasificación de textos ha suscitado mucho interés en el área de Inteligencia Artificial.
b) Se consideró un analizador léxico que identifique los lexemas y las subsecuencias formadas por estos lexemas. La exploración se realizó de izquierda a derecha. Es necesario, sin embargo, puntualizar que la exploración de derecha a izquierda adicional incrementaría las subsecuencias en común; sin embargo, incrementaría el costo computacional
c) Para establecer la similitud de dos textos se utilizó la técnica StringKernel. Mientras más secuencias en común tengan dos textos , más similares son.
d) Se consideró subsecuencias de textos de lexemas contiguos en un orden de izquierda a derecha, y se eliminaron los lexemas irrelevantes que no añaden significado al texto.
e) La indexación semántica latente permitió que cuando se recupere un texto, la búsqueda se realice por conceptos y no solo por términos
f) Cuando se construye el kernel, se consideran dos alternativas: considerar subcadenas de caracteres contiguos, o no. Sin embargo, se debe tener en cuenta que permitir la no contigüidad eleva el costo computacional de la aplicación. Se realiza un análisis léxico izquierda a derecha. Un análisis que incluya la exploración derecha izquierda permitiría ampliar el número de subcadenas en común, pero el costo computacional se elevaría. Por ello deberá determinarse la alternativa en función de las prioridades y restricciones de la aplicación a desarrollar.

REFERENCIAS BIBLIOGRÁFICAS

Angulo, C. (2001). "Aprendizaje con máquinas núcleos en entornos de multiclasificacion". Tesis doctoral, Universidad Politécnica de Cataluña.

Ciapuscio, G. (1994). *Tipos textuales*. Buenos Aires: EUDEBA.

Comesaña (2010). "Modelos de regresión de máquinas de vectores soporte de mínimos cuadrados para la predicción de la cristalinidad de catalizadores de craqueo por espectroscopia infrarroja". Revista CENIC - Ciencias Químicas Redalyc. Centro Nacional de Investigación Científica. La Habana, ISSN 0254-0525.

Cortez Vásquez, Augusto. (2013). Revista RISI (Revista de Investigaciones de Sistema e Informática). Vol. 10 N°1. Lima. Pág.33

Gonzales, Abril. *Modelos de Clasificación basados en Maquinas de Vectores Soporte. Departamento de Economía Aplicada I*, Universidad de Sevilla.

Gutiérrez Alonso, Esther. (2007). "Aplicación de las máquinas de soporte vectorial para reconocimiento de matrículas". Proyecto de fin de carrera, Universidad Pontificia Comillas – Escuela Técnica superior de Ingeniería –Ingeniería Industrial (ICAI). Madrid.

Hernández, José. (2008). *Introducción a la minería de datos*. España: Pearson Prentice Hall.

Krikorian, Mauro. "Reconocimiento de dígitos manuscritos aplicando transformadas Wavelet sin submuestreo y Máquinas de Soporte Vectorial". Tesis de licenciatura. Departamento de Computación. Facultad de Ciencias Exactas y Naturales. Universidad de Buenos Aires.

Leija, Lorenzo. (2009). *Métodos de procesamiento avanzado e inteligencia artificial en sistemas sensores y biosensores*. México. Reverse Editores.

Lodhi, Huma. et al. (2002). *Text Clasification using String Kernels Royal Holloway*, University of London, Egham Surrey TW20 0EX.

Mendoza , M. (2011). "Categorización de texto en bases documentales a partir de modelos computacionales livianos". Revista Signos Versión. Valparaíso. ISSN 0718-934 Vol. 44 N° 77.

Muller, A.; Smola, J.; Ratsch, G., Scholkopf, B., Kohlnorgen, J. & Vapnik, V. (1997). "Predicting times series with support vector machine". Notas de trabajo.

Muñoz, P. (2006). "Sistema para el Reconocimiento Fuera de Línea de Caracteres Manuscritos". Grupo GAMA5 CEIFI, Universidad del Quindío.

Perea Ortega, José; Martín Valdivia, María Teresa; Montejo Raez, Arturo & Díaz Galiano, Manuel Carl . *Categorización de textos biomédicos usando UMLS*. ISSN 1135-5948

Palma, José (2008). *Inteligencia Artificial*. Madrid: Edit Mc Graw Hill.

Pedroza, Juan. (2007). *Aplicación de las máquinas de soporte vectorial al reconocimiento de hablantes*. Universidad Autónoma Metropolitana.

Rosas, Marta. (2010). "Un Análisis Comparativo de Estrategias para la Categorización Semántica de Textos Cortos". Revista Procesamiento del Lenguaje Natural N° 44, pp 11-18.

Russell, Stuar. (2003). *Inteligencia Artificial, Un enfoque moderno*. México: Edit Pearson.

Salazar Blandon, Diego Alejandro (2012). "Comparación de máquinas de soporte vectorial vs Regresión Logística ¿Cual es más recomendable para discriminar?". Tesis de grado de Magíster en Ciencias- Estadística. Universidad de Colombia. Facultad de Ciencias. Escuela de Estadística. Medellín.

Stitson, M.; Weston, J.; Gammerman, A.; Vovk, V. & Vapnik, V. (1996). "Theory of support vector machines. InformeTécnico". Bajado de http://svm.rst.gmd.de/, 1996.

Solera Ureña, Rubén (2011). "Máquinas de vectores de soporte para reconocimiento robusto del habla". Tesis doctoral. Dpto. de Teoría de la Señal y Comunicaciones. Universidad Carlos III de Madrid Leganez, Madrid.

Venegas, Rene. Clasificación de textos académicos en función de su contenido léxico-semántico Academic text classification base don lexical-semanticcontent. Pontificia Universidad Católica de Valparaíso, Chile.

Villasana, Sergio (2008). "Categorización de documentos usando máquinas de vectores de soporte". Revista Ingeniería Uc. Vol. 15, No 3, 45-52. ISSN (Versión impresa): 1316-6832. Universidad de Carabobo Venezuela.