RISI 10(1), 33 - 44 (2013)
REVISTA DE INVESTIGACIÓN DE SISTEMAS E INFORMÁTICA
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

ISSN 1815-0268 (versión impresa)
ISSN 1816-3823 (versión electrónica)

Categorización de Textos mediante Máquinas de Soporte Vectorial

Text Categorization using Support Vector machines

Augusto Cortez Vasquez, Luzmila Pro Concepción, Oswaldo Rojas Lazo, Roberto Calmet Agnelli

Universidad Nacional Mayor de San Marcos
Facultad de Ingenieria de Sistemas e Informatica

cortez_augusto@yahoo.fr, lproc2003@hotmail.com, orojasla@hotmail.com, robertocalmet@yahoo.com

RESUMEN

La categorización de textos es una aplicación que se enmarca dentro de la disciplina de lenguaje de procesamiento natural y está estrechamente relacionado con el concepto de clasificación. 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. Las máquinas de soporte vectorial (MSV) pertenecen a la familia de clasificadores lineales, y puede utilizarse para resolver el problema de la categorización de textos(CT) que consiste en etiquetar un texto o documento con una o varias categorías temáticas predefinidas. La razón por la cual se aborda el problema es su aplicación en diversos escenarios del área de recuperación de información (RI) tales como la organización automática de documentos, filtrado de documentos. El enfoque de las MSV considera fundamentalmente lo siguiente: El objetivo es crear un modelo que permita etiquetar un texto con una categoria predefinida dado un conjunto de documentos D y un conjunto de categorías C, se trata de encontrar una función que haga corresponder a un documento d tomado de D, una categoría determinada **c en C**.

Palabras Clave: Categorización de textos, clasificación de textos, máquinas de soporte vectorial, clasificadores lineales.

ABSTRACT

The categorization of texts is an application that falls within the discipline of natural language processing and is closely related to the concept of classification. Due to the abundant existing information becomes necessary to organize, maintain, and process any information available from a deeper knowledge of the language of the support vector machines (MSV) belong to the family of linear classifiers, and can be used to resolve the problem of the categorization of texts(CT) which consists in label text or document with one or several predefined thematic categories. The reason which tackles the problem is their application in different scenarios of the area of information retrieval (IR) such as the automatic organization of documents, filtering of documents. The approach of the MSV basically considers the following: Given a set of documents D and a set of categories C, it is important to find a function that match to a document d taken from D, a particular category **c in C**.

Keywords: text categorization, text classification, support vector machines, linear classifiers.

1. INTRODUCCIÓN.

El desarrollo de la ciencia y tecnología viene avanzando aceleradamente, la información en cada área de conocimiento se incrementa en forma exponencial y su tratamiento así como su almacenamiento se hace más complejo. El explosivo crecimiento de la información disponible en documentos digitales en el área de informática y sistemas, 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. Es frecuente que para facilitar la búsqueda de información se proceda a la categorización de los documentos en un conjunto acotado de clases. Estas clases permiten representar áreas específicas del conocimiento y son generalmente consolidadas por expertos. En nuestra vida cotidiana, categorizar es fundamental para la comprensión de ideas, para saber qué hacer, a quién enviar la información, donde guardarla. Si queremos que las computadoras interactúen efectivamente con nosotros y el medio, deben comprender la información que van a categorizar. Por ejemplo, si un folleto de un hotel en un lugar aislado en las montañas describe las características del hotel e incluye mapas y fotografías de los alrededores montañosos, el categorizador descubrirá automáticamente el contenido y vinculará el texto y las imágenes. De esta forma, alguien que busque alojamiento en un lugar aislado en las montañas dentro de cierto rango de precio traería a su pantalla o recuperaría el folleto del hotel aún cuando la expresión "alojamiento en un lugar aislado en las montañas" nunca se mencionara en el texto real.

"Un viajero desea combinar fotos de vacaciones con su diario de viaje para producir un álbum con anotaciones o un blog. Como el categorizador maneja tanto texto como imágenes, éste puede identificar las fotografías, automáticamente corresponderlas con el texto escrito y luego enriquecer las imágenes con información adicional mediante hipervínculos a bases de conocimientos.

ANTECEDENTES

Desde la antigüedad el hombre ha intentado construir máquinas inteligentes. A partir de 1937 comienza el desarrollo de las primeras computadoras como la Maquina de Turing hasta llegar a 1957 donde A. Newell, H. Simon y J. Shaw presentaron el primer programa capaz de razonar sobre temas arbitrarios. Hacia 1960 John McCarthy, acuña el término de inteligencia artificial, para definir los métodos algorítmicos capaces de simular el pensamiento humano en los computadores. Formalmente en 1952 Ashby propone en la conferencia de Darmouth el objetivo ambiciosos de construir dispositivos que actuasencomo "amplificadores de la inteligencia", desde ese entonces la IA ha evolucionado creándose diferentes técnicas que han aportado a la propuesta de Ashby. Entre los métodos teóricos más utilizados están la técnica: Máquinas de Soporte Vectorial (MSV). Las MSV son un paradigma aparte de la Redes Neuronales, pero a pesar de tener similitudes están mejor fundamentadas en la teoría y tienen mucho mejor capacidad de generalización. Uno de los primeros, sino el primero que desarrollo el concepto de Máquinas de Soporte Vectorial (MVS) durante los años 90 del siglo pasado fue Cortes y Vapnik (1995) junto con un equipo de laboratorios AT γT . La MSV se conceptualizo como un método de clasificación óptima, constituida por un conjunto de algoritmos de aprendizaje supervisado desarrollado[13,16]. La propuesta se sustenta en dos columnas vertebrales derivadas de dos ideas:

1. El uso de Kernel y su interpretación geométrica, introducida por Aizerman et al.(1964)
2. La construcción de un hiperplano de separación optima en un contexto no paramétrico, desarrollado por Vapnik y Chervonenkis (1969)

En la actualidad, las Máquinas de Soporte Vectorial pueden ser utilizadas para resolver problemas tanto de clasificación como de regresión. Salazar en [15] señala que la estrategia de regresión logística (RL) y MSV para dos grupos como métodos de clasificación parten de una idea similar aunque ambas reposan sobre soporte teorico distintos. Algunas de las aplicaciones de clasificación o reconocimiento de patrones son: reconocimiento de firmas, reconocimiento de imágenes como rostros y categorización de textos. Por otro lado, las aplicaciones de regresión incluyen predicción de series de tiempo y problemas de inversión en general [9,15,16].

2. PLANTEAMIENTO DEL ESTUDIO

Problema: Debido a al creciente número de documentos accesibles digitalmente, se hace necesario disponer de herramientas automáticas para la gestión de dominios de recuperación de información RI así como

filtrado de documentos. El método de las máquinas de soporte vectorial MSP se considera superior respecto de otros métodos.

Objetivos :

Objetivo general

Construir un modelo que permita etiquetar un texto con una Categoría Temáticas Predefinida

Objetivos específicos

1. Construir un conjunto de aprendizaje a partir de ejemplos
2. Obtener una función que asigne a cada documento un conjunto de categorías asociadas
3. Elegir una técnica para representación de documentos

3. MARCO TEÓRICO

Máquina de soporte vectorial (MSV)

Las máquinas de soporte vectorial o máquinas de vectores de soporte consiste de un conjunto de algoritmos de aprendizaje supervisado que están relacionados con problemas de clasificación y regresión. Su función es construir un hiperplano o conjunto de hiperplanos en un espacio de alta dimensionalidad que separe de forma óptima a los puntos de una clase de otra[3,6].

Las máquinas de soporte vectorial constituyen un método relativamente nuevo que ha concitado interés en los teóricos de la computación específicamente los teóricos en clasificación binaria. La idea básica consiste en encontrar un hiperplano que separa los datos perfectamente en dos clases. Esto permitirá determinar frente a una muestra no conocida previamente de que lado del hiperplano se encuentra.

Consideremos un conjunto de M muestras denotadas como pares ordenados (X_i, Y_i), donde $X_i \in R^n$ y un valor de verdad Y_i que representa un positivo o negativo sobre la observación tratada. Asumiendo que existe cierta distribución de probabilidad no conocida P(x,y) en donde P representa la función de probabilidad acumulada y p su densidad, de donde la data fue obtenida. Nuestro objetivo es obtener lo que denominaremos una 'maquina entrenada. La tarea de nuestra MSV es aprender la relación entre Xi e Yi para todos los elementos de la muestra, esta estará definida por un conjunto de mapeos posibles del tipo $x \rightarrow f(x,a)$. Estas funciones son determinísticas y definidas por ciertos parámetros a, donde una determinada elección de n parametros, en forma adecuada.

En dicha formulación el error de prueba esperado estará definido por

$$R(a) = \int 1/2 |\, y - f(x,a)| \, d\, P(x,y)$$

R(a) es el riesgo esperado o simplemente riesgo que llamaremos riesgo actualpara indicar que es la cantidad en la que estamos interesados en acotar. El riesgo empírico al que denotaremos remplazar por otro lado estará definido como la tasa de error promedio en el conjunto de entrenamiento utilizado

$$R_{emp}(a) = \tfrac{1}{2} l \sum | y_i - f(x_i, a)|$$

MSV Binaria

En los procesos de clasificación binaria solo existen dos clases: una es considerada como la positiva (y=1) y la otra como la negativa (y=-1). No obstante puede ser que exista interferencia dado por un cierto nivel de ruido, o en algunos casos los datos no son linealmente separables. Según esto se pueden emplear distintos tipos de MSV: MSV lineal con margen máximo, MSV para la clasificación no lineal o MSV con margen blando[5]

MSV lineal con margen máximo

Solo se debe usar cuando los datos son linealmente separables, es decir cuando se puede usar como frontera de decisión un hiperplano H(x) tal que

$$H(x) = w^t x + b = 0$$

Donde W y X están en R^d siendo d la dimensión del espacio de entrada

MSV para clasificación no lineal

En los casos que los puntos de la muestra no son linealmente separables, se puede aplicar un proceso de transformación de los datos a un espacio de mayor dimensión £ (denominado espacio de características) en el que los puntos si pueden ser separados por un

hiperplano[5,6]. Cuando esto ocurre se utiliza la función Ø, que cumple

$$\emptyset: \quad R^D \rightarrow £$$
$$X \rightarrow \emptyset(x)$$

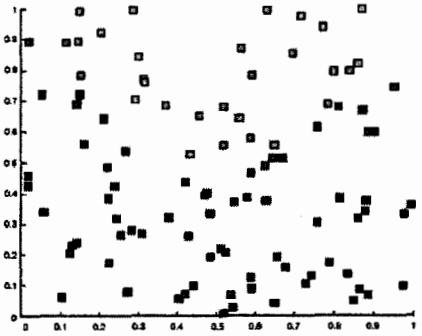


Figura N.° 2. Conjunto de datos no linealmente separable [5]

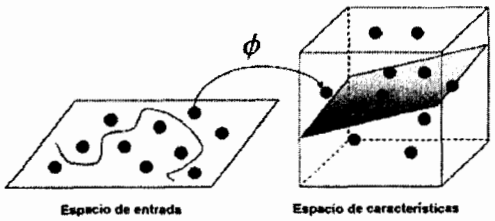


Figura N.° 3. Transformación de los datos de entrada a un espacio de mayor dimensión [5]

Texto

Definiremos un Texto como un documento que está compuesto de palabras.En nuestro contexto excluimos documentos que contienen información no textual como videos, audios, imágenes, etc, debido a la posibilidad que estos archivos estén sujetos a interpretaciónde naturaleza semántico o contenido que depende en gran medida del observador. En el caso de los textos, si bien la problemática anterior también se puede dar, esta está más acotada por que las palabras cuentan con una definición establecida y común para todos[14,15].

Para Daniel Insa y Rosario Morata

> *"El texto refuerza el contenido de la información y se usa básicamente para afianzar la recepción del mensaje icónico, para asegurar una mejor comprensión aportando más datos y para inducir a la reflexión" (1998: 5). La inclusión de texto en las aplicaciones multimedia permite desarrollar la comprensión lectora, discriminación visual, fluidez verbal, vocabulario, etc. El texto tiene como función principal favorecer la reflexión y profundización en los temas, potenciando el pensamiento de más alto nivel. En las aplicaciones multimedia, además permite aclarar la información gráfica o icónica. Atendiendo al objetivo y usuarios a los que va destinada la aplicación multimedia podemos reforzar el componente visual del texto mediante modificaciones en su formato, resaltando la información más relevante y añadiendo claridad al mensaje escrito.* ***[Belloch Ortí, p 5]***

Categorización de textos

La categorización de textos es un tema que forma parte de Recuperación de Información RI (InformationRetrieval) 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]. El contexto del presente trabajo definiremos categorizar como distinguir las características propias de un objeto y establecer las diferencias con otros objetos. En este contexto, categorizar de textos significa relacionar un texto con categorías.

Estrategias de categorización

Existen dos estrategias para la categorización 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 desambiguacion del sentido de las palabras(***WSD*** las siglas en ingles para Word Sense Disambiguation). La segunda estrategia consiste en para abordar el problema anterior, es 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 ob-

tenidos con métodos basados en copus, constituyenen muchos casos la única alternativa realista si se desea hacer uso de información semántica en la representación de documentos. Visto desde esta perspectiva, puede considerarse que el enfoque basado en conocimiento constituye una opción apropiada para la categorización de textos cortos [14].

Hipertexto

Theodor Nelson propuso en 1967 el término hipertexto, para referirse a la estructura interactiva que permite la lectura no secuencial atendiendo a las decisiones del usuario. El hipertexto es una red de información formada a partir de un conjunto de unidades de texto que se conectan por múltiples enlaces.Con ello, en las aplicaciones multimedia interactivas se pueden establecer diferentes tipos de interrelación entre el usuario y el programa, dando mayor o menor libertad al usuario para poder establecer su propio recorrido dentro de la aplicación. El usuario utiliza un sistema de navegación determinada por la estructura de la aplicación atendiendo a la finalidad y características de la aplicación multimedia interactiva.Algo es hipertextual cuando se remplaza la estructura lineal de la información, por un sistema de interactividad basada en los sistemas de hipertexto, de tal forma que el usuario puede decidir y seleccionar la tarea que desea realizar[10,14].

4. MÉTODO Y TÉCNICAS UTILIZADAS.

Metodología

Para modelar el problema P de categorización de textos con las MSV se seguirá la siguiente metodología:

- Definimos:
  D : dominio de todos los documentos
  C : dominio de todas las categorías predefinido
- Binarizar el problema P. Para tal efecto se considerara cada categoría posible de C como un problema binario por separado.
- Para cada categoría aprenderemos una MSV M_i que decidirá si cada documento d pertenece o no a la categoría asociada.

  El objetivo es aprender una función:

$$\emptyset : D \longrightarrow C, \text{ tal que } \emptyset(d_i)=c_i;$$

  d_i es un documento cualquiera y c_i es el vector de las categorías a las que pertenece el documento d_i.

$$\emptyset(d_i) \subseteq C$$

- Para categorizar un documento d en D aplicaremos cada M_i al documento, y devolveremos como resultado las categorías asociadas a M_i que han clasificado como positivo.
- Construccion de un prototipo para hallar la similitud de un texto con alguna clase ya creada. Cuando se ingresa un texto se compara con cada uno 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

Técnica a utilizar

Existen dos técnicas para representar los documentos:

1. Bag-of-words
2. String Kernel

Bag-of-words: Utiliza vectores de palabras para representar los documentos. Cada palabra de un diccionario prefijado corresponde a una dimensión del espacio de documentos.

String Kernel: Cuando se quiere clasificar un documentos d_i, se compara cada subcadena del documento con un referente. Cuanto más sudcadenas tengan en común dos subcadenas mas 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 mas subcadenas en común

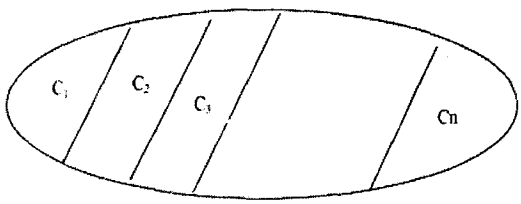


Figura N.° 4. Categorias de texto

Kernel y Máquinas de Vectores Soporte

Las soluciones basadas en MSV no depende de la dimensionalidad de el espacio donde tiene lugar la separación. De esta manera, es posible trabajar en muy

alto espacios dimensionales, tales como los inducidos por los núcleos. Una función que calcula el producto interno entre los ejemplos mapeados en un espacio de características es una función del núcleo, que es para cualquier asignación[5,9]

$$\emptyset: D \rightarrow F, \quad K(di, d_j) = (\emptyset(d_i), \emptyset(d_j))$$

La función Ø transforma un ejemplo n-dimensional en un vector de característica N-dimensional $\emptyset(d) = (\emptyset_1(d), \emptyset_2(d) ..., \emptyset_n(d)) = (\emptyset_i(d))$ para todo i = 1,. . . , N

Hay muchas maneras de combinar núcleos simples para obtener núcleos más complejos. Por ejemplo dado un K del núcleo y un conjunto de n vectores. El kernel polinomio está dado por

$$K_{poly}(di, dj) = (K(di, dj) + c)^p$$

donde p es un número entero positivo y c es una constante no negativa

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, la 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 asignará en la comparación Vemos así, por ejemplo: la subcadena 'r-a' está presente tanto en la "**a**r**e**a" de la palabra y en la palabra pe**r**a, 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 cadenas 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].

$$\Sigma^* = U_{n=0}^{\infty} \Sigma^n$$

Dado un índice de secuencia $i = (i_1, i_2, ..., i_{|u|})$ con $1 \leq i_1 < ... < i_{|u|} \leq |s|$

la subsecuencia se define como $u = s(i) = s(i_1),..., s(i_{|u|})$.

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 $\mathbf{H}_n = R^{(\Sigma^n)}$, 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 1

Considere las palabras ***cima, iman y tina***. Si tenemos en cuenta sólo 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
iman	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

Ejemplo 2

Si consideramos k = 3, obtenemos un espacio de características 8-dimensional, donde las palabras se asignan de la siguiente manera:

	Cim	cia	ima	imn	ian	tin	tia	ina
cima	γ^3	γ^4	γ^3	0	0	0	0	0
iman	0	0^3	γ^3	γ^4	γ^4	0	0	0
tina	0	0	0	0	0	γ^3	γ^4	γ^3

Ejemplo 3

Considere las palabras ***area, pera y barer.*** Si tenemos en cuenta sólo k = 2, obtenemos un espacio de características 10-dimensional, donde las palabras se asignan de la siguiente manera:

	ae	ac	aa	re	ra	ea	ba	be	rr	er
area	γ^2	γ^3	γ^4	γ^2	γ^3	γ^2	0	0	0	0
barer	γ^2	γ^3	0	γ^2	0	0	γ^3	γ^4	γ^3	γ^2
pera	0	0	0	0	0	γ^3	0	0	0	γ^3
rema	0	0	0	γ^2	γ^4	γ^3	0	0	0	0

γ pondera si los elementos de la cadena están contiguos o no.

Así en la palabra área: la subsecuencia "ra" esta contenida en la frase a**rea**, por tanto se pondera γ^3

Así en la palabra barer: la subsecuencia "be" esta contenida en la frase **bare**r, se pondera γ^4

Diseño del clasificador de texto basado en MSV

Se seleccionó una lista de títulos de tesinas de pregrado correspondientes a la carrera de Ingeniería de Sistemas de la Universidad de San Marcos suministradas por la Biblioteca de la Facultad; esta lista de tesis corresponden al lapso de los años 2009 al 2011.

Se consideraron 42 títulos de tesinas:

43 de Ingeniería de Software

39 de Sistemas, Informática y Sociedad

Corpus de entrenamiento: Ochenta palabras con subconjuntos de cuarenta (40) palabras relevantes,

Líneas de Investigación:	Ingeniera de Software	23	
	Sistemas, Informática y Sociedad	19	
	Total		42
	Base de datos	8	
	Inteligencia artificial	4	
	Algoritmos y combinatoria	3	
	Redes y telecomunicaciones	2	
	Total		17

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 consideraron 40 palabras agrupadas en dos grupos:

Ingeniera de software	20
Sistemas, informática y Sociedad	20

Kernel utilizado

Se utilizará un Kernel SSK (Kernel de subsecuencia de cadenas) Sea un alfabeto Σ, entonces definimos

$$\Sigma^* = U_{n=0}^{\infty} \Sigma^n$$

Numero	Título de tesis	Programa de investigación
1	*Solución De Explotación De Información Utilizando Inteligencia De Negocios En Empresas Que Brindan Servicios De Monitoreo Satelital y Gestión De Flotas*	*Ingeniería de software*
2	*Sistema Para La Gestión De La Programación De La Producción En Una Empresa De Manufactura Basado En Agentes*	*Sistemas, informática y Sociedad*
3	*Sistema De Consulta Y Reserva De Citas Médicas Usando Tecnología Móvil Para Mejorar El Servicio De Atención De Pacientes En La Clínica Internacional*	*Sistemas, informática y Sociedad*
4	*Desarrollo De Un Sistema Inteligente De evaluación Para El Otorgamiento De Créditos Financieros en Cooperativas Usando Redes Neuronales Artificiales*	*Ingeniería de software*
5	*Análisis e Implementación Sap Business One Starter Package Para Pequeñas empresas en el Peru*	*Sistemas, informática y Sociedad*
6	*Desarrollo De Software Para La Evaluación Del Aprendizaje En El Curso De Fundamentos De Programación Utilizando El Modelo Sistema Tutor Inteligente*	*Ingeniería de software*
7	*Implementación Del Acuerdo De Nivel De Servicio Para La Gestión De Control Del Outsoursing En Tecnología De La Información Para El Grupo Santillana*	*Sistemas, informática y Sociedad*
8	*Votación Electrónica Para Instituciones Mediante Un Sistema Open Source - Caso: UNMSM*	*Sistemas, informática y Sociedad*
9	*Sistema De Información De Gestión De Tarjetas De crédito Usando Data Mart E Inteligencia De Negocios Para El Area Comercial Del Banco Ripley Peru*	*Sistemas, informática y Sociedad*
10	*Implementación Del Modulo De Gestión De Proyectos De Ti Usando Project Server Para Empresas De TI*	*Ingeniería de software*
11	*Implementación De Un Sistema De Gestión Empresarial Aplicando La Metodología De Programación Extrema, Para Una Empresa Distribuidora De Artículos Descartables*	*Ingeniería de software*
12	*Desarrollo De Aplicación Para Cálculo De Plan De Compras De Insumos, Basado En Mrp, Utilizando Metodología Xp, Para Empresas Manufactureras*	*Ingeniería de software*
13	*Integración De Los Sistemas Informáticos De La Empresa Adaltex Utilizando Oracle ServiceBus Como Middleware Basado En El Estandar Soa*	*Ingeniería de software*
14	*Sistema De Telecomunicaciones Para Casos De Desastres Naturales Usando La Plataforma .Net*	*Sistemas, informática y Sociedad*

Figura N.° 5. Muestra de 14 tesis por linea de investigacion

(a)

$\Sigma 0 = \{ \lambda \}$ define el conjunto de cadenas de longitud 0

$\lambda =$ " " es la única cadena de longitud cero

$\Sigma 1$ define el conjunto de cadenas de longitud 1

$\Sigma 2$ define el conjunto de cadenas de longitud 2

En general

Σn define el conjunto de cadenas de longitud n

A cada cadena $w \in \Sigma n$

Σ^* define el conjunto de todas las secuencias construidas a partir de Σ

le corresponde $\emptyset(w)$

los elementos de w son w1 w2 w3... wn

Si $s, t \in \Sigma n$ entonces $s.t \in \Sigma n$

Sea Σ alfabeto

$\Sigma^* = U \Sigma k$ Para k : 1..n

$W \in \Sigma^*$ si tomamos $|W| = n$

W* cerradura de secuencias construidas sobre W

$W^* = U Wk$ Para k : 1..n

Si consideramos secuencias de lexemas contiguos

$$|W^*| = \sum_{i=1}^{n} |Wi|$$

Ejemplo Sea $\Sigma = \{a,b,c\}$

sea la secuencia w = 'abc'

$W_1 = \{'a','b','c'\}$

$W_2 = \{'ab', 'bc'\}$

$W_3 = \{'abc'\}$

$W = \{'abc'\}$ $\sum_{i=1}^{n} |W| = 2^n - 1$

.....

$W = \{'a','b','c','ab',bc','abc'\}$

Si consideramos secuencias de lexemas no necesariamente contiguos

$W^* = U Wk$ Para k : 1..n

$$|W^*| = \sum_{i=1}^{n} |Wi| = 2n - 1$$

La complejidad asintótica seria de orden exponencial

Si $s, t \in \Sigma^n$ entonces $s.t \in \Sigma^n$

a partir de los ejemplos disponibles de entrenamiento, se debe obtener una función que separe dos clases, esta función se denomina *SVM* entrenada que clasifica apropiadamente las clases, el objetivo es clasificar correctamente los datos o ejemplos que estén fuera del conjunto de entrenamiento, es decir, que la *SVM* capaz de generalizar.

Pueden obtenerse muchos hiperplanos, pero se seleccionará aquel que maximiza la distancia entre el mismo y el objeto más cercano de cada clase (línea sólida), a este hiperplano lineal se le denomina hiperplano separador óptimo (*HSO*).

Nuestro conjunto de entrenamiento es

$$E = \{(x^1, y^1), \ldots, (x^l, y^l)\}, \qquad x \in R^n, \quad y \in \{-1, 1\}$$

La ecuación del hiperplano será

$$\langle w, x \rangle + b = 0. \qquad w \in R^n \, . \, b \in R$$

El propósito es separar en forma optima de tal forma que la distancia entre el vector mas próximo al hiperplano sea máxima [21]. Se tomará el hiperplano canónico en donde se deben restringir los valores de w y b de acuerdo a lo siguiente:

$$\min_i \left| \langle w \, . \, x^i \rangle + b \right| = 1$$

El Hiperplano óptimo en su forma canónica debe satisfacer:

$$y^i \left[\langle w, x^i \rangle + b \right] \geq 1. \qquad i = 1, \cdots, l.$$

la distancia d(w,b;x) de un punto x al HSO es

$$d(w, b; x) = \frac{\left| \langle w, x^i \rangle + b \right|}{\|w\|}$$

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

(b)

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

Funcion Clasificar()

```
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 CN+1
        Sino
                clasificar T en clase K
        FinSi
Fin
```

Funcion Vectorizar(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 siguiente 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
        FinPara
        Retornar contador
Fin
```

Sea el alfabeto $\sum$

y el texto $T_x = a_1 a_2 ... a_n$ $\quad T_x \in \sum$

Sea la secuencia $w = a_1 a_2 ... a_k$

subsecuencia de T_x $\quad w \in T_x^*$

Sea $X = \{\alpha / \text{ Si } E\, w \in \sum^* \text{ y } w \in T_x \rightarrow \alpha = w\}$

clase de texto

$X = n$ cardinalidad de la clase X

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

$T_x \rightarrow \sum_i$

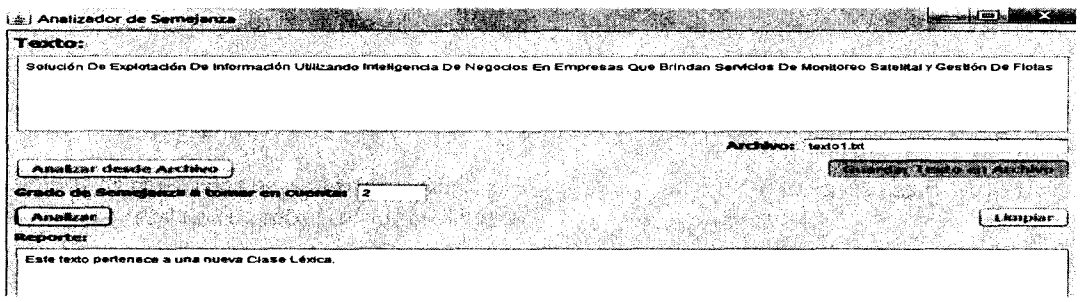
Para todo texto T_x , $\emptyset$ le hace corresponder la clase $\sum_i$

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

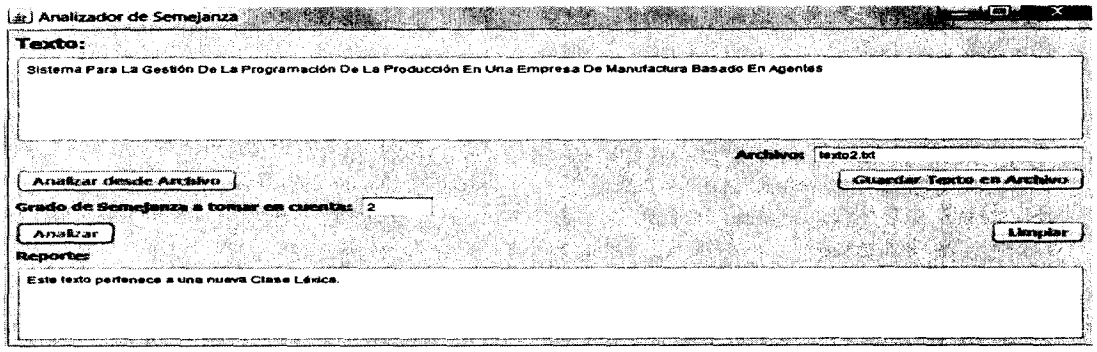
Si $T_u \in T$ elegir v talque $g_{uv} = max\{ g_{ui}$ *para todo i: 1..n}*

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

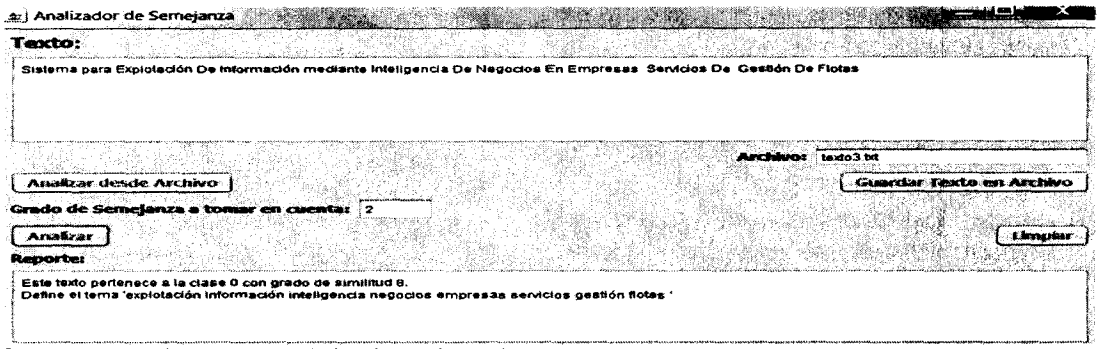
Se analiza el texto de la tesis 1, como es el primer texto ingresado se crea una nueva clase lexica



Se analiza el texto de la tesis 2, se compara el grado de similitud con el texto 1. como es el primer texto ingresado se crea una nueva clase lexica



Se analiza el texto 3
Sistema para explotación de información mediante inteligencia de negocios en empresas servicios de gestión de flotas



Se detecta que el texto 3 tiene similitud con el texto1

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

5. CONCLUSIONES

1. Se utilizó análisis lexicográfico para identificar los lexemas. Constructos aportes de expresión regulares.
2. Se realizó comparación de subcadenas, para determinar el grado de semejanza entre dos textos a mayor subcadenas en comun mayor es el grado de semejanza.
3. La aplicación se desarrolla en JAVA Netbeans Ide 7.2.
4. Cuando se construyó el kernel, se consideró dos alternativas: considerar subcadenas de caracteres contiguas, o no. Sin embargo se debe tener en cuenta que permitir la no contigüidad eleva el costo computacional de la aplicación.
5. La complejidad de las aplicaciones está determinada por la complijidad de la función similar. Las demas funciones son de orden menor.
6. Las investigaciones sobre Máquinas de Soporte Vectorial tienen ciertas características que en ventaja respecto a otras técnicas populares de clasificación y/o regresión como son las redes neuronales y los arboles de decisión.

   El diseño de la función kernel utilizada determinó la eficacia de la MSV construida

   Por ello deberá determinarse la alternativa en función de las prioridades y restricciones de la aplicación a desarrollar.

6. REFERENCIAS BIBLIOGRÁFICAS

[1] Angulo 2001. C, Angulo. Aprendizaje con maquinas núcleos en entornos de multiclasificacion. Tesis doctoral, Universidad Politécnica de Cataluña, Abril 2001.

[2] Ciapuscio, G. 1994. Tipos textuales. Buenos Aires: EUDEBA.

[3] 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 Cuba ISSN 0254-0525

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

[5] Gutiérrez, 2007. Esther Gutiérrez Alonso, 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 2007

[6] Hernández, José. 2009.Introducción a la minería de datos Pearson Prentice Hall España 2008

[7] Krikorian, Mauro Krikorian. Reconocimiento de dígitos manuscritos aplicando transformadas Wavelet sin submuestreo y Maquinas de Soporte Vectorial Tesis de licenciatura Departamento de computación Facultad de ciencias exactas y naturales Universidad de Buenos Aires

[8] Leija, Lorenzo. 2009 Métodos de procesamiento avanzado e inteligencia artificial en sistemas sensores y biosensores. Reverse Editores México 2009.

[9] Lodhi 2002 Huma Lodhi et al Text Clasification using String Kernels. Royal Holloway, University off London, Egham Surrey TW20 0EX, UK

[10] Mendoza, M. 2011. Categorización de texto en bases documentales a partir de modelos computacionales livianos, Revista signos *versión* ISSN 0718-934 vol.44 no.77 Valparaíso dic. 2011

[11] Muller, A. 1997. J. Smola, G. Ratsch, B. Scholkopf, J. Kohlnorgen, and V. Vapnik. Predicting times series with support vector machine. Notas de trabajo, 1997.

[12] Muñoz, P. 2006. Sistema para el Reconocimiento Fuera de Línea de Caracteres Manuscritos Grupo GAMA5 CEIFI, Universidad del Quindío, Agosto de 2.006

[13] Ortega Perea.Categorización de textos biomédicos usando UMLS Perea Ortega, José Manuel, Martín Valdivia, María Teresa, Montejo Ráez, Arturo, Díaz Galiano, Manuel Carl ISSN 1135-5948

[14] Palma, José 2008. Inteligencia Artificial. Edit Mc Graw Hill Madrid 2008

[15] Pedroza, Juan 208. Aplicación de las maquinas de soporte vectorial al reconocimiento de hablantes. Universidad Autónoma Metropolitana Junio 2007

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

[17] Russell, Stuar 2003, Inteligencia Artificial, Un enfoque moderno Edit Pearson México 2003

[18] Salazar 2012. Diego Alejandro Salazar Blandon "Comparación de máquinas de soporte vectorial Vs Regresión Logística ¿Cual es mas recomendable para discriminar ?" Tesis de grado de Magister en Ciencias- Estadística. Universidad de Colombia Facultad de Ciencias Escuela de Estadística Medellín Colombia 2012

[19] Stitson,M. 1996. J. Weston, A. Gammerman, V. Vovk, and V. Vapnik. Theory of support vector machines .InformeTécnico. Bajado de http://svm.rst.gmd.de/, 1996.

[20] Solera 2011. Rubén Solera Ureña "Maquinas 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 2011

[21] Venegas, Rene. Clasificación de textos académicos en función de su contenido léxico-semántico Academic text classification based on lexical-semantic content. Pontificia Universidad Católica de Valparaíso, Chile

[22] Villasana, Sergio. 2008. Categorización de documentos usando máquinas de vectores de soporte. REVISTA INGENIERÍA UC. Vol. 15, No 3, 45-52, 2008 ISSN (Versión impresa): 1316-6832 Universidad de Carabobo Venezuela