CLASIFICACIÓN DE TEXTOS INFORMÁTICOS MEDIANTE INDEXACIÓN SEMÁNTICA LATENTE

CLASSIFICATION OF COMPUTER TEXTS USING LATENT SEMANTIC INDEXING

Augusto Cortez Vásquez*

PRESENTACIÓN: SETIEMBRE DE 2017 APROBACIÓN: OCTUBRE DE 2017

RESUMEN

Debido a la ingente cantidad de información en la web, se requiere de un mecanismo para recuperar contenidos relacionados entre sí en función de su significado semántico. La clasificación de textos tiene como objetivo asociar textos considerando no solo palabras o términos, sino también conceptos. En esta investigación, se presenta una aplicación que crea una matriz de aproximación que contiene información latente relacionada al concepto. De allí el nombre de indexación semántica latente (ISL). Asimismo, dispone de un árbol morfológico que contiene, en cada nodo, una raíz morfológica y un apuntador que se dirige a un conjunto de textos o términos asociados a la misma raíz. Cuando se ingresa un texto, se le clasifica considerando la técnica de análisis léxico e ISL.

Palabras clave: clasificación de textos, recuperación de textos, indexación semántica latente, ISL

ABSTRACT

Due to the huge amount of information on the web, a mechanism is required to retrieve related information based on its semantic meaning, ie related concepts. The classification of texts aims to associate texts considering concepts and not just words or terms. An application that creates an approximation matrix containing latent information related to the concept is presented in this research, hence the name Latent Semantic Indexing (ISL), as well as a morphological tree that contains in each node a morphological root and a pointer to a set of texts or terms associated with the same root. When entering a text, it is classified considering the technique of lexical analysis and ISL.

Keywords: classification of texts, retrieval of texts, latent semantic indexing, ISL

* Universidad Ricardo Palma, Facultad de Ingeniería, <cortez_augusto@yahoo.fr>

1. Introducción

El ser humano tiene la capacidad de comunicarse de diferentes formas utilizando una lengua, lo cual incluye no solo el conocimiento de un conjunto de lexemas y de las estructuras gramaticales, sino también la capacidad de producir y reconocer diferentes tipos de textos en diferentes contextos [1]. Piaget planteó un desarrollo conceptual caracterizado por enfatizar la importancia de la categorización como principal función de los conceptos. Esto se debe a la capacidad de captar semejanzas en función de las apariencias [2]. En ese sentido, la indexación semántica latente pretende descubrir una nueva información a partir de colecciones de documentos de textos no estructurados, entendiéndose por ello "texto libre" generalmente en lenguaje natural [3]. En el contexto de la psicología cognitiva, se ha llegado al convencimiento de que los modelos de la organización conceptual humana, basados en los nodos y las relaciones de las redes semánticas, y en la lista de rasgos de los conceptos, tal vez eran adecuados para abordar el problema de la representación del significado de las palabras y de las oraciones, pero presentaban serias dificultades para dar cuenta del conocimiento estructurado correspondiente a tareas cognitivas más complejas como la percepción, la comprensión o el aprendizaje [2].

Planteamiento del estudio

Problema

El uso de las TIC ha propiciado un creciente número de documentos informáticos accesibles digitalmente, derivado de ello aparece el problema de tener una gran cantidad de información existente en internet. Esto, por supuesto, conduce a la dificultad de acceder a información relevante y de calidad en un contexto en donde, al realizar una búsqueda en un portal especializado como Google, se obtienen miles de enlaces.

Objetivo general

Construir un modelo que permita clasificar textos informáticos considerando sus conceptos asociados y no solo sus términos.

Objetivos específicos

1) Construir un conjunto de clases de textos informáticos.
2) Obtener una función que asigne a cada documento una clase.
3) Elegir una técnica para representación de documentos.

2. Marco teórico

2.1. Recuperación de información RI

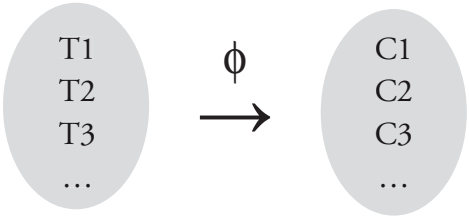
La tarea de recuperación de datos consiste en encontrar documentos que sean relevantes para el usuario que necesita información [4]. Un sistema de recuperación de datos (SRD) requiere las siguientes características:

- **Una colección de documentos**: puede ser un párrafo, una página o un texto de múltiples páginas. EL SRD debe especificar cómo representa los documentos.
- **Un lenguaje de consulta**: puede ser un simple texto, libro de lenguajes o compiladores.
- **Un conjunto resultado**: puede ser subconjunto de documentos que el SRD considera relevantes para la consulta.

- **Una representación del conjunto resultado**: puede ser una lista alineada de títulos de documentos o una colección representada en una estructura.

2.2. Clasificación de textos

La clasificación consiste en reconocer qué es lo que tienen en común algunos textos con otros.



ϕ Hace corresponder a un texto t_i una clase C_j

Decimos que dos textos t_i R t_j (se relacionan) si es que son similares, entendiéndose por similaridad el hecho de que tengan un número en común de subsecuencias [5, 6].

2.3. Indexación semántica latente (ISL)

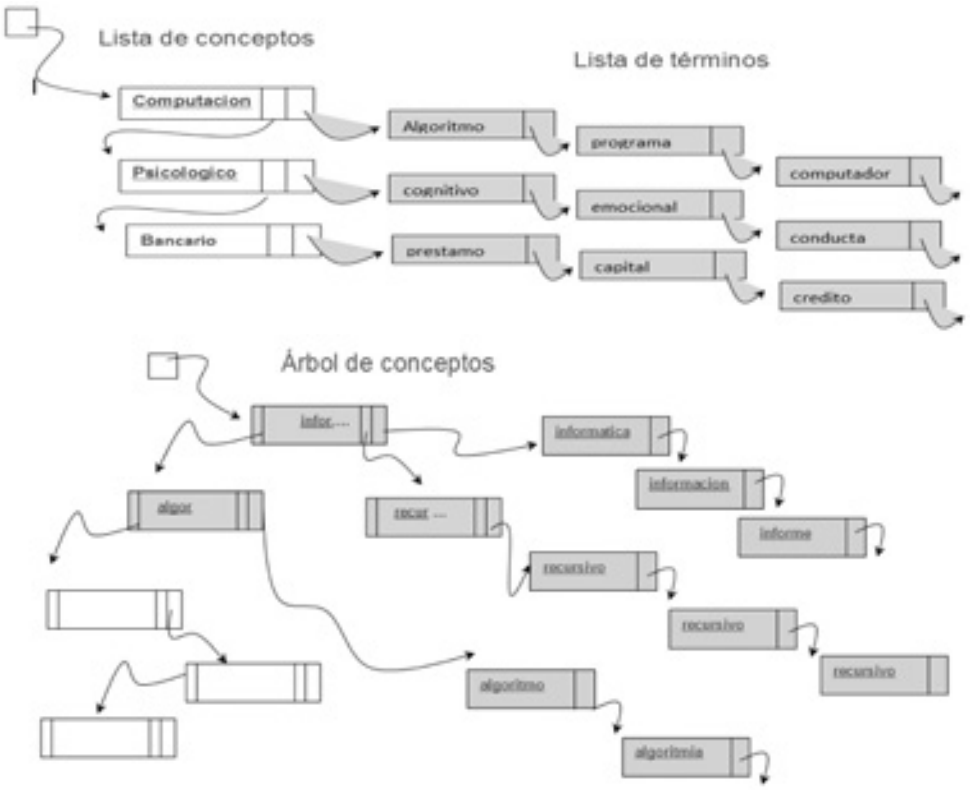
La ISL consiste de una determinada aplicación del Análisis Semántico Latente (ASL), propuesta por Thomas K. Landauer (Pearson Knowledge Technologies) y Susan Dumais (Microsoft) entre otros, quienes postularon al ASL como una nueva teoría general de adquisición y representación del conocimiento [7]. La propuesta consiste en un modelo estadístico que compara las similitudes semánticas entre segmentos de información textual compuesta por lexemas (palabras o grupos de palabras). Este modelo realiza una descomposición de los contenidos en valores singulares (SVD) que, a través de determinados cálculos, establece interrelaciones débiles o latentes (similitudes) entre grupos de palabras y otras áreas del documento a partir de las cuales se infiere un nuevo aprendizaje.

Lo que pretende la ISL es determinar la relevancia de un término o valor en relación a la totalidad del documento u otras partes o términos y valores del mismo, en base su ocurrencia o distancia entre ellos. El principio de ISL se basa en que muchas palabras utilizadas en textos pueden tener significados similares (similaridad de conceptos). La idea principal es emparejar lexemas considerando sus conceptos asociados y no solo por términos (textos). O sea, un documento podría ser recuperado si comparte conceptos con otro que es relevante para la consulta dada. Esto se consigue mapeando los documentos (vector índice de términos) y los vectores consultados dentro de un espacio dimensional reducido el cual está asociado con conceptos, y puede que la recuperación de información, en este espacio reducido, sea superior a la obtenida en el espacio de términos indexados [8, 9].

Aspectos por considerar:

Variaciones de singular y plural: niños, niñas, pelota y pelotas.
Variaciones de masculino y femenino: niño y niña, abogado y abogada.

Sinónimos: balón, pelota, bola etc.
Sustantivos, adjetivos, verbos y variaciones lingüísticas en general: peso, pesado, pesar, etc.



Definimos R ={(x,y) / y está en la lista con cabecera en el nodo del árbol con valor x}
Es decir, xRy si y pertenece a la lista con cabecera en el nodo del árbol con valor x.
Así: Algor **R** Algoritmo, Algoritmica R Algoritmia
Así: *infor* **R** *informática*, *infor* R *información* e *infor*R *informe*
Así: recur **R** *recurrencia*, recur R *recursividad*

3. Método y técnicas utilizadas

Metodología

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

1) Definimos:
   - D: dominio de todos los documentos
   - C: dominio de clases
2) Construir el árbol de conceptos y sus términos asociados.
3) Construir un analizador léxico que detecte los lexemas y las subsecuencias que componen al texto.
4) Para clasificar un documento d en D, encontraremos el grado de similitud entre dos textos. Para esto, utilizaremos la técnica de String Kernel e indexación semántica latente.

5) Construir 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. Se utilizará la técnica de indexación semántica latente, que permita recuperar textos a partir de conceptos y no solo de términos.

Ø verifica a que clase pertenece el texto Tx

Especificación formal:
Precondición: El texto Tx se encuentra depurado.
Fun Categoriza (Tx: secuencia de texto) dev (C_k: clase)
Postcondición: Existe K tal que T_k está en C_k y similar (Tx,Tk).

Tx debe encontrarse limpio, es decir, tienen que haberse abstraído los lexemas irrelevantes.

Función de similitud

T_x y T_k pertenecen a una clase $\mathbf{C_k}$ si se relacionan por la función similar (T_x,T_k) para un α: grado de similitud.

Sea S_x: conjunto de subsecuencias del texto T_x
S_k: conjunto de subsecuencias del texto T_k
$|S_x| = m$ $|S_k| = n$
Existe $S_{x\cap k}$

Conjunto de subsecuencias en común de S_x y S_k
el número de subsecuencias en común entre T_x y T_k denotado $| S_{x\cap k} |$ es mayor o igual a α
Ø: $D \rightarrow F$,
$K (d_i, d_j) = (Ø (d_i), Ø (d_j))$

La función Ø transforma un ejemplo n-dimensional en un vector de característica N-dimensional $Ø (d) = (Ø_1(d), Ø_2(d) ..., Ø_n (d)) = (Ø_i (d))$ 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, las más similares [8, 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 palabra "area" y en la palabra pera, pero con diferente ponderación.

Puede ocurrir que en un texto aparezcan los mismos lexemas, aunque en orden diferente. 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 [10].

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].

$$\Sigma^* = U \sum_{n=0}^{\infty} n$$

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

Para el análisis **ISL**, primero se construye una matriz C donde las filas representan los términos y las columnas los documentos, esta matriz establece las relaciones término documento por lo que cada elemento C_{ij} representa el peso del término ***i*** en el documento ***j***. Estos pesos pueden ser calculados como el producto del peso local del término l_{ij} en un documento específico y el peso global del término en la colección de documentos g_{ij}. Los pesos anteriores pueden ser calculados de diversas formas como se muestran en las tablas a continuación [11, 7].

Binaria	$l_{ij} = 1$ si el término existe en el documento, 0 en otro caso
Frecuencia de término	$l_{ij} = \mathrm{tf}_{ij}$, el número de ocurrencias del término i en el documento j
Log	$l_{ij} = \log(\mathrm{tf}_{ij} + 1)$
Augnorm	$l_{ij} = \dfrac{\left(\frac{\mathrm{tf}_{ij}}{\max_i(\mathrm{tf}_{ij})}\right) + 1}{2}$

Funciones de peso local

Binaria	$g_i = 1$
Normal	$g_i = \dfrac{1}{\sqrt{\sum_j \mathrm{tf}_{ij}^2}}$
GfIdf	$g_i = \mathrm{gf}_i / \mathrm{df}_i$, donde gf_i es el número de veces que i ocurre en toda la colección, y df_i es el número de documentos en los cuales ocurre el término i.
Idf	$g_i = \log_2 \dfrac{n}{1 + \mathrm{df}_i}$
Entropy	$g_i = 1 + \sum_j \dfrac{p_{ij} \log p_{ij}}{\log n}$, donde $p_{ij} = \dfrac{\mathrm{tf}_{ij}}{\mathrm{gf}_i}$

Funciones de peso global

El objetivo es construir la matriz C_k que se aproxime a la matriz de términos de documentos C. En esa aproximación, se va a obtener información que no estaba disponible directamente en la matriz C, sino que se encontraba latente en esta.

Diseño del categorizador de texto basado en String Kernel

Kernel utilizado

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

$$\Sigma^* = U \sum_{n=0}^{\infty} n$$

$\Sigma^0 = \{\lambda\}$ 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

A cada cadena $w \in \Sigma^n$
le corresponde Ø(w)
los elementos de w son $w_1\ w_2\ w_3 \ldots\ w_n$

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

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

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

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 C_{N+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

Función similar (X, Y)
Inicio
contador = 0
Mientras existan w SubSecuencias de X
Para cada w' / w' R w
Si w' es Subsecuencia de Y
contador = contador +1
FinSi
FinPara
FinMientras
Retornar contador
Fin

Sea Ø: T $\longrightarrow$ $\sum$
T_x $\longrightarrow$ $\sum_i$

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*** talque $g_{uv} = max\{g_{u\,i}$ *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 relevante. 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 también por conceptos.

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

Concepto: Pelota	**Textos**
Conceptos relacionados: Balon bola	Los muchachos lanzaron la pelota Los muchachos lanzaron el balon 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 y halla el grado de similitud con las categorías existentes.

Ejemplo

Consideremos los siguientes textos:

T_1	Las computadoras digitales son herramientas para procesamiento de datos
T_2	La computación permite procesar datos en forma eficiente
T_3	Procesar datos mediante computadoras es eficiente
T_4	Las redes neuronales son herramientas eficientes

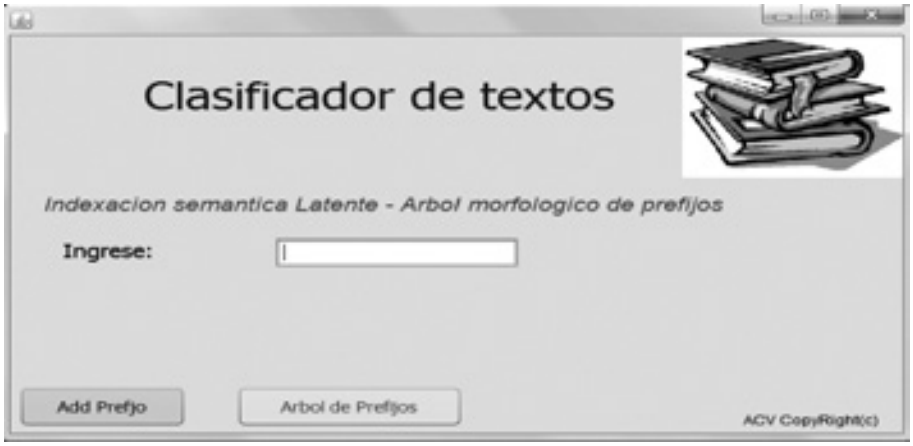
Textos vectorizados: se eliminan los lexemas irrelevantes y se considera las raíces morfológicas.

V_1	**comp**	**Digi**	**herr**	**proc**	**dat**	
V_2	**Comp**	**Proc**	**dat**	**efic**		
V_3	**Proc**	**Dat**	**comp**	**efic**		
V_4	**Red**	**neur**	**herr**	**efic**		

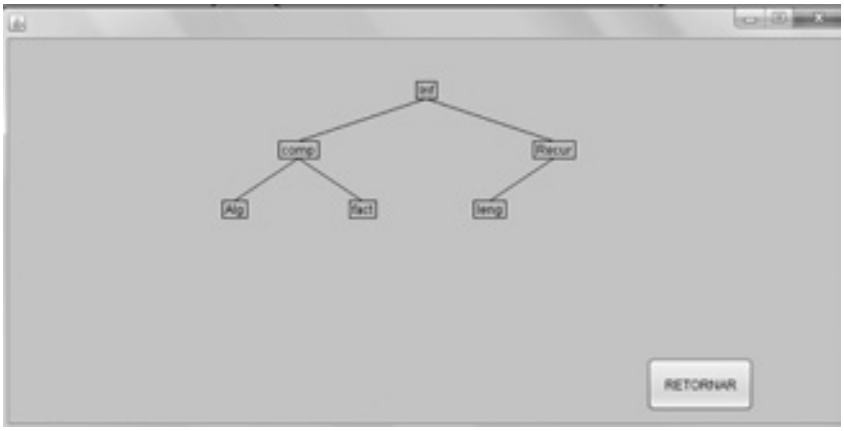
Textos relacionados según subsecuencias

Textos vectorizados	**Subsec long 1**	**Subsec long 2**	**Subsec long 3**	**Subsec long 4**	**Número subsec en comun**
V_1 R V_2	Comput Proc Dat	Comp proc Comp dat	Comp proc dat		6
V_1 R V_3	Comput Proc Dat	Comp proc Proc comp	Comp proc dat Proc dat comp		7
V_1 R V_4	Herr	--	--		1
V_2 R V_3	Comp Proc Dat efic	Comp proc Comp dat Comp efic Dat efic Proc dat Proc efic	Comp proc dat Comp proc efic Comp dat efic Proc dat efic	Comp proc dat efic	15

Se concluye que V_1 tienen mayor grado de asociación con V_3

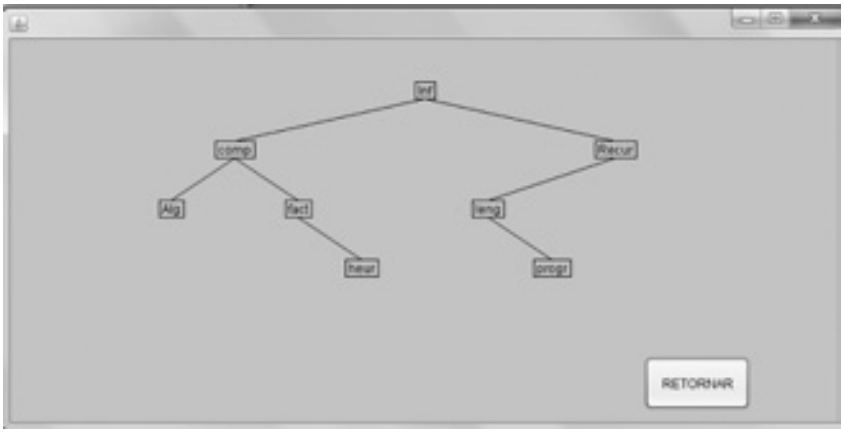


Después de ingresar las raíces: inf, recur, comp, alg, fact, leng, el árbol queda de la siguiente forma:



Cuando se ingresa las palabras *informática* e *información*, estas se insertarán en la lista apuntada por el nodo ***infor*** del árbol. De la misma forma, cuando se ingresen los términos *algorítmica, algoritmia, algoritmo*, se insertarán en la lista apuntada por el nodo ***alg*** del árbol.

Después de ingresar las raíces ***heur*** y ***prog***, el árbol quedará de la siguiente forma:



Los términos ***heurística*** y ***programación*** se insertará en las listas apuntadas por los nodos heur y prog respectivamente.

5. Conclusiones

a) La indexación semántica (ISL) resulta una buena aproximación de solución a dos de los principales problemas de las consultas booleanas: la sinonimia y la polisemia. Se puede utilizar para realizar una categorización automática de los documentos y particionarlos.
b) Mediante la indexación semántica latente, los buscadores consiguen "contextualizar" los términos y conceptos de las búsquedas, para ofrecer así al usuario un resultado más correcto.
c) Para establecer la similitud de dos textos, se utilizó la técnica String Kernel. Mientras más secuencias en común tengan dos textos, se determina que son más similares.

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 permite que, cuando se recupera un texto, la búsqueda se realiza por conceptos y no solo por términos.

6. Referencias bibliográficas

[1] G. Ciapuscio, *Tipos textuales*. Buenos Aires: EUDEBA, 1994.
[2] A. Rafael Linares, "Desarrollo Cognitivo: Las teorías de Piaget y de Vygotsky". Universidad Autónoma de Barcelona. [En línea]. Disponible en http://www.paidopsiquiatria.cat/files/teorias_desarrollo_cognitivo.pdf. [Accedido: 15-may-2017]
[3] L. Gómez, "La iniciativa de Archivos Abiertos (OAI), un nuevo paradigma en la comunicación científica y el intercambio de información", *Revista Códice*, no. 2, 21-47/julio-diciembre, 2005.
[4] M. Rosas, "Un análisis comparativo de estrategias para la categorización semántica de textos cortos", *Revista Procesamiento del Lenguaje Natural*, no. 44, pp. 11-18, marzo de 2010.
[5] A. Cortez Vásquez, "Categorización de textos mediante máquinas de soporte vectorial", *Revista RISI* (Revista de Investigaciones de Sistema e Informática) vol. 10, no. 1, Lima-Perú p. 33, 2013.
[6] J.M. Perea Ortega, *Categorización de textos biomédicos usando UMLS*, J. M. Perea Ortega, Ma. T. Martín Valdivia, A. Montejo Ráez, M. C. Díaz Galiano, ISSN 1135-5948.
[7] R. Venegas, "Análisis Semántico Latente: una panorámica de su desarrollo", *Revista Signos*, 36(53), 121-138: Pontificia Universidad Católica de Valparaíso, Chile, 2003.
[8] J. Palma, *Inteligencia artificial*. Madrid: Mc Graw Hill, 2008.
[9] S. Russell, *Inteligencia Artificial, Un enfoque moderno*. México: Pearson, 2003.
[10] L. Huma et al., *Text Clasification using String Kernels, Royal Holloway*. London: University of London, Egham Surrey TW20 0EX, UK, 2002.
[11] R. Venegas, "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.
[12] C. Angulo, "Aprendizaje con máquinas núcleos en entornos de multiclasificación", tesis doctoral, Universidad Politécnica de Cataluña, 2001.
[13] A. Gonzales, "Modelos de Clasificación basados en Maquinas de Vectores-Soporte". Departamento de Economía Aplicada I, Universidad de Sevilla. [En línea]. Disponible en http://www.asepelt.org/ficheros/File/Anales/2003%20-%20Almeria/asepeltPDF/55.pdf. [Accedido: 14-may-2017]
[14] E. 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.
[15] J. Hernández, *Introducción a la minería de datos*. España: Pearson Prentice Hall, 2008.
[16] M. 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.
[17] L. Leija, *Métodos de procesamiento avanzado e inteligencia artificial en sistemas sensores y biosensores*. México: Reverse Editores, 2009.
[18] M. Mendoza, "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.
[19] A. Muller, J. Smola, G. Ratsch, B. Scholkopf, J. Kohlnorgen, and V. Vapnik, *Predicting times series with support vector machine*. Notas de trabajo, 1997.

[20] P. Muñoz, *Sistema para el Reconocimiento Fuera de Línea de Caracteres Manuscritos Grupo GAMA5 CEIFI*, Universidad del Quindío, agosto de 2006.

[21] J. Pedroza, "208 Aplicación de las máquinas de soporte vectorial al reconocimiento de hablantes", Universidad Autónoma Metropolitana, junio 2007.

[22] D. Salazar Blandon, "Comparación de máquinas de soporte vectorial vs regresión logística. ¿Cuál es más recomendable para discriminar?", tesis de Magíster en Ciencias Estadísticas. Universidad de Colombia, Facultad de Ciencias Escuela de Estadística, Medellín-Colombia, 2012.

[23] M. Stitson, J. Weston, A. Gammerman, V. Vovk, and V. Vapnik. "Theory of support vector machines", informe técnico, 1996. [En línea]. Disponible en http://svm.rst.gmd.de/. [Accedido: 22-ago-2017]

[24] R. Solera Ureña, "Máquinas de vectores de soporte para reconocimiento robusto del habla", tesis doctoral. Departamento de Teoría de la Señal y Comunicaciones. Madrid: Universidad Carlos III de Madrid Leganez, 2011.

[25] S. Villasana, "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.

[26] E. Martínez. ¿Qué es la Indexación Semántica Latente y cómo usarla en tu estrategia de Posicionamiento SEO?, [En línea]. Disponible en http://comunidad.iebschool.com/iebs/general/indexacion-semantica-latente/. [Accedido: 22-jun-2017].