CLASIFICACIÓN AUTOMÁTICA DE TEXTOS MEDIANTE COMPARACIÓN DIFUSA Y SISTEMA EXPERTO EN UN CONTROLADOR DOMÓTICO.

Héctor Manuel Quej Cosgaya
Juan Miguel Durán Lugo
Guadalupe Manuel Estrada Segovia

RESUMEN

El presente trabajo, describe el diseño de un agente inteligente de clasificación automática de textos para un Controlador Domótico, la categorización autónoma de las diversas cadenas léxicas de información, utilizan técnicas y métodos de Inteligencia Artificial. Estas técnicas y métodos son Lógica Difusa Compensatoria (LDC), Sistemas Expertos (SE) y la Distancia de Edición (DE) de Levenshtein en forma articulada en un motor inteligente para la toma de decisiones, para así, categorizar de una tupla de información los diversos tipos de palabras mediante un controlador domótico y proporcionar a los usuarios de una experiencia confiable, acertada y a la vez, eficaz y rápida al momento de realizar sus órdenes en una casa habitación.

INTRODUCCIÓN

Durante la historia de la humanidad han surgido variadas formas de comunicación y expresión, una de ellas y la más importante es la escritura que trasciende a través de los tiempos. El hombre se ha empeñado en plasmar sus conocimientos, pensamientos, anhelos, etc., a través de ella, creando acervos documentales y muchas otras formas de concentración de información. La tecnología, compañera fiel de la contemporánea humanidad, ha hecho posible que todo acervo se concentre de forma digital, binaria, ceros y unos, creando la fuente de información más grande y extensa de nuestra época. La masividad de información a través de los distintos medios electrónicos actuales como: bibliotecas digitales, páginas web, comentarios en foros o redes sociales, tuits, transcripciones de llamadas telefónicas, correos electrónicos, etc, generan la necesidad de desarrollar diversas formas de manipulación para su extracción y análisis, entre dichos desarrollos se encuentra la clasificación de textos. Sin embargo, el constante y exagerado crecimiento de información hace que la tarea de clasificar documentos de forma manual sea costosa y que requiera de mucho tiempo, por lo que ha surgido el interés por realizar la clasificación de manera automática.

Actualmente existen diversas líneas de investigación para el tratamiento automático de textos, entre las que se encuentran: la recuperación de información, la extracción de información, la búsqueda de respuestas y la clasificación de textos entre otras (Baeza-Yates., 2000). La clasificación de textos es una tarea importante que facilita la organización de información y consiste en determinar la categoría de un texto, de entre varias categorías predefinidas, de acuerdo a ciertas características identificadas en dicho texto. En estudios realizados para la Clasificación Automática de Textos se han enfocado en clasificar textos por su tema, es decir, determinar a que tema o temas pertenece un documento, de entre varios temas a priori. (Erikson, 2011) (G.J.Klir, 1995).

En este contexto, se ha diseñado un agente de clasificación automática de cadenas de texto para determinar su categorización por tipo de palabra mediante órdenes dadas en un Sistema Domótico (SD) utilizando técnicas de inteligencia artificial (IA) (Russell, 2004), dichas técnicas son: Lógica Difusa Compensatoria (LDC) (Raya, 2006) y Sistemas Expertos (SE) (Giarratano & Riley, 2001) (K. Aas, 1999). Ambas técnicas trabajan sincronizadamente con un algoritmo de cálculo de la Distancia de Edición (DE) de Levenshtein para realizar la clasificación automática de las tuplas de información obtenidas de los documentos fuente e identificar de forma autónoma el tipo-palabra de texto codificado en un sistema de lectura-escritura, que tiene como propósito, entregar información y emitir una categorización de manera automática e inteligente. El artículo se organiza de la siguiente forma: en la sección 2 abordamos la descripción del agente de clasificación automática, en la sección 3 se describe la metodología del algoritmo inteligente, en la sección 4 y 5 se enumeran los resultados alcanzados y las conclusiones, de igual forma se proponen las líneas de trabajo futuro.

JUSTIFICACIÓN

Agente de clasificación automática

En este trabajo, nos enfocamos en la categorización por tipo de palabra de un universo de textos obtenidos por nuestro (SD) de forma automática mediante un análisis léxico difuso (ALD) y el algoritmo relativo conocido como: “la distancia de edición” (DE) de Levenshtein (Baeza-Yates., 2000) en combinación de una base de conocimientos, reglas y hechos de nuestro SE. En investigaciones similares previas se han propuesto el uso de diferentes conjuntos de atributos para realizar la clasificación de textos como si fuera una bolsa de palabras, o partes de la oración (Lewis., 1998) (Téllez, 2003) (N. Castell, 1997). Otro punto relevante de nuestro trabajo es que se propone evaluar el método en el idioma español. Prácticamente, todos los trabajos previos que se han realizado dentro del área de clasificación de textos, se ha evaluado en textos en el idioma inglés. Por supuesto, esto conlleva la necesidad de conformar un corpus para el español.

En este sentido, nuestro algoritmo debe contar con algún tipo de corpus-conocimiento previo en español, esto no es más que una serie de atributos léxicos que son llamamos *“patrones”* clasificados por grupos que denominamos “categorías” {artículo, sustantivo, pronombre, adjetivo, verbo, adverbio, preposición, conjunción e interjección}. Esto nos permitirá realizar la DE y ALD de los patrones encontrados en un texto y determinar la categoría perteneciente.

El algoritmo, por lo tanto, además de realizar el reconocimiento nuevos patrones en un texto es capaz de analizar una gran cantidad de textos u oraciones para hallar patrones comunes a un determinado grupo, pues de lo contrario, el proceso de registro de patrones llevaría en sí mismo un tiempo bastante considerable de ser realizado por un agente humano.

Tomando en cuenta esto, el algoritmo del agente inteligente cumple con las siguientes características:

a) Permite identificar cadenas semejantes, no necesariamente idénticas, de la misma manera que lo haría un ser humano.
b) Aplica criterios de selección de la categoría más adecuada para la correcta clasificación del tipo de texto en caso que se encuentren dos o más coincidencias para una determinada categoría.
c) Analizar gran cantidad de cadenas de texto, para encontrar semejanzas entre ellas que permitan generar nuevos patrones.
d) Aplicar criterios de restricción para evitar que palabras demasiado comunes puedan ser consideradas como patrones.
e) Realizar la categorización del texto en consideración de su base de reglas del SE

METODOLOGÍA

El primer paso del análisis es filtrar palabras repetidas: esto con la finalidad de reducir el espacio muestra y no comprometer el rendimiento. Ya con las palabras idénticas eliminadas, cada lexema de cada conjunto de palabras se agrega a una nueva lista, y el proceso se repite de nuevo: los lexemas idénticos que aparezcan más de una vez se contabilizan para determinar el número de incidencias, conteo que se almacena en una lista de lexemas repetidos. Estas dos listas: la lista de lexemas repetidos y la lista de lexemas únicos son las que se utilizan durante el análisis.

El análisis de lexemas repetidos es bastante simple: Se determina si un lexema como tal debe convertirse en un patrón basado en el número de incidencias, de esta manera los lexemas que aparezcan en un número suficiente de veces son automáticamente consideradas como patrones y se incluyen en una bolsa de categoría. El análisis de los lexemas únicos, por su parte, requieren un poco más del uso de técnicas de Inteligencia Artificial para concretarse. Primeramente, cada lexema único se somete a un análisis de comparación difuso, el cual determina en qué porcentaje un lexema se parece a otro (Ozsoyoglu., 1997). Si ambos lexemas superan un determinado umbral, ambas se consideran como candidatos a generar un nuevo patrón, por lo que se agregan a una lista de candidatos. Los lexemas que no superen la prueba se descartan para compararse con otros lexemas en iteraciones posteriores. La Figura 1 muestra el análisis de cada lexema de cada una de las listas configuradas para la clasificación correcta.

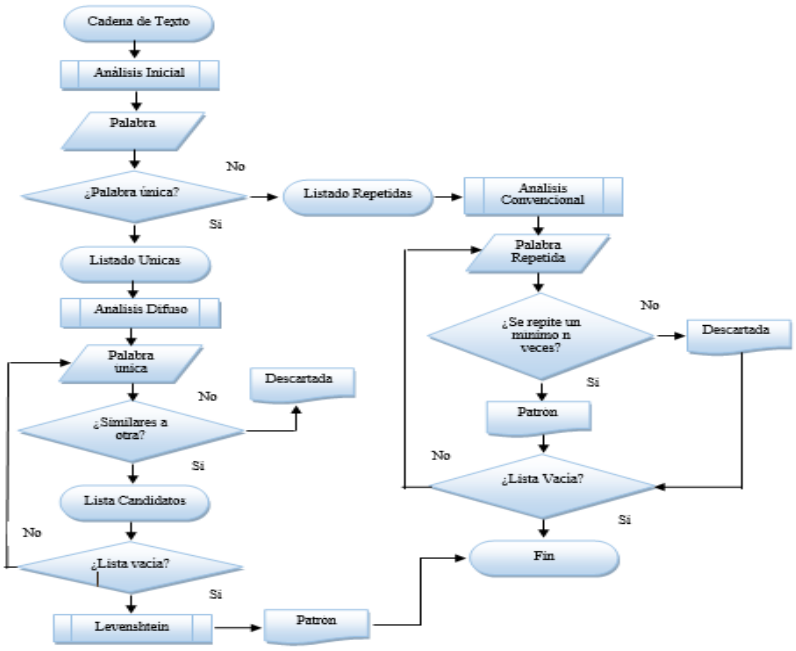

Figura. 1: Diagrama de flujo del agente inteligente

Como podemos observar en la Figura 1, realizamos el ALD y luego el DE para el reconocimiento de cadenas de texto. De esta manera, contamos con dos métodos principales para desarrollar la metodología del ALD:

1) El más simple, es el “contiene”, que verifica si un patrón aparece en determinada cadena. Este es el método utilizado en la lógica tradicional.

2) El método “porcentaje”. Este método utiliza la comparación difusa de cadenas para comprobar “que tanto una cadena se parece a otra”, es decir su grado de pertenencia (Raya, 2006) (G.J.Klir, 1995). El grado de pertenencia de la colección léxica es determinado por el método trapezoidal mostrado en la Figura 2.

$$\mu(x)=\begin{cases}0 & \text{para } x\le a\\ \dfrac{x-a}{b-a} & \text{para } a<x\le b\\ 1 & \text{para } b<x\le c\\ \dfrac{d-x}{b-c} & \text{para } c<x\le d\\ 0 & \text{para } x>d\end{cases}$$

Función de pertenencia

Figura. 2: Función de pertenencia del método trapezoidal (Raya, 2006)

El análisis difuso tratará buscar a todos los lexemas parecidos entre sí, para intentar determinar patrones. Si tiene éxito en buscar semejanzas entre lexemas, el resultado es un nuevo patrón. Dicho patrón se almacena de nuevo en la sección clasificadora para que los nuevos lexemas puedan ser clasificadas posteriormente, en la Ecuación (1), se describe el grado de pertenencia para el cálculo del método trapezoidal del subconjunto a y b.

$$\mu_A(x) = \frac{(x-a)}{(b-a)}$$

Ecuación. 1: Función de pertenencia (Raya, 2006)

Dichos grados de pertenecía se calculan en relación a los valores asociados a las cadenas de caracteres de los textos, es decir el valor nominal del carácter. En la siguiente tabla (Tabla 1) se indican los valores asociados al intervalo del universo en estudio.

Tabla 1: Nomenclaturas de los grados de pertenecía (Raya, 2006)

Nomenclatura	Concepto Asociado
$\mu_A(x) = 0$	El objeto no pertenece al conjunto
$0 < \mu_A(x) < 1$	El objeto pertenece parcialmente al conjunto
$\mu_A(x) = 1$	El objeto pertenece totalmente al conjunto

La lista de lexemas candidatos se somete a un análisis final: mediante la técnica de la Distancia de Levenshtein (ver figura. 3), que utiliza una matriz de asociación difusa que extrae la raíz que comparten los lexemas candidatos. Con cada iteración sucesiva, la raíz se va haciendo cada vez más y más refinada, hasta que se obtiene una estructura léxica común a todas y cada una de los lexemas que fueron candidatos. Se determina entonces si dicho lexema cumple los parámetros necesarios para considerarse un patrón, como la longitud mínima de caracteres que posea. Si se determina que la estructura léxica obtenida satisface los requerimientos, se registra como un nuevo patrón, el cual será capaz de clasificar a todas las futuras transacciones que incluyan alguno de los lexemas candidatos de la cual surgió.

```
int ALD(char cadena1[1..longitud1], char cadena2[1..longitud2)
  declare int dis[0..longitud1, 0..longitud2]
  declare int ren, col, costo

  for ren from 0 to longitud1
     d[ren, 0] := ren
  for j from 0 to longitud2
     d[0, col] := col

 for ren from 1 to longitud1
     for col from 1 to longitud2
        if cadena1[ren] = cadena2[col] then costo := 0
                    else costo:= 1
        dis[ren, col] := minimo(d[ren-1, col] + 1, d[ren, col-1] + 1,d[ren-1, col-1] + cost)

  return dis[longitud1, longitud2]
```

Figura. 3: Algoritmo de la distancia de edición de Levenshtein

(Ozsoyoglu., 1997) (Heeringa, 2004)

En la Tabla 2, se ejemplifica el algoritmo de la distancia de edición de Leveshtein identificando en cada intersección matricial el número mínimo de operaciones requeridas para transformar una cadena de caracteresen otra, se usa ampliamente en teoría de la información y ciencias de la computación.

Tabla 2: Distancia de Edición de Levenshtein

DISTANCIA DE EDICIÓN						
	d[j]	L	U	N	E	S
d[i]	0	1	2	3	4	5
L	1	**(2,2,0)=0**	(3,1,2)=1	(4,2,3)=2	(5,3,4)=3	(6,4,4)=4
U	2	(1,3,2)=1	**(2,2,0)=0**	(3,1,2)=1	(4,2,3)=2	(5,3,4)=3
N	3	(2,4,2)=2	(1,3,2)=1	**(2,2,0)=0**	(3,1,2)=1	(4,2,2)=2
A	4	(3,5,4)=3	(2,4,3)=2	(1,3,2)=1	**(2,2,1)=1**	(3,2,2)=2
S	5	(4,6,5)=4	(3,5,4)=3	(2,4,3)=2	(2,3,2)=2	**(3,3,1)=1**

DISCUSIÓN DE RESULTADOS

Las reglas de inferencia del SE para la solución del único lexema se conforma por 3 partes: la base de conocimiento, que es en donde se almacena todo el conocimiento almacenado por el sistema, el cual utiliza para resolver las situaciones que se le presenten; el motor de inferencia, que es el que el sistema utiliza para resolver situaciones imprevistas para las cuales el conocimiento con el que cuenta no es suficiente; y la tercera y última es la interfaz, que es la manera en como el sistema recibe datos del mundo exterior y envía a éste sus respuestas (Giarratano & Riley, 2001) (Martinez, 2004). En la Figura 4 se muestra el motor de conocimientos relacionado con el análisis difuso para la resolución de conflictos, dicho análisis compara con el método de pertenencia trapezoidal de la Figura 2, las cadenas de texto entrantes con una lista de palabras almacenadas, si esta es similar a otra se considera un candidato para un próximo análisis de distancia de edición, en caso contrario se descarta.

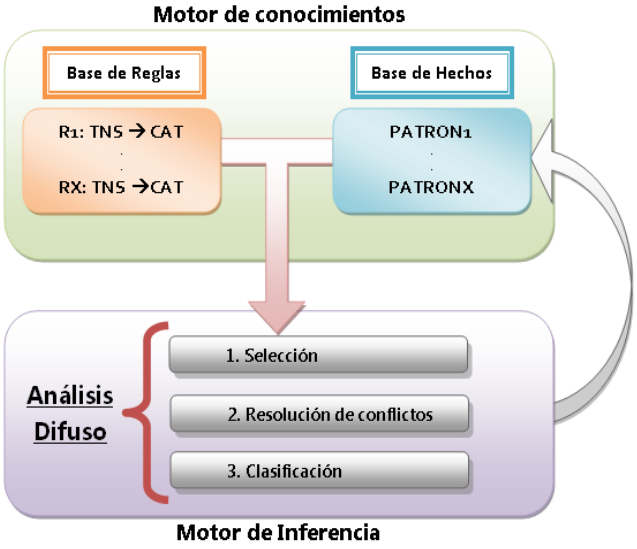

Figura. 4: Motor de Inferencia del SE

Estas reglas se focalizan en categorizar los lexemas que no son reconocidos en los apartados de la metodología ALD y DE previamente mencionados, cuando lexema no puede ser clasificado, se almacena temporalmente en una lista. Periódicamente, esta lista se somete a un proceso de análisis (base de hechos de nuestro experto) para extraer nuevos patrones de dichas transacciones lo que ayudará a sus futuras clasificaciones.

Las reglas de solución de la nominada "lista negra", se utilizan en caso de que se encuentre un lexema único, es decir no tiene parecido con patrón alguno, este se compara de forma algorítmica difusa con una lista de candidatos únicos, si este encuentra una coincidencia y supera el umbral se logra un nuevo patrón, a continuación se muestra el pseudocódigo de la regla de conflicto de la "lista negra".En la tabla 3 se enlista el Pseudocódigo de las reglas para el caso del único lexema.

Tabla 3: Pseudocódigo de reglas para el caso "único lexema"

Clasificación:	**Análisis: (La lista negra de conflictos)**
INICIO Leer lexema Por cada patrón en lista de patrones Comparar lexema con patrón Si hay una coincidencia Imprimir categoría Ir a FIN	INICIO Leer lista negra Por cada lexema en lista negra Extraer lexema de la lista Agregar lexema extraídas a lista de lexemas únicos Si el lexema ya se encuentra en lista única: Incrementar contador de lexema En caso contrario Agregar lexema a lista única Por cada lexema en lista de lexemas únicos Si el lexema se repite un número suficiente de

Si hay más de una coincidencia Aplicar reglas de Conflictos coincidentes Imprimir categoría vencedora Ir a FIN Si no hay coincidencia **Enviar lexema a lista** **Negra de conflictos** FIN	veces Agregar lexema a lista de patrones encontrados En caso contrario Por cada otro lexema en lista única Si lexema y otro lexema se asemejan Agregar ambos lexemas a lista de semejantes Si hay lexemas en lista de semejantes Extraer patrón de la lista de lexemas Agregar patrón a lista de patrones encontrados Eliminar lexema que se apeguen al patrón de lista negra Regresar lista de patrones encontrados FIN

CONCLUSIONES

Se ha propuesto el diseño de un algoritmo que permita identificar los lexemas de una tupla de información obtenidas por un Controlador Domótico y clasificar de forma automática las cadenas de textos de un documento y categorizarlos según su tipo. Las técnicas de inteligencia artificial LDC someten a los lexemas de las tuplas de cada patrón a un análisis de comparación difuso para determinar en qué porcentaje un lexema se parece a otro y poder realizar la clasificación de la categoría, o en su caso, la creación de un nuevo patrón.

Dichos patrones ya comprobados, tendrán asociados su categoría correspondiente, desde el inicio. Los patrones y su correspondiente clave se almacenarán en una estructura tipo tabla hash o parecida, que los mantenga fácilmente ordenados y que puedan ser recuperados de manera eficiente, sin tener que recorrer cada uno de ellos.

Se propone el diseño en pseudocódigo de un motor de inferencia basado en un SE, este resuelve las reglas de conflictos de lexemas únicos que se presenten durante la fase de clasificación de la categoría tal como lo realizaría un experto humano que tiene años trabajando en un proceso. El algoritmo deja un pequeño universo de lexemas en una lista para su clasificación humana después de realizar un análisis profundo de fuertes reglas de comparación de la Distancia de Levenshtein y no superaron el umbral establecido para categorizar.

Cuando un patrón es clasificado por un agente humano y lo aprueba, este se agregará a la lista oficial de patrones. Los patrones rechazados se descartan. De la misma manera, los lexemas en la lista negra que se apeguen a los nuevos patrones se eliminarán de la misma.

Trabajos futuros

El funcionamiento del algoritmo propuesto debe ser analizado después que la lista de patrones sea de un tamaño considerable. Se requiere desarrollar esta tarea para evaluar el desempeño del algoritmo de clasificación y medir la eficiencia de las reglas propuestas.

Para la categorización donde aún se requiera una intervención humana, se proponen dos líneas de investigación:

a) Desarrollar un análisis semántico de lexemas utilizando un acervo de diccionarios en línea para abordar la solución de las pequeñas listas que requieren de un proceso de clasificación humana.
b) Mejorar las estructuras de datos haciendo referencia a los algoritmos de expresiones regulares para la comparación de nuevos patrones que son categorizados de forma humana con los previamente clasificados.

BIBLIOGRAFÍA

Baeza-Yates., G. N. (2000). A guided tour to approximates string matching. . *ACM Computing Surveys*.

Erikson, B. (2011). *Sentiment Classification of Movie Reviews using Linguistic Parsing.* University of Wisconsin-Madison. EEUU.: Technical Report of the.

G.J.Klir, B. (1995). *Fuzzy Sets and Fuzzy Logic: Theory and Applications.* N.J., U.S.A: Prentice-Hall Inc.

Giarratano, J., & Riley, G. (2001). *Sistemas expertos principios y programación.* México: International Thomson Editores.

Heeringa, W. (2004). Measuring Dialect Pronunciation Differences. *Groningen Dissertations in Linguistics 46*.

K. Aas, L. E. (1999). *Text Categorisation: a Survey.* Norwegian Computing Center: Technical Report. Recuperado el 03 de 07 de 2015, de http://www.definicionabc.com/comunicacion/texto.php

Lewis., D. (1998). Naive (Bayes) at forty: The independence assumption in information. *In Proceedings of the 10th European Conference on Machine Learning.*

Martinez, R. G. (2004). *Ingenieria de Sistemas Expertos.* México: Nueva Libreria.

N. Castell, N. C. (1997). *Construcción Automática de Diccionarios de Patrones de Extracción de Información. Procesamiento del Lenguaje Natural. N° 21.*

Ozsoyoglu., T. B. (1997). Distance-based indexing for high-dimensional metric spaces. *ACM SIGMOD International Conference on Management of Data.*

Raya, A. M. (2006). *Introducción al análisis de datos difusos. .* Recuperado el 3 de Julio de 2015, de www.eumed.net/libros/2006b/amr/

Russell, S. (2004). *Inteligencia Artificial: Un enfoque moderno. .* Madrid España: Pearson Education.

Téllez, A. M. (2003). Clasificación automática de textos de desastres naturales en México. . *In Congreso Internacional en Investigaciones de Ciencias Computacionales.* México.