

Caracterización Automática de Perfiles de Usuarios basados en Consultas Lingüísticas Multigranulares usando un Algoritmo Genético Multiobjetivo

M. Luque

Dpto. Informática y A.N.,
Universidad de Córdoba,
14071-Córdoba (España)
mluque(at)uco.es

E. Herrera-Viedma, O. Cordon, A. Zafra

Dpto. Ciencia de la Computación e I.A.
Universidad de Granada
18071- Granada (España)
viedma,ocordon@decsai.ugr.es

A.G. López

Dpto. Informática
Universidad de Jaén
23071-Jaén (España)
aglopez@ujaen.es

Resumen

El objetivo de esta contribución es presentar un algoritmo genético multiobjetivo para derivar de manera automática consultas persistentes representadas como consultas lingüísticas multigranulares. Los experimentos desarrollados mostrarán que los perfiles derivados por nuestro sistema, aunque menos precisos que los obtenidos con un método clásico de generación de perfiles (representados como "bag of words"), son más flexibles, interpretables y expresivos.

Palabras Clave: Consultas Persistentes, Información Lingüística Multigranular, Algoritmos Genéticos, Recuperación de Información

1 Introducción

Las técnicas de Soft Computing se han revelado como una excelente herramienta en la personalización de los sistemas de recuperación de información (SRI) [1], dotándolos de flexibilidad y de algún tipo de "inteligencia" [4, 8]. Una de las formas de dotar de flexibilidad a un SRI es haciéndolo tolerante a la incertidumbre y la imprecisión –ambos inherentes en la interacción usuario-sistema– lo que permitirá a los usuarios representar sus necesidades de manera más natural [8]. Para llevar a cabo esto, se han propuesto lenguajes de consulta flexibles los cuales permiten expresar las necesidades de información subjetivas de forma simple y aproximada [2]. En concreto, el uso de información lingüística para definir los pesos de consultas Booleanas extendidas incrementa la comprensión de éstas y mejora la interacción SRI-usuario [15, 12].

Por otro lado, la capacidad de adaptación de los SRIs puede modelarse desde el punto de vista del aprendizaje automático, utilizando, entre otros, algoritmos

evolutivos. La aplicación de éstos a la RI ha generado resultados prometedores, como demuestra la gran cantidad de publicaciones al respecto [5]. La unión de estas técnicas y los lenguajes de consultas flexibles permite obtener SRIs "inteligentes" [4].

El objetivo de este trabajo es presentar un algoritmo genético (AG) multiobjetivo para generar de forma automática varias consultas lingüísticas multigranulares en una única ejecución a partir de un conjunto de documentos proporcionados por el usuario que representan sus necesidades de información. Las consultas derivadas serán usadas como consultas persistentes, consultas de RI que representan las necesidades de información, constantes durante un periodo de tiempo, de un usuario [9, 18]. Estas consultas se convertirán en perfiles de usuario para sistemas de búsqueda de información, permitiendo al usuario estar permanentemente informado sobre los tópicos que le interesan [10]; además, su estructura, más interpretable para el usuario, mejorará la representación clásica de perfiles de usuario.

La propuesta será validada en un entorno simulado de recuperación, considerando siete necesidades de información diferentes, todas ellas extraídas de la conocida colección Cranfield. Su rendimiento, tanto en eficacia de recuperación como en interpretabilidad de las consultas, será comparado con los perfiles de usuario derivados por uno de los mejores métodos clásicos para la generación de perfiles, VSR-OKAPI [9].

Con este propósito, la contribución está estructurada como sigue. La Sección 2 expone la necesidad de utilizar información lingüística multigranular. El AG multiobjetivo propuesto para aprender automáticamente consultas lingüísticas multigranulares se describe en la Sección 3. La Sección 4 presenta los experimentos desarrollados y el análisis de resultados, mientras que las conclusiones se recogen en la Sección 5.

mult-objective
an Journal of
0-71.

r, A tutorial
ochastic mul-
er Engineer-
ETH Zurich,
1.

he program-
ral selection,

ickles, and
l fuzzy logic
query opti-
-vance feed-
Shibata, T.
fic), 1997.

m retrieval
ic Publish-

The trans-
-formation

roduction
Graw-Hill,

and non-
York, NY:

of genetic
r text re-
nal of In-
-431.

retrieval
agement

val, But-

mparison
: Empi-
3 (2000),

3PEA2:
ry algo-
olution-
id Con-
ns (EU-
(Eds.),

Fuzzy

2 Información Lingüística Multigranular

Una cuestión importante en los SRIs es cómo facilitar la interacción usuario-sistema, incluso cuando la complejidad de los lenguajes difusos de consulta dificulta formular las consultas.

Es el subsistema de consulta el que soporta esta interacción usuario-SRI y, por lo tanto, debería tener en cuenta la imprecisión y la vaguedad típica de la comunicación humana. Este aspecto puede ser modelado introduciendo pesos numéricos o lingüísticos en el lenguaje de consulta usando para ello la Teoría de Conjuntos Difusos [2]. Parece más natural, caracterizar el contenido de los documentos asociando descriptores lingüísticos a los términos de una consulta, tales como "importante" o "muy importante", en vez de valores numéricos. En este sentido, el uso de SRI difuso-lingüísticos ha aumentado en los últimos años [15].

La mayoría de los SRIs lingüísticos suponen que los usuarios proporcionan sus necesidades de información por medio de consultas Booleanas cuyos términos se ponderan mediante valores lingüísticos representados por la variable lingüística "Importancia", valorada en el conjunto de etiquetas S . De igual forma, el valor de relevancia de los documentos (RSV), tras evaluar una consulta, se representa mediante la variable "Relevancia", también valorada sobre S . Usar el mismo conjunto de etiquetas para expresar la entrada y la salida de un SRI disminuye las posibilidades de comunicación en la interacción usuario-SRI. Además, puesto que ambas variables representan diferentes conceptos, parece lógico usar diferentes conjuntos de etiquetas para modelarlos, p.e., utilizar información lingüística multigranular [11], es decir, utilizar conjuntos de etiquetas con diferente semántica/granularidad para representar los diferentes tipos de información que aparecen en el proceso de recuperación.

Teniendo en cuenta lo anterior, en esta contribución, consideraremos una modificación de nuestra propuesta anterior de un SRI basado en información lingüística multigranular [13] como subsistema de evaluación para las consultas persistentes generadas por el AG que presentamos en la Sección 3.

La modificación consiste en utilizar funciones homogéneas [3] para modelar la semántica de importancia relativa en la evaluación de las subexpresiones. Esto nos permite eliminar la restricción de que una subexpresión tenga que estar compuesta de al menos dos términos para que tenga sentido aplicar dicha semántica. Así, conseguimos que el proceso de adaptación de las consultas, del conjunto de entrenamiento al de prueba, sea menos agresivo, lo que se

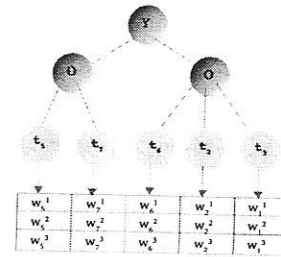


Figura 1: Consulta Booleana ponderada con pesos lingüísticos

espera que mejore el rendimiento del sistema.

Las características principales del SRI son: i) la ponderación de los términos de las consultas Booleanas y los RSV asociados a los documentos se expresan utilizando información lingüística multigranular; ii) cada término se pondera simultáneamente de acuerdo a tres semánticas: umbral simétrico, cuantitativa, importancia relativa; iii) los operadores Booleanos se modelan de manera flexible utilizando el operador OWA [21].

3 Un Algoritmo Genético Multiobjetivo para Aprender Consultas Persistentes Lingüísticas

En esta sección presentamos el AG multiobjetivo para aprender consultas lingüísticas multigranulares (perfiles de usuario expresivos), componente por componente.

3.1 Esquema de Codificación

La estructura de las consultas lingüísticas puede representarse como un árbol de expresión, donde los nodos terminales son los términos de la consulta y los nodos internos los operadores Booleanos. Además, cada nodo terminal tiene asociados tres pesos lingüísticos correspondientes a las tres semánticas indicada en la sección anterior. La Figura 1 muestra un ejemplo gráfico de este tipo de consultas. Lo natural sería codificar la consulta como un árbol y trabajar con Programación Genética [14] como se ha hecho en enfoques anteriores orientados al aprendizaje de consultas Booleanas [6, 19] o de consultas Booleanas extendidas (consultas difusas con pesos numéricos) [7, 16].

Sin embargo, el hecho de que nuestras consultas lingüísticas siempre estén en forma normal disyuntiva (DNF) o conjuntiva (CNF) nos permite tratar con una representación mucho más simple, sólo las subexpresiones de la consulta tienen que ser almacenadas (los operadores son siempre los mismos y no hace falta

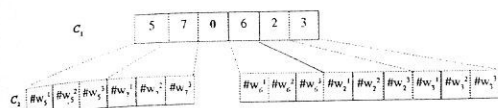


Figura 2: Estructura de un cromosoma

mantenerlos). Así, un cromosoma C que codifique una consulta lingüística estará compuesto de dos vectores de enteros:

1. C_1 , que codifica el árbol de consulta, donde el alelo 0 representa un separador entre subexpresiones y el resto de números representan los diferentes términos índice en la base de documentos. Los números negativos se asocian a los términos negados en la consulta.
2. C_2 , que codifica los pesos de los términos almacenando la etiqueta lingüística asociada como su posición dentro del conjunto de etiquetas.

Para ilustrar esto, la consulta de la Figura 1 se codificaría en el cromosoma de la Figura 2.

3.2 Generación de la Población Inicial

Todos los individuos de la población inicial son creados aleatoriamente, generando por separado ambas partes del cromosoma:

1. Las consultas (C_1) se compondrán de términos seleccionados de aquellos incluidos en el conjunto de documentos proporcionado por el usuario, teniendo mayor probabilidad aquellos que aparezcan en los documentos relevantes.
2. Los pesos (C_2) se calculan aleatoriamente, variando cada gen en su respectivo intervalo de definición: $\{1, \dots, \text{granularidad_conjunto_etiquetas}\}$.

3.3 Función de Evaluación

Los criterios clásicos de precisión y exhaustividad [1] – calculados como muestra la ecuación 1– se maximizan simultáneamente.

$$P = \frac{\sum_d r_d \cdot f_d}{\sum_d f_d} \quad ; \quad E = \frac{\sum_d r_d \cdot f_d}{\sum_d r_d} \quad (1)$$

donde $r_d \in \{0,1\}$ es la relevancia del documento d para el usuario, y $f_d \in \{0,1\}$ es la recuperación del documento d en el procesamiento de la consulta actual. Ha de tenerse en cuenta, que ambas medidas están definidas en el intervalo $[0,1]$, siendo 1 el valor óptimo.

¿Cuál es el conjunto de documentos recuperados?

El algoritmo evalúa cada consulta en el SRI Lingüístico indicado en la Sección 2, asociando a cada documento su correspondiente RSV. Los pasos que sigue son: i) preprocesamiento de la consulta para ponerla en DNF o CNF; ii) evaluación de los átomos respecto a la semántica de umbral simétrico; iii) evaluación de los átomos respecto a la semántica cuantitativa; iv) evaluación de las subexpresiones y modelado de la semántica de importancia relativa; y v) evaluación de la consulta completa.

3.4 Enfoque Multiobjetivo Considerado

Como se mencionó, el objetivo de nuestra propuesta es generar automáticamente varias consultas lingüísticas con diferente balance precisión-exhaustividad en una única ejecución. Para conseguir esto, hemos considerado el enfoque elitista de SPEA [22] como AG multiobjetivo. El esquema de selección implica los siguientes pasos:

1. La población intermedia se crea a partir de la población actual P y la población externa (P_e) por medio de torneo binario.
2. Los operadores genéticos se usan sobre los nuevos individuos para obtener una nueva población (P).
3. Las soluciones no dominadas existentes en la nueva población se copian a la población elitista P_e .
4. Las soluciones dominadas y duplicadas se eliminan.

De esta forma, la nueva población elitista está compuesta por las mejores soluciones no dominadas encontradas hasta el momento, incluyendo soluciones elitistas antiguas y nuevas. Para limitar el crecimiento de la población elitista, se restringe el tamaño a un número máximo de soluciones utilizando técnicas de clustering (ver [22] para más detalle).

3.5 Operadores Genéticos

La estrecha relación entre las dos partes de un cromosoma requiere operadores que trabajen de forma cooperativa en C_1 y C_2 , con el fin de sacar el máximo rendimiento a la representación elegida:

1. *Crucc*: Se emplean dos cruces diferentes en función de la estructura codificada por los padres:

- Si los dos padres codifican el mismo árbol de consulta (misma parte C_1), los dos descendientes se obtienen manteniendo los mismos valores de C_1 y aplicando un cruce en dos puntos sobre C_2 (pesos).
- De lo contrario, se aplica un operador de cruce simple en ambas partes del cromosoma seleccionando un punto de cruce aleatorio (pc) en C_1 para cada padre e intercambiando los genes existentes entre cp y el final de C_1 . En C_2 , el cruce se hace de la misma forma, utilizando los puntos de corte equivalentes.

2. *Mutación*: Se han utilizado siete operadores diferentes, seis que actúan sobre C_1 , y uno sobre C_2 . Los seis que actúan sobre C_1 son: cambio de un término por otro generado aleatoriamente, negación de un término, eliminación de un separador elegido aleatoriamente, adición de un separador en una posición aleatoria válida, desplazamiento de un separador y cambio de una subexpresión por otra generada aleatoriamente con más o menos términos (C_2 se actualiza automáticamente). Por otro lado, el operador de mutación sobre C_2 es similar al propuesto por Thrift en [20]. Cuando se realiza una mutación sobre un gen de C_2 , la etiqueta se cambia por la inmediatamente anterior o posterior. Si la etiqueta que se va a modificar es la primera o la última, se realiza el único cambio posible.

4 Experimentos y Análisis de Resultados

4.1 Experimentos Desarrollados

En esta sección mediremos el rendimiento del algoritmo MOGA-LPQ para el aprendizaje automático de perfiles de usuario representados como consultas lingüísticas persistentes. Como método de comparación, hemos elegido uno de los mejores métodos clásico para la generación de perfiles representados como "bag of words", VSR-OKAPI [9], basado en el modelo espacio vectorial y en la teoría de probabilidades [1]. Considera el Valor de Selección de Robertson como fórmula para la creación de los perfiles y la función OKAPI BM25 como función de similitud para el emparejamiento del perfil y los documentos.

Sin embargo, esta comparación no sería adecuada por dos razones. En primer lugar, estamos diseñando perfiles de usuario más expresivos, lo que normalmente lleva asociado un descenso en la eficacia de la recuperación (problema del balance entre interpretabilidad y exactitud). Por otro lado, nuestro algoritmo es capaz de derivar varias consultas lingüísticas (perfiles de

usuario) con diferente balance precisión-exhaustividad en una sola ejecución. Por lo tanto, la elección de una única consulta persistente para compararla con el perfil generado por VSR-OKAPI limita las posibilidades de nuestra propuesta. No obstante, y con el fin de conocer la competitividad del algoritmo presentado, se ha decidido llevar a cabo esta comparación.

La base de datos considerada para diseñar nuestros experimentos ha sido la colección clásica de Cranfield, compuesta de 1398 documentos sobre Aeronáutica. Estos documentos han sido automáticamente indizados de la forma usual, eliminando las palabras vacías, obteniendo un total de 3857 términos índice diferentes, e indizando de forma binaria para generar los términos relevantes en las representaciones de los documentos.

De entre las 225 consultas asociadas a Cranfield, se han seleccionado aquellas que presentan 20 o más documentos relevantes¹. Las siete consultas resultantes (números 1, 2, 23, 73, 157, 220 y 225) tienen asociados 29, 25, 33, 21, 40, 20, 25 documentos relevantes, respectivamente. Para cada una de estas consultas, la base de datos ha sido dividida aleatoriamente en dos conjuntos de documentos diferentes y no solapados, entrenamiento y prueba, cada uno de ellos compuesto de un 50% de documentos relevantes e irrelevantes para la consulta.

MOGA-LPQ ha sido ejecutado, para cada consulta, cinco veces con diferentes inicializaciones durante 50000 evaluaciones de la función de fitness en un ordenador AMD Turion64 MT30. Los valores considerados para los parámetros son un tamaño de población igual a 100 individuos, 25 individuos para la población elitista, un máximo de 10 términos para la consulta, y 0.8 y 0.2 para las probabilidades de cruce y mutación (para ambas partes, C_1 y C_2). El umbral de recuperación se ha establecido en la tercera etiqueta del conjunto. Por otro lado, VSR-OKAPI se ha ejecutado una única vez, puesto que no tiene componentes aleatorios, con un tamaño de perfil igual a 10 y con QTW (pesos de los términos) igual al Valor de Selección de Robertson.

Después de calcular las estadísticas individuales, los conjuntos Paretos obtenidos de las cinco ejecuciones de cada consulta se han fusionado y eliminado las soluciones dominadas. Las consultas con mejor exactitud sobre el conjunto de prueba ha sido seleccionada para

¹Nuestro método, basado en IQBE, genera una consulta a partir de un conjunto de documentos relevantes e irrelevantes. Por tanto, a) es necesario que haya un número suficientemente representativo de ejemplos positivos (documentos relevantes), y b) las ejecuciones son más largas y costosas que en relevance feedback. Por esas razones, reducimos el número de consultas consideradas y nos quedamos con las de mayor número de documentos relevantes, las más representativas.

de precisión-exhaustividad tanto, la elección de una para compararla con el per limita las posibilidades stante, y con el fin de algoritmo presentado, a comparación.

para diseñar nuestros n clásica de Cranfield, s sobre Aeronáutica. máticamente indiza, o las palabras vacías, os índice diferentes, generar los términos le los documentos.

las a Cranfield, se tan 20 o más doc- sultas resultantes 25) tienen asocia- entos relevantes, las stas consultas, la rriamente en dos y no solapados, ellos compuesto irrelevantes para

ada consulta, ones durante is en un orde- considerados blación igual oblación eli- sulta, y 0.8 ación (para cuperación conjunto. una única orios, con (pesos de ertson. ales, los uciones as solu- actitud a para

1 con- antes ra un posi- más s ra- is y itos

compararlas con la generadas por VSR-OKAPI.

4.2 Análisis de los Conjuntos Paretos Generados

Para medir la calidad de los conjuntos Paretos generados por MOGA-LPQ, se han considerado tres métricas clásicas: número diferente de soluciones no dominadas en el conjunto Pareto (en nuestro caso, número diferente de vectores objetivo (p.e., parejas precisión-exhaustividad)) ($\#d$), y las métricas $M_2^* \in [0, \#d]$ y $M_3^* \in [0, \sqrt{2}] = 1.4142$ [22]. La Tabla 1 recoge estos datos, mostrando el valor medio y la desviación estándar. En cada caso, cuanto mayor es el valor, mejor es la calidad del conjunto Pareto obtenido.

Tabla 1: Estadísticas de los conjuntos Pareto obtenidos por el algoritmo MOGA-LPQ

#q	#d	$\sigma_{\#d}$	M_2^*	$\sigma_{M_2^*}$	M_3^*	$\sigma_{M_3^*}$
1	6.2	0.335	5.680	0.325	1.062	0.032
2	5.6	0.537	5.400	0.456	1.097	0.046
23	9.6	0.669	8.676	0.540	1.199	0.010
73	5.4	0.537	5.400	0.537	1.086	0.054
157	11.2	0.867	10.127	0.813	1.213	0.011
220	4.6	0.358	4.600	0.358	1.022	0.073
225	7.2	0.522	7.019	0.459	1.203	0.035

A la vista de los resultados mostrados en la Tabla 1, el frente de los Pareto obtenidos son de buena calidad. Podemos observar como todas las ejecuciones generan al menos 5 consultas lingüísticas con diferente balance precisión-exhaustividad y con una desviación estándar de alrededor de 0.5. Los valores de M_2^* y M_3^* son muy apropiados también, mostrando una buena distribución del frente de los Paretos. Para ilustrar esto, la Figura 3 representa el frente unificado del Pareto obtenido para la consulta 157.

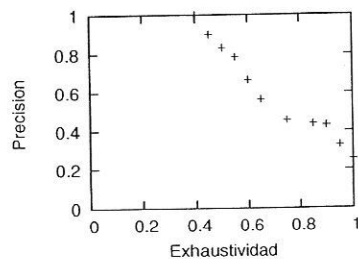


Figura 3: Frente del Pareto unificado generado por la consulta 157

4.3 MOGA-LPQ versus RSV-OKAPI

Para comparar ambos algoritmos, hemos elegido la precisión media a 11 niveles de exhaustividad (P_{avg})[1]

como medida de comparación, puesto que ambos algoritmos pueden devolver los documentos ordenados de acuerdo a su relevancia. Cuando se usa esta medida, el conjunto de documentos se ordena de acuerdo a su RSV de forma descendente, siendo interesante que los documentos relevantes estén al inicio del conjunto.

La Tabla 2 muestra la comparación entre la eficacia de recuperación de las mejores consultas sobre el conjunto de prueba, obtenidas por MOGA-LPQ y las correspondientes de VSR-OKAPI. En esta tabla, $\#rel$ representa el número de documentos relevantes asociados con la consulta, y $\#top$ el número de documentos relevantes situados en las primeras $\#rel$ posiciones del conjunto de documentos (ordenados por su RSV). En vista de los resultados, podemos observar como, sobre el conjunto de entrenamiento, nuestro método supera en seis de las siete consultas a VSR-OKAPI, además de generar perfiles más expresivos, como veremos más adelante. De hecho, la media global de los valores de P_{avg} para MOGA-LPQ es algo más alta que la de VSR-OKAPI (0.63 frente a 0.53). Además, el número de documentos relevantes en las primeras posiciones del conjunto de documentos recuperados es muy similar, con una pequeña ventaja en favor de nuestra propuesta.

Por otro lado, como era de esperar, RSV-OKAPI obtiene mejores resultados de prueba que MOGA-LPQ en cuatro de las siete consultas. Las diferencias son especialmente altas para las consultas 1 y 2. En este caso, la media global de los valores de P_{avg} de las siete consultas es 0.273 y 0.405 para MOGA-LPQ y VSR-OKAPI, respectivamente, lo que muestra claramente la existencia de sobreaprendizaje en nuestra propuesta. No obstante, aunque nuestra propuesta obtiene peores resultados que VSR-OKAPI sobre el conjunto de prueba, la representación de los perfiles de usuario como consultas persistentes mejora la expresividad de éstos, al aprenderse no sólo los términos que componen el perfil, sino también su estructura.

Finalmente, mencionar que los resultados obtenidos en esta experimentación mejoran a los obtenidos en [17]. Los algoritmos en ambos trabajos son idénticos, salvo por la modelización de la semántica de importancia relativa. En nuestra propuesta, utilizamos funciones homogéneas [3] para modelar esta semántica, consiguiendo que el proceso de adaptación de la consultas del conjunto de entrenamiento al conjunto de prueba sea menos agresivo que en [17].

5 Conclusiones

El uso de técnicas de soft computing para diseñar perfiles de usuario flexibles ha sido analizado, construyendo consultas lingüísticas por medio de un AG

Tabla 2: Perfiles de VSR-OKAPI versus Consultas lingüísticas de MOGA-LPQ

VSR-OKAPI							MOGA-LPQ						
#q	Entrenamiento			Prueba			#q	Entrenamiento			Prueba		
	P_{avg}	#rel	#top	P_{avg}	#rel	#top		P_{avg}	#rel	#top	P_{avg}	#rel	#top
1	0.510	14	6	0.484	15	6	1	0.649	14	9	0.074	15	2
2	0.522	12	5	0.789	13	9	2	0.585	12	6	0.296	13	6
23	0.475	16	7	0.357	17	7	23	0.746	16	12	0.385	17	7
73	0.616	10	5	0.268	11	2	73	0.632	10	4	0.405	11	4
157	0.532	20	10	0.298	20	7	157	0.610	20	9	0.245	20	5
220	0.725	10	6	0.544	10	5	220	0.641	10	6	0.411	10	2
225	0.370	12	5	0.099	13	2	225	0.535	12	6	0.100	13	2

multiobjetivo a partir de un conjunto de documentos extraído de la colección Cranfield. Como se ha comentado, nuestra propuesta es competitiva con el método clásico sobre el conjunto de entrenamiento, teniendo, además, la ventaja de generar perfiles más expresivos; mientras que el método clásico obtiene mejores resultados sobre el conjunto de prueba. No obstante, nuestra propuesta es prometedora ya que aprende tanto los términos del perfil como su estructura, además de generar un conjunto de consultas con diferente balance precisión-exhaustividad en una única ejecución. De todas formas, aún tenemos que refinarla para mejorar su capacidad de encontrar nueva información relevante para el usuario.

Agradecimientos

Este trabajo está soportado por el Ministerio de Ciencia y Tecnología en el marco de los proyectos TIC2003-07977 y TIC2003-00877, además de fondos FEDER.

Referencias

[1] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. Addison, 1999.

[2] G. Bordogna, P. Carrara, and G. Pasi. Fuzzy Approaches to Extend Boolean Information Retrieval. In P. Bosc and J. Kacprzyk, editors, *Fuzziness in Database Management Systems*, pages 231-274. Springer-Verlag, 1995.

[3] W. Cholewa. Aggregation of Fuzzy Opinions: An Axiomatic Approach. *Fuzzy Sets and Systems*, 17:249-259, 1985.

[4] O. Cerdón and E. Herrera-Viedma. Editorial: Special Issue on Soft Computing Applications to Intelligent Information Retrieval on the Internet. *International Journal of Approximate Reasoning*, 34(2-3):89-95, 2003.

[5] O. Cerdón, E. Herrera-Viedma, C. López-Pujalte, M. Luque, and C. Zarco. A Review of the Application of Evolutionary Computation to Information Retrieval. *International Journal of Approximate Reasoning*, 34:241-264, 2003.

[6] O. Cerdón, E. Herrera-Viedma, and M. Luque. Improving the Learning of Boolean Queries by means of a

Multiobjective IQBE Evolutionary Algorithm. *Information Processing and Management*, 42(3):615-632, 2006.

[7] O. Cerdón, E. Herrera-Viedma, M. Luque, F. Moya, and C. Zarco. Analyzing the Performance of a Multiobjective GA-P Algorithm for Learning Fuzzy Queries in a Machine Learning Environment. In *Lecture Notes in Artificial Intelligence 2715. Proc. of the 10th IFSA World Congress*, pages 611-615, Istanbul (Turkey), 2003.

[8] F. Crestani and G. Pasi, editors. *Soft Computing in Information Retrieval*. Physica-Verlag, 2000.

[9] W. Fan, M.D. Gordon, and P. Pathak. Effective Profiling of Consumer Information Retrieval Needs: A Unified Framework and Empirical Comparison. *Decision Support Systems*, 40(2):213-233, 2005.

[10] U. Hanani, B. Shapira, and P. Shoval. Information Filtering: Overview of Issues, Research and Systems. *User Modeling and User-Adapted Interaction*, 11:203-259, 2001.

[11] F. Herrera, E. Herrera-Viedma, and L. Martínez. A Fusion Approach for Managing Multi-Granularity Linguistic Term Sets in Decision Making. *Fuzzy Sets and Systems*, 114:43-58, 2000.

[12] E. Herrera-Viedma. Modeling the Retrieval Process for an Information Retrieval System using an Ordinal Fuzzy Linguistic Approach. *Journal of the American Society for Information Science and Technology*, 52(6):460-475, 2001.

[13] E. Herrera-Viedma, O. Cerdón, M. Luque, A. G. López, and A. M. Muñoz. A Model of Fuzzy Linguistic IRS Based on Multi-Granular Linguistic Information. *International Journal of Approximate Reasoning*, 34:221-239, 2003.

[14] J. Koza. *Genetic Programming. On the Programming of Computers by Means of Natural Selection*. The MIT Press, 1992.

[15] D.H. Kraft, G. Bordogna, and G. Pasi. An Extended Fuzzy Linguistic Approach to Generalize Boolean Information Retrieval. *Information Sciences*, 2:119-134, 1994.

[16] D.H. Kraft, F.E. Petry, B.P. Buckles, and T. Sadasivan. Genetic Algorithms for Query Optimization in Information Retrieval: Relevance Feedback. In

- E. Sanchez, T. Shibata, and L.A. Zadeh, editors, *Genetic Algorithms and Fuzzy Logic Systems*, pages 155–173. World Scientific, 1997.
- [17] M. Luque, O. Cordon, and E. Herrera-Viedma. A Multiobjective Genetic Algorithm for Linguistic Persistent Query Learning in Text Retrieval. In Y. Jin, editor, *Multi-objective Machine Learning*, pages 601–627. Springer, 2006.
- [18] D.W. Oard and G. Marchionini. A Conceptual Framework for Text Filtering. Technical Report CS-TR-3643, University of Maryland, College Park, 1996.
- [19] M.P. Smith and M. Smith. The Use of Genetic Programming to Build Boolean Queries for Text Retrieval through Relevance Feedback. *Journal of Information Science*, 23(6):423–431, 1997.
- [20] P. Thrift. Fuzzy Logic Synthesis with Genetic Algorithms. In *Proceedings of the Fourth International Conference on Genetic Algorithms*, pages 509–513, 1991.
- [21] R.R. Yager. On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decision Making. *IEEE Transactions on Systems, Man, and Cybernetics*, 18:183–190, 1988.
- [22] E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

s de 97	Aproximaciones simples a una superficie mediante multiresolución y TSK 197 Adolfo R. de Soto; Conrado A. Capdevila
..... 103	Hacia un cálculo fuzzy numérico/simbólico 205 Adolfo R. de Soto
..... 111	Uso de gráficos de cajas borrosos para controlar el final de una aplicación del método delphi 211 Sarabia, A., Maté, C., Caro, R. y J. Tapia
..... 117	An Evolutionary Fuzzy System for Scheduling in Wood-based Furniture Manufacturing . 217 Juan C. Vidal Aguiar, M. Mucientes, M. Lama and A. Bugarin
..... 123	Verificación de Taxímetros usando mediciones GPS como conjuntos borrosos 223 Jose R. Villar, Luciano Sanchez, Adolfo Otero, Jose Otero

Sesiones Especiales

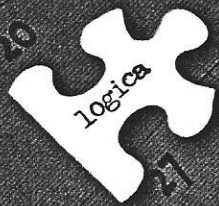
Soft Computing para la recuperación de información y minería en Web

.... 135	Organizadores: Enrique Herrera-Viedma, María J. Martín-Bautista, José Ángel Olivas, Alejandro Sobrino (Red Temática Nacional sobre Sistemas de Acceso a la Información en la Web basados en Soft Computing).
.... 141	Uso de modelos de restricción del contexto para el desarrollo de aplicaciones móviles inteligentes 229 Fernando Bobillo, Miguel Delgado, Juan Gómez-Romero.
... 147	Minería Web: Aplicaciones con Lógica Difusa 235 V.H. Escobar Jeria, M.J. Martín-Bautista, D. Sánchez, M.A. Vila.
.. 153	Un enfoque deductivo para la minería de textos 241 C. Justicia de la Torre, M.J. Martín-Bautista, D. Sánchez, M.A. Vila
e . 159	Aprendiendo consultas lingüísticas difusas con técnicas multiobjetivo. Estudio preliminar 247 A.G. López-Herrera, F. Herrera, E. Herrera-Viedma, A. Zafra, M. Luque.
165	Caracterización automática de perfiles de usuarios basados en consultas lingüísticas multigranulares usando un algoritmo genético multiobjetivo 253 M. Luque, E. Herrera-Viedma, O. Cordon, A. Zafra, A.G. Lopez.
171	Mejorando la búsqueda web mediante la adaptación de consultas en GUMSe 261 Javier de la Mata, José A. Olivas, Jesús Serrano.
179	Técnicas Difusas y No-Difusas para el Clustering del Contenido de Páginas en un Sitio Web 267 David Nettleton, Barbara Poblete
'85	Un sistema de recomendaciones lingüístico difuso para filtrar recursos en bibliotecas digitales universitarias 273 C. Porcel, A.G. Lopez, S. Alonso, M. Luque, E. Herrera-Viedma.
91	

Estylf 2006

XIII Congreso Español sobre Tecnologías y Lógica Fuzzy

Organización
Grupo de Investigación ORETO
Escuela Superior de Informática
Universidad de Castilla-La Mancha



ESTYLF 2006
Ciudad Real
20/22 Septiembre



Este libro de actas recoge todos los trabajos que han sido presentados como contribuciones al *XIII Congreso Español de Tecnologías y Lógica Fuzzy (ESTYLF)* de la *European Society for Fuzzy Logic and Technology (EUSFLAT)*.

El XIII Congreso ESTYLF ha sido organizado por el grupo de investigación Oreto, del departamento de Tecnologías y Sistemas de Información de la Universidad de Castilla-La Mancha, en las instalaciones de la Escuela Superior de Informática en Ciudad Real.

Desde estas líneas queremos agradecer a todos los que han contribuido a la elaboración de estas actas, tanto en su aspecto visual como en sus contenidos. De igual forma agradecemos la colaboración de las instituciones que han participado y con su apoyo han permitido la celebración de este congreso científico bianual.

Esperando que este encuentro en Ciudad Real sirva para mostrar el potencial que esta joven Universidad de Castilla-La Mancha es capaz de generar, así como ser unos dignos representantes del carácter acogedor de esta tierra.

Entidades Colaboradoras



Universidad de Castilla-La Mancha
*Vicerrectorado del Campus de
Ciudad Real y Extensión Universitaria
Vicerrectorado de Investigación*



Castilla-La Mancha

Consejería de
Educación y Ciencia



Escuela Superior de
Informática

Departamento de
Tecnologías y Sistemas
de Información

tsi

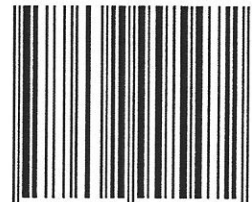


Diputación Provincial
de Ciudad Real

Ayuntamiento
de Ciudad Real



ISBN 84-689-9547-9



90000>



9 788468 995472

Editado por:

Grupo de Investigación Oreto.

Impreso en España por:

Artes Gráficas Lince

ISBN: 84-689-9547-9

N. REG: 06/50473