ACTAS de las

I Jornadas de Tratamiento y Recuperación de Información (JOTRI)

editores
Emilio Sanchis
Lidia Moreno
Isidoro Gil

entidad organizadora
Facultad de Informática

entidades colaboradoras
Universidad Politécnica de Valencia
Biblioteca Valenciana
Generalitat Valenciana
Escuela Universitaria de Informática
Departamento de Sistemas Informáticos y Computación
Biblioteca General (U.P.V.)

I Jornadas de Tratamiento y Recuperación de Información (JOTRI)

editores
Emilio Sanchis
Lidia Moreno
Isidoro Gil

Facultad de Informática
Valencia, 4 y 5 de Julio de 2002

Universidad Politécnica de Valencia
Editorial UPV **Ref.: 2000.2432**

Edita: EDITORIAL DE LA UPV
Camino de Vera, s/n
46071 VALENCIA
Tel.96-387 70 12
Fax 96-387 79 12

Imprime: REPROVAL, S.L.
Tel.96-369 22 72

Depósito Legal: V-2499-2002
I.S.B.N.: 84-9705-199-8

Comité organizador (Universidad Politécnica de Valencia)

Emilio Sanchis Arnal (Co-presidente)
Lidia Moreno Boronat (Co-presidente)
Isidoro Gil Leiva (Secretario)
Pedro Blesa Pons
Lluis Hurtado Oliver
Antonio Molina Marco
Ferran Pla Santamaria
Natividad Prieto Saez

Comité de programa

Iñaki Alegría Loinaz, Universidad del Pais Vasco
Manuel de Buenaga Rodríguez, Universidad Europea
Antonio Ferrández Rodríguez, Universidad de Alicante
Vicente Guerrero Bote, Universidad de Extremadura
Isidoro Gil Leiva, Universidad Politécnica de Valencia
José Antonio Moreiro González, Universidad Carlos III de Madrid
Lidia Moreno Boronat, Universidad Politécnica de Valencia
Manuel Palomar Sanz, Universidad de Alicante
Horacio Rodríguez Hontoria, Universidad Politécnica de Cataluña
José Vicente Rodríguez Muñoz, Universidad de Murcia
Emilio Sanchis Arnal, Universidad Politécnica de Valencia
Encarna Segarra Soriano, Universidad Politécnica de Valencia
Alfonso Ureña López, Universidad de Jaén
Felisa Verdejo Maillo, Universidad Nacional a Distancia

Revisores

Iñaki Alegría
Manuel de Buenaga
Arantza Casillas
María José Castro
Arantza Diaz de Ilarraza
Antonio Ferrández
Isidoro Gil
Julio Gonzalo
Vicente P. Guerrero
J. J. Merelo
José A. Moreiro
Lidia Moreno
Manuel Palomar
Horacio Rodríguez
Emilio Sanchis
Encarna Segarra
L. Alfonso Ureña
M. Felisa Verdejo

Índice

Aplicación de redes neuronales y redes bayesianas en la detección de multipalabras para tareas IR.

F. Martínez, M.C. Díaz, M.T. Martín, V.M. Rivas, L.A. Ureña
{dofer, mediaz, maite, vrivas, laurena}@ujaen.es
Universidad de Jaén
España

Resumen: En este artículo, se compara el uso de dos métodos distintos para detectar si una pareja de términos son o no multipalabras. Por un lado se usa una red neuronal para clasificar dichos bigramas, y por otro, una red bayesiana para obtener la confianza en que los bigramas sean multipalabras. La clasificación está basada en diferentes estimadores, actualmente disponibles en la literatura, usados como entradas a las dos redes. El resultado obtenido en esta clasificación ha sido usado en tareas de recuperación de información. Los experimentos demuestran que los dos métodos mejoran la precisión alcanzada por un sistema IR, y entre ellos es la red bayesiana la que mejores resultados ofrece.

1 Introducción

En este artículo presentamos una nueva manera de resolver el problema de la detección de multipalabras exógenas. Una multipalabra exógena, frente a las conocidas como endógenas, es una sucesión de términos cuyo significado es distinto a la suma de significados de dichos términos. Por lo tanto, una multipalabra exógena, de aquí en adelante multipalabra, puede ser considerada como un nuevo concepto.

Otros investigadores han probado, con un éxito moderado, que la detección de multipalabras mejoran las tareas tradicionales de Recuperación de Información. Además, David Hull y Gregory Grefenstette [5] demostraron que la correcta traducción de multipalabras mejoraban la precisión de los sistemas de Recuperación de Información Multilingües (CLIR, *Cross Language Information Retrieval*) hasta en un 40%.

La nueva propuesta que realizamos en este artículo usa dos métodos distintos para encontrar parejas de términos que son multipalabras. El primero de ellos utiliza una red neuronal supervisada basada en el modelo de Kohonen [8]: Se trata del aprendizaje por cuantificación vectorial (Learning Vector Quantization - LVQ), ampliamente usada para tareas de clasificación [9]. La entrada de la red son los valores dados por estimadores utilizados en la literatura para realizar esta misma tarea, y la salida que da la red neuronal es un clase que determina si los valores dados pertenecen a una multipalabra o no. El aprendizaje de la red se realiza mediante el entrenamiento de ésta con valores dados por los citados estimadores para parejas de términos de las cuales conocemos si son o no multipalabras.

El segundo método estudiado utiliza una red bayesiana. Esta toma como entrada los mismos estimadores utilizados para la red neuronal, y nos devuelve una valor que indica la confianza de que la pareja de términos sea una multipalabra. Por lo tanto, al contrario que la red neuronal, este método obtiene una lista de multipalabras y su valores, sobre los cuales hay que aplicar un valor umbral para obtener así una lista de verdaderas multipalabras.

Para probar la eficacia de estos métodos, las redes (neuronal y bayesiana) han sido aplicadas para detectar multipalabras en un corpus. Posteriormente, se han lanzado un conjunto de consultas sobre tal corpus. Por lo tanto, la calidad de nuestro enfoque ha sido medida en términos de mejora de precisión en la tarea IR.

El resto del artículo está organizado de la siguientes manera: en el apartado 2 se da una pequeña introducción del estado del arte, mostrando brevemente algunos métodos usados para detectar multipalabras. Estos métodos incluyen diferentes estimadores que serán utilizados más tarde en nuestros métodos. El apartado 2.1 describe un nuevo estimador desarrollado para ampliar el espectro de multipalabras detectadas. El apartado 3 muestra los experimentos llevados a cabo y los resultados obtenidos. Finalmente, en el apartado 4 se exponen las conclusiones obtenidas y las futuras líneas de investigación.

2 Un nuevo enfoque

Normalmente los métodos para la automatización terminológica han sido, tradicionalmente, estadísticos [5] [2], basados en la coocurrencia de cada pareja de palabras dentro del corpus. Otros trabajos [1] obtienen el grado de similitud entre términos usando el factor de coocurrencia y la fórmula de pesado estándar *tf*idf*. Recientemente, se han desarrollado enfoques híbridos incorporando información lingüística: Diana Maynard y Sophia Ananiadou [10] utilizan diferentes tipos de información contextual: sintáctica, semántica, terminológica y estadística. Sin embargo, la integración de distintas fuentes de información debe realizarse de alguna manera. La forma más sencilla es usando una función lineal, aunque esto no significa que sea la mejor manera de abordar este problema.

Para cualquiera de las características (sintácticas, semánticas, terminológica y estadísticas) que queramos integrar para mejorar la detección de multipalabras, existen estimadores que ya se han utilizado con buenos resultados. Nosotros hemos seleccionado cuatro de ellos, más uno propio, procurando tener un conjunto heterogéneo y representativo de los diversos métodos ya conocidos. Los estimadores que se han usado en este trabajo son:

1. χ^2 *de Pearson*. Una variante del coeficiente estadístico χ^2 [5].
2. Medida de la importancia de coocurrencia de los elementos de un conjunto mediante la métrica *em* [2].
3. *Coeficiente de similitud de Dice*, que obtiene el grado de similitud o asociación entre términos usando una medida de similitud de conjuntos [1].
4. El *ratio de información mutua*, o ratio de asociación, μ [6].
5. Finalmente, se ha desarrollado un nuevo estimador, una variante del coeficiente de similitud de Dice basado en el índice de Simpson.

2.1 Un nuevo estimador: el coeficiente de Similitud de Simpson

Aproximadamente, el índice Dice está basado en la asociación entre dos términos, calculando el coeficiente de intersección de los dos conjuntos y su unión. Normalmente, esta aproximación es suficiente para estimar la correlación entre dos palabras, pero no siempre. "Bill Clinton" por ejemplo, es un multipalabra, pero "Bill" es una palabra muy común, por lo que su frecuencia es muy alta, y el conjunto "Bill" es enorme. "Clinton" no es tan frecuente, por lo tanto el conjunto "Clinton" es pequeño. De ese modo, el coeficiente de intersección y la unión de ambos conjuntos sería pequeño, porque el conjunto "Clinton" es pequeño. Por otro lado, el índice Simpson estima la asociación entre dos conjuntos calculando el coeficiente de intersección de los dos conjuntos y el más pequeño de ellos, por lo tanto "Bill Clinton" alcanzaría un valor alto para el coeficiente Simpson, y un valor bajo para el coeficiente Dice

$$DICE:\ xy = 2\frac{\sum_{i=1}^{n}(w'_{ix}\cdot w'_{iy})}{\sum_{i=1}^{n} w'^{2}_{ix} + \sum_{i=1}^{n} w'^{2}_{iy}}$$

$$SIMPSON:\ xy = 2\frac{\sum_{i=1}^{n}(w'_{ix}\cdot w'_{iy})}{\min\left(\sum_{i=1}^{n} w'^{2}_{ix}, \sum_{i=1}^{n} w'^{2}_{iy}\right)}$$

donde:

w_{ix} = peso del término x en el documento i.

w_{iy} = peso del término y en el documento i.

$w'_{iy} = w_{iy}$ si el término y también aparece en el documento i, o 0 en otro caso.

$w'_{ix} = w_{ix}$ si el término x también aparece en el documento i, o 0 en otro caso.

n = número de documentos en la colección.

2.2 Red Neuronal: el algoritmo LVQ

El algoritmo LVQ es un método de clasificación basado en el aprendizaje competitivo neuronal, el cual permite definir un grupo de categorías en el espacio de los datos de entrada para reforzar el aprendizaje, o positivo (premio) o negativo (castigo). El algoritmo LVQ usa un aprendizaje supervisado para definir regiones de clases en el espacio de los datos de entrada. Con este propósito, un subconjunto de vectores de similitud etiquetados de forma similar forman una región de una clase. A estos vectores se les denomina vectores de pesos, vectores de referencia o codebooks. A cada clase se le asocia un conjunto de vectores de pesos w_k. Cada vector de pesos w_k está etiquetado con una clase, de manera que durante el proceso de aprendizaje, uno de ellos será seleccionado y la clase a la que pertenece será elegida como ganadora de la competición.

El algoritmo LVQ trabaja de la siguiente manera: En cada iteración, el algoritmo selecciona un vector de entrada, x_i, y lo compara con cada vector de pesos, w_k, usando alguna medida de similitud (en nuestro caso, concretamente se ha utilizado la distancia euclídea $\|x_i, w_k\|$); el vector w_c será el ganador si es el más cercano a x_i, por lo que c será la clase asignada:

$$\|x_i - w_c\| = \min_k\{\|x_i - w_k\|\}$$

Las clases compiten entre ellas para encontrar el vector de entrada más parecido, para que el ganador sea el que menor distancia euclídea tenga respecto al vector de entrada. Sólo la clase ganadora podrá modificar el vector de pesos usando un algoritmo de aprendizaje reforzado, o positivo o negativo, dependiendo de que la clasificación sea correcta o no. De este modo, si la clase ganadora pertenece a la misma clase que el vector de entrada (la clasificación ha sido correcta), se incrementará el peso, acercándose ligeramente al vector de entrada (premio). Por el contrario, si la clase ganadora es diferente a la clase del vector de entrada (la clasificación no ha sido correcta), se decrementará el peso, alejándose ligeramente del vector de entrada (castigo).

Sea $x_i(t)$ un vector de entrada en el tiempo t, y $w_k(t)$ el vector de pesos para la clase k en el tiempo t. La siguiente ecuación define el proceso de aprendizaje básico para el algoritmo LVQ.

$$w_c(t+1) = w_c(t) + s\cdot\alpha(t)\cdot\left(x_i(t) - w_c(t)\right)$$

donde $s = 0$, si $k \neq c$; $s = 1$, si $x_i(t)$ y $w_c(t)$ pertenecen a la misma clase; y $s = -1$, si no lo son, y donde $\alpha(t)$ es el ratio de aprendizaje, siendo $0<\alpha(t)<1$, una función monótona decreciente del tiempo. Se recomienda que $\alpha(t)$ sea más bien pequeña inicialmente, es decir, menor de 0,5, y que decrezca hasta un umbral dado, u, muy cercano a 0 [9].

Los experimentos mostrados en la sección 3, fueron llevados a cabo usando la implementación descrita en la documentación de LVQ_PAK [7] con los parámetros por defecto. Así, cada experimento se inicia con el mismo número de vectores de pesos por clase (10 para la clase 0 y 10 para la clase 1) y el ratio de aprendizaje se inicializa a 0,3.

2.3 Red bayesiana

Para integrar los estimadores mostrados anteriormente también hemos usado de una red bayesiana [4]. Una red bayesiana es un grafo dirigido acíclico, donde los nodos representan las variables del problema que se desea resolver. El conocimiento del problema se representa mediante la instanciación de aquellos nodos cuyo valor es conocido, propagándose tal conocimiento a través de la red mediante ciertas reglas probabilísticas. Así, consideramos que los cinco estimadores antes enumerados son indicios de que efectivamente estamos ante una multipalabra. Una vez conocido el valor alcanzado por cada uno de estos nodos para una candidata dada, la red bayesiana propagará tal información hacia el resto de la red, que en nuestro caso se corresponde con un único nodo que representa nuestra creencia en que un determinado bigrama sea o no una multipalabra. En la Figura 1 puede apreciarse el aspecto de la sencilla red bayesiana que hemos utilizado en nuestros experimentos. Existirá una de estas redes bayesianas para cada candidata a multipalabra que consideremos.

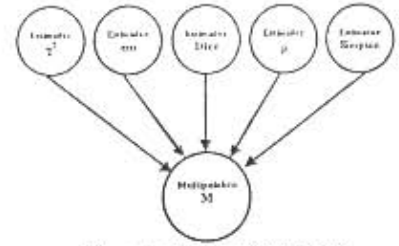

Figura 1. Red bayesiana para el reconocimiento de multipalabras

El significado de cada nodo es el siguiente:

- La variable *multipalabra M*, que puede tomar dos valores, verdadero o falso, representan nuestra creencia de que M es una multipalabra o no. Es el valor a inferir.
- Los nodos *estimador X* representan nuestra confianza en ese estimador. ¿Cómo de probable es que el *estimador X* afirme que M es una multipalabra?. Esto es, tenemos que conocer la siguiente matriz de probabilidades:
  i. P(estimador_X=K/M=true).
  Cómo de probable es que el *estimador X* afirme que M es una multipalabra, supuesto que M es una multipalabra.

ii. P(estimador X=K/M=true)
Cómo de probable es que el *estimador X* afirme que M es una multipalabra, supuesto que M es una multipalabra.

iii. P(estimador_X=K/M=false).
Cómo de probable es que el *estimador X* afirme que M es una multipalabra, supuesto que M no es una multipalabra.

iv. P(estimador_X=K/M=false)
Cómo de probable es que el *estimador X* niegue que M es una multipalabra, supuesto que M no es una multipalabra.

Dónde K es un valor entero entre 1 y 5. Esto es debido a que los diferentes estimadores devuelven valores continuos, lo cual complica la implementación de la red bayesiana. Así, hemos establecido cinco intervalos para cada estimador, convirtiéndolos en una variable discreta.

Los casos i y iv representan la precisión del estimador, y son las probabilidades a maximizar. Estas probabilidades pueden calcularse mediante un entrenamiento previo, para el cual debemos tener dos listas, una de multipalabras y otra de no multipalabras, le aplicamos el estimador que estamos evaluando a cada elemento de ambas listas. De esta manera sabremos en cuantas ocasiones acierta y en cuantas falla en su diagnóstico.

Esta sería la fase de *evaluación* del estimador. Una vez que hemos evaluado nuestros estimadores, podemos crear la red bayesiana tal como aparece en la figura 1.

Debido a que la red bayesiana nos proporciona un valor numérico y no una clase, tal y como lo hace la red neuronal, podemos utilizar un umbral para discernir entre aquellas palabras que son multipalabras y las que no.

3 Experimentos y resultados

Para poder entrenar y evaluar las redes (neuronal y bayesiana), se construyeron dos listas de ejemplos. Una primera formada por multipalabras, y otra por bigramas escogidos aleatoriamente. Cada ejemplo corresponde a una pareja de términos, junto con los valores alcanzados por tal pareja para cada estimador.

Para obtener la lista de multipalabras hemos recurrido a WordNet [11], una base de datos léxica rica en multipalabras, tanto endógenas como exógenas. Sin embargo, tan sólo las exógenas son objeto de nuestro estudio. Por esta razón, cada multipalabra extraída de WordNet, fue nuevamente buscada en la versión en línea de la enciclopedia Encarta[1], con la finalidad de eliminar multipalabras poco frecuentes. Finalmente, la lista así obtenida fue filtrada por los autores de este artículo, eliminando multipalabras de clara naturaleza endógena.

La lista de no multipalabras (necesaria para entrenar la red) fue obtenida del corpus usado en el CLEF 2000[2]. Las parejas de palabras se tomaron de este corpus, eliminando aquellas que aparecían en WordNet o en Encarta.

Se crearon ambas listas, multipalabras y no multipalabras, se aplicaron los estimadores citados anteriormente, y se obtuvo un fichero que fue utilizado tanto para entrenar la red neuronal como la red bayesiana.

La colección de Los Angeles Times 1994, obtenida de la colección en inglés del CLEF 2000, se usó para computar la incidencia de la localización de multipalabras en tareas de Recuperación de Información (IR). Esta colección está compuesta por 113.005 artículos de la edición de 1994 de Los Angeles Times, y 40 consultas (Título + Descripción + Narrativa) con sus juicios de relevancia. La colección fue indexada utilizando el software *Zprise*[3], con la fórmula de pesado Okapi [13].Con la finalidad de medir la calidad de la red neuronal y la red bayesiana como herramientas de integración de los estimadores, hemos aplicado las multipalabras obtenidas a la tarea de I.R.: hemos construido cinco índices para una colección de 113.005 documentos correspondiente al anuario de Los Angeles Times 1994:

i. Índice sin multipalabras. Es el caso base, en el cual se ha usado como unidad de indexación la palabra.

ii. Índice con multipalabras detectadas por la Red Neuronal. La unidad de indexación usada son palabras y multipalabras encontradas por la Red Neuronal.

iii. Índice con multipalabras detectadas por la Red Bayesiana, con un valor de corte de 0.95. Igual que el caso ii, pero usando la Red Bayesiana descrita en el anterior apartado. Se consideran multipalabras tan sólo aquellas en las que la Red Bayesiana alcance una confianza igual o superior a 0.95

iv. Índice con multipalabras detectadas por la Red Bayesiana, con un valor de corte de 0.90.

v. Índice con multipalabras detectadas por la Red Bayesiana, con un valor de corte de 0.75.

[1] Encarta está disponible en http://www.encarta.com [2/2/2002]. Se ha usado Encarta porque incluye nombres propios son considerados multipalabras.

[2] Cross Language Evaluation Forum (CLEF) promueve la investigación y el desarrollo de tareas CLIR. Para más información, véase: http://www.clef-campaign.org

[3] ZPrise es un software desarrollado por el NIST. Está disponible en: http://www.itl.nist.gov/iaui/894.02/works/papers/zp2/zp2.html [2/2/2002]

Sobre cada uno de estos índices, se han aplicado cuarenta consultas, junto con sus juicios de relevancia, provenientes de las jornadas CLEF[4] del año 2000. En las consultas se han marcado las multipalabras que proceden en cada caso. La precisión obtenida se detalla en la Tabla 1:

Índice	*Precisión media*
Sin multipalabras	0.375
Red Neuronal	0.390
Red Bayesiana con una confianza de 0.95	0.427
Red Bayesiana con una confianza de 0.90	0.410
Red Bayesiana con una confianza de 0.75	0.356

Tabla 1. Precisión media obtenida.

De la Tabla 1 se desprende que la Red Bayesiana obtiene una mejora sensiblemente superior sobre la Red Neuronal y el caso base (no usar multipalabras). Esto puede ser debido a que una red bayesiana nos permite conocer su grado de creencia en el hecho inferido. Así, podemos marcar como multipalabras sólo aquellas expresiones en las que nuestra confianza sea muy alta. Esto es, estamos primando la precisión sobre la cobertura en la tarea de detección de multipalabras. Como se puede observar, al bajar nuestro grado de confianza hasta el 0.75, los resultados obtenidos son peores que el caso base, reforzando así la hipótesis de que marcar multipalabras que realmente no lo son es muy contraproducente. Como apoyo a esta hipótesis, a continuación mostramos un estudio pormenorizado de cuatro consultas, con las multipalabras marcadas según la red neuronal.

El análisis más detallado de los resultados nos lleva a concluir que la detección de multipalabras es útil para las tareas de IR. La Tabla 2 muestra la precisión alcanzada por algunas consultas, y las multipalabras detectadas en cada una de ellas utilizando la red neuronal.

Consulta	*AvgP. Original*	*AvgP. con multi-palabras*	*Multi-palabras detectadas*
#7	0,2969	0,4452	"world soccer"
#9	0,1022	0,2027	"war ii", "ii war", "war rwanda", "world war"
#3	0,3913	0,3220	"decisions made", "hard soft"
#32	0,4126	0,2511	"women priest", "change direction"

Tabla 2. Cuatro consultas detalladas.

Como la Tabla 2 muestra, la consulta #7 gana un 5% de precisión absoluta porque "world soccer" fue correctamente detectada como multipalabra. Los resultadoas de la consulta #9 fueron igual de buenos, porque se detectaron correctamente las multipalabras "world war", "war rwanda" "war ii" and "ii war". Como puede verse, la precisión obtenida en estas consultas mediante los nuevos métodos es el doble de la precisión obtenida sin detectar multipalabras.

Por otro lado, la consulta #3 pierde un 7% de precisión con la inclusión de multipalabras. Los bigramas "decisions made" y "hard soft" son de hecho no multipalabras, pero la red neuronal a marcado ambas como multipalabras. Finalmente, la consulta #32 pierde un 16% de precisión, ya que, "women priest" y "change direction" tampoco son multipalabras.

[4] CLEF: CLIR Evaluation Forum, es una competición similar al TREC, pero de ámbito multilingüe. Para más información: http://clef.iei.pi.cnr.it:2002

4 Conclusiones y trabajos futuros

En este artículo presentamos dos nuevos métodos para detectar multipalabras. Estos métodos usan los valores obtenidos por varios estimadores, presentados en la literatura y desarrollados para mejorar esta misma tarea, como entrada de las redes utilizadas que automáticamente determinan si estos valores pertenecen a una verdadera multipalabra o simplemente a una pareja de términos que aparecen juntos en el documento.

Los resultados muestran que la detección automática de multipalabras es útil para IR. Sin embargo, los métodos usados deberían tener una mayor exactitud, porque una mala detección de multipalabras daña la precisión del sistema IR. Por lo tanto, se deben usar métodos más cautelosos para encontrar multipalabras. Clasificar multipalabras como no multipalabras es mejor que reconocer demasiadas multipalabras. Es decir, la detección de multipalabras debe mejorar la precisión en lugar de la cobertura.

Futuras líneas de investigación incluirán el uso de nuevas redes neuronales, como la Radial Basis Function Nets [3] [12], así como las RCE [15], y también redes de entrenamiento no supervisado como las redes de Mapas Auto-organizativo [9].

Se usarán nuevos estimadores basados en información semántica para mejorar los resultados. También se investigarán otras aplicaciones de estos métodos, especialmente la influencia en Recuperación de Información Multilingüe.

Referencias

[1] Adriani, M. and C.J. van Rijsbergen. Term Similarity Based Query Expansion for Cross Language Information Retrieval. In *Proceedings of Research and Advanced Technology for Digital Libraries*, Third European Conference (ECDL'99), p. 311-322. Springer Verlag: Paris, September 1999.

[2] Ballesteros, L y Croft, W.B. Resolving ambiguity for cross-language retrieval. In: Croft, W.B., Moffat, A., van Rijsbergen, C.J., Wilkinson, R. and Zobel, J. eds. Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, NY: ACM, 1998, 64—71.

[3] D.S. Broomhead, D.Lowe. Multivariable Functional Interpolation and Adaptative Networks. In Complex Systems, vol. 11, pp.321-355., 1988

[4] L. Castillo, J.M. Gutierrez, and A.S. Hadi, Sistemas expertos y modelos de redes probabilísticas. Academia de Ingeniería, 1996.

[5] David A. Hull, Gregory Grefenstette. Experiments in Multilingual Information Retrieval. In Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1996.

[6] Christer Johansson. Good Bigrams. In Proceedings from the 16th International Conference on Computational Linguistics (COLING-96). Copenhagen: 592-597. 1996

[7] T. Kohonen, J. Hynninen,J. Kangas, J. Laaksonen, K. Torkkola. LVQ PAK: The Learning Vector Quantization program package. Helsinki University of Technology Laboratory of Computer and Information Science. Finland, 1991-1995.

[8] T. Kohonen, J. Kangas, J. Laaksonen, K. Torkkola. LVQ_PAK: A program package for the correct application of Learning Vector Quantization algorithms. In Proceedings of the International Joint Conference on Neuronal Networks, Pages I 725-730, Baltimore, June 1992. IEEE.

[9] T. Kohonen, Self-Organization and Associative Memory. 2nd Ed. Springer-Verlag, Berlin, 1995.

[10] Diana Maynard and Sophia Ananiadou. TRUCKS: a model for automatic term recognition, Journal of Natural Language Processing, December 2000.

[11] G. Miller. WORDNET: A lexical database for English. Communications of the ACM, 38 (11), 1995.

[12] V.M. Rivas, J.J. Merelo, P.A. Castillo. Evolving RBF Neural Networks. Lecture Notes in Computer Science, vol. 2064, pp.506-513. 2001

[13] Robertson, S. E., Walker, S. & Beaulieu, M. Experimentation as a way of life: Okapi at TREC. Information Processing & Management, 36(1), 95-108. 2000

[14] Salton, Gerard, and McGill, Michael J. Introduction to Modern Information Retrieval, New York: McGraw-Hill, 1983.

[15] Zboril, F. Zboril,F. The use of the RCE network in a Pattern Recognition. Proceedings of MOSIS 2000, pp. 65-70. 2000