Sobre la factibilidad de detectar consultas en ráfaga utilizando redes neuronales artificiales

Rodrigo Henríquez*, Francisco Cruz†, Carlos Gómez-Pantoja‡,

*Universidad de Santiago de Chile, Chile, rodrigo.henriquez@usach.cl,
†Universidad Andrés Bello, Chile, fcruz.cl@gmail.com,
‡Universidad Andrés Bello, Chile, carlos.gomez.pantoja@unab.cl

Palabras Clave: Motor de Búsqueda Web, Consulta en Ráfaga, Redes Neuronales Artificiales.

Resumen

Los motores de búsqueda Web deben lidiar con el tráfico de consultas de usuario que es variable e impredecible. Como las consultas siguen distribuciones zipfianas, un grupo muy reducido de consultas tiene un alto impacto en el desempeño del motor de búsqueda Web. Algunas veces el usuario es atraído por eventos sociales, económicos y naturales, generando un gran volumen de peticiones en torno a un conjunto reducido de consultas en un período de tiempo muy breve. Estas consultas se denominan *consultas en ráfaga*. Este trabajo está dedicado a resolver este problema usando redes neuronales artificiales para predecir la aparición de esta clase de consultas. Esta solución ayuda a tomar soluciones proactivas antes que estas consultas sobrecarguen el motor de búsqueda Web.

1 Introducción

Hoy en día, los Motores de Búsqueda Web (*Web Search Engines*, WSEs) son herramientas importantes para varios aspectos de nuestras vidas. Cada vez que estamos interesados en algún tópico, explotamos la gran cantidad de datos almacenados e indexados por los WSEs. La idea es hacer una consulta al WSE y éste nos da un conjunto de documentos relevantes a la consulta [1].

Los actuales WSEs son dimensionados de tal forma que puedan hacer frente a cientos de miles de consultas por segundo [2] y miles de usuarios concurrentes [3]. A pesar de este hecho, existen algunas situaciones donde el WSE puede sobrecargarse, o incluso colapsar, debido a una alta cantidad de consultas por unidad de tiempo.

Cuando los usuarios, o la población mundial, se sienten atraídos por un evento social, económico o natural específico (por ejemplo, un terremoto, situaciones que están relacionadas con celebridades, etc.), los usuarios comienzan a requerir información al WSE acerca de tal evento y sus tópicos relacionados. Por ejemplo, cuando una celebridad muere, los usuarios se interesan en sus fotografías, biografías, su información personal, etc. Entonces, el tópico emergente está compuesto de varias consultas. A estas consultas las denominamos como consultas en ráfaga (*bursty queries*).

Por otro lado, el WSE está compuesto por servicios desplegados en clusters de computadores conectados por una red de alta velocidad [4]. Uno de esos servicios, el servicio de cache, es susceptible a desbalance cuando las consultas en ráfaga aparecen. El desbalance incrementa el tiempo de respuesta de las consultas y la probabilidad de servidores congestionados. Detectar consultas en ráfaga puede ayudar a los diseñadores del WSE a tomar acciones proactivas ante la aparición de este tipo de consultas. Entre estas acciones está la distribución en más nodos de este tipo de consultas.

Este trabajo es el primer intento en analizar la factibilidad de usar redes neuronales artificiales para detectar la aparición de consultas en ráfaga. Para este fin, analizamos el tráfico de un WSE comercial, y analizamos las más frecuentes para entrenar la red neuronal artificial. Este proceso será descrito más adelante.

La siguiente sección presenta la arquitectura del WSE utilizada. La sección 3 explica el marco teórico de las redes neuronales. Luego, la sección 4 presenta el desarrollo experimental. Finalmente, la sección 5 da las conclusiones de este trabajo.

2 Motor de Búsqueda Web (WSE)

Para hacer frente al actual volumen de consultas por unidad de tiempo y la cantidad de usuarios concurrentes, el WSE divide el procesamiento de consultas en funcionalidades bien definidas. Cada una de estas funcionalidades es implementada como un servicio. Los principales servicios son: servicio de Front-End (FS), servicio de Cache (CS) y servicio de Índice (IS) [4].

El FS recibe y rutea las consultas de usuario en el CS y el IS. El CS mantiene respuestas precomputadas a las consultas de usuario más frecuentes. La idea detrás de esto es utilizar menos recursos computacionales (cpu, disco y transferencias de red). Finalmente, el IS usa un índice invertido para responder consultas cuando éstas no están en el CS.

El flujo del procesamiento de consultas es como sigue: cuando una consulta llega al FS, se pregunta al CS para chequear si la respuesta está en el CS o no. Si la respuesta está en el CS es un *hit cache*, sino un *miss cache*. En caso de *hit*, la respuesta a la consulta es enviada al FS y posteriormente al usuario (el procesamiento finaliza). En caso de *miss*, la consulta es enviada al IS para obtener el conjunto de documentos relevantes a la consulta (resultados *top-k*). Esta respuesta es enviada al FS, y este servicio envía la respuesta final al usuario y paralelamente al CS para que éste la almacene.

El CS usa *Consistent Hashing* [5,6] para localizar los ítemes (consultas) en un conjunto de *N* nodos, los cuales mantienen las respuestas a las consultas más frecuentes en memoria (para hacer el proceso más rápido). Con Consistent Hashing, una consulta *q* es asignada a un y sólo un nodo en el servicio. Para efectos de alto desempeño, no es posible enviar la consulta a más de un nodo. Este método es determinístico y produce mínimo trastorno del servicio en caso de fallas en los nodos.

Las consultas de usuario siguen distribución Zipf [1], lo cual significa que un conjunto muy reducido de consultas son muy frecuentemente usadas por los usuarios. Como cada consulta es asignada a un solo nodo vía Consistent Hashing, una consulta muy frecuente puede sobrecargar el nodo seleccionado para atenderla.

La explicación dada anteriormente es la principal razón de porqué es importante diseñar un mecanismo que detecte las consultas en ráfaga. Para prevenir que una consulta en ráfaga sobrecargue un nodo debido a su alta frecuencia, esta consulta debe ser distribuida en múltiples nodo (en vez de uno). Con esta acción, se alivia la carga del nodo ante la aparición de consultas en ráfaga.

En este trabajo se usan redes neuronales para predecir la frecuencia de estas consultas. La idea es que si la frecuencia estimada está sobre un umbral previamente establecido, entonces es considerada ráfaga y es distribuida en múltiples nodos.

3 Redes Neuronales

Una red neuronal artificial es un modelo de caja negra inspirado en la estructura y funcionamiento del cerebro para la resolución de tareas en las que el ser humano se desempeña bien, como reconocimiento de formas. Sus unidades fundamentales son las neuronas, las cuales se encuentran interconectadas entre sí.

A diferencia de un enfoque algorítmico, una red neuronal es capaz de identificar los patrones presentes en el fenómeno estudiado. Básicamente, lo que hace la red neuronal es crear una función aproximada de la que en realidad explica el fenómeno. Entre los problemas usualmente tratados con ella se encuentran los de regresión, clasificación y reconocimiento de patrones.

Para verificar la existencia del aprendizaje usualmente se preparan dos conjuntos de datos: uno de aprendizaje, con el que se entrena a la red y otro de generalización o validación, con el que se verifica la validez del modelo. Ambos conjuntos deben ser independientes entre sí pero representan el mismo fenómeno estudiado.

Con el fin de procesar la información, las neuronas se organizan en diferentes grupos o capas, así, se habla de capa de entrada y de salida dependiendo de la función que cumplan las neuronas respecto a las variables del sistema representado. En la Figura 1 se puede apreciar un esquema de perceptrón multicapas.

Para la validación del modelo neuronal de caja gris propuesto se realiza el cálculo de índices de calidad, tales como el IA, RMS y RSD [7]. Las ecuaciones de los índices de calidad son las siguientes:

$$IA = 1 - \frac{\sum_{i=1}^{n}(o_i - p_i)^2}{\sum_{i=1}^{n}\left(|o_i'| + |p_i'|\right)^2} \quad RMS = \sqrt{\frac{\sum_{i=1}^{n}(o_i - p_i)^2}{\sum_{i=1}^{n} o_i^2}} \quad RSD = \sqrt{\frac{\sum_{i=1}^{n}(o_i - p_i)^2}{N}}$$

Donde o_i y p_i son los valores observados y predichos respectivamente en el tiempo i, y N es el número total de datos. Luego, $p_i'=p_i-o_m$ y $o_i'=o_i-o_m$, donde o_m es el valor medio de las observaciones.

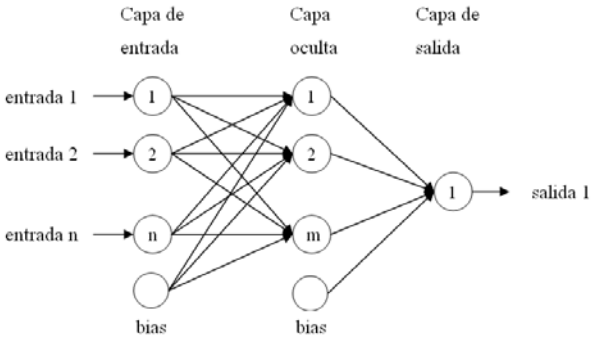

Figura 1: Perceptrón multicapas, con una capa oculta.

Las redes neuronales artificiales son capaces de modelar sistemas dinámicos mediante la inclusión de recurrencia en su arquitectura. Para este trabajo se utilizaron los modelos NAR, AR, NARMA y ARMA.

El modelo NAR (Non-linear Auto-Regressive) corresponde a la versión no lineal del modelo AR. Los parámetros para ambos modelos constan de autoregresores (AR) y retardos para los mismos.

El supuesto establecido sobre el ruido es que afecta al proceso de salida, de forma que en el tiempo k son influenciadas tanto la salida actual como las n_y salidas pasadas [8]. El efecto del ruido sobre el sistema queda representado por la siguiente ecuación:

$$y(k)=\psi[y(k-1),\ldots,y(k-n_y)]+w(k)$$

Donde y(k) corresponde a la salida medida, w(k) a un ruido aditivo y n_y a las mediciones pasadas. La representación gráfica del sistema corresponde a la de la Figura 2.

El modelo NARMA (Non-linear Auto-Regressive Moving Average) corresponde a la versión no lineal del modelo ARMA. Como parámetro adicional a los autoregresores (AR), considera una media móvil relativa al ruido (MA).

El supuesto establecido sobre el ruido es que además de afectar al proceso de salida, como en el modelo NAR y AR, también lo hace sobre la salida en si misma [8]. La influencia del ruido en el sistema queda representada por la siguiente ecuación:

$$y(k)=\psi[y(k-1),\ldots,y(k-n_y),w(k-1),\ldots,w(k-n_w)]+w(k)$$

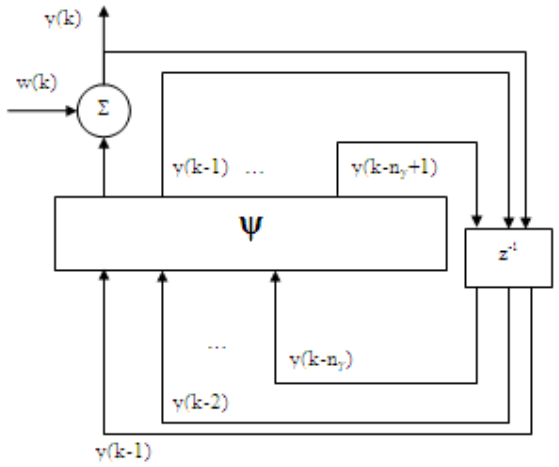

Figura 2: Estructura modelo NAR.

Donde adicionalmente a las variables y(k), w(k), y n_y, mencionadas para el modelo NAR, n_w corresponde al retardo en el error cometido. Esta situación se representa en la Figura 3.

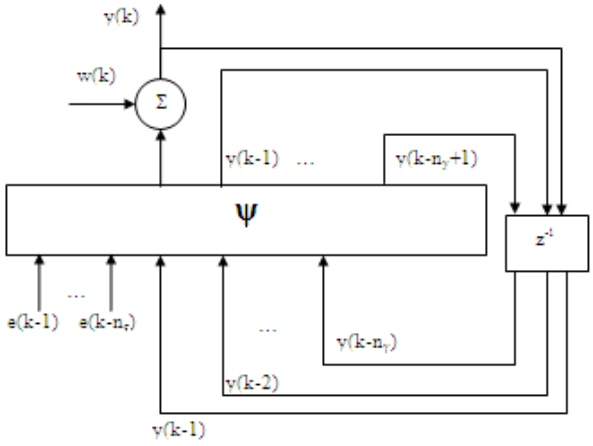

Figura 3: Estructura modelo NARMA.

Es importante chequear la capacidad de generalización de la red neuronal artificial, lo que también se conoce como validación del modelo neuronal. Hay dos formas básicas de validar un modelo dinámico, que es eminentemente predictivo, en predicción “un paso adelante” u OSA (one step ahead) y en predicción “múltiples pasos adelante” o MPO (model predictive output).

En la estimación OSA ingresan las variables de estado en un tiempo determinado y el modelo determina el valor del vector de estado en el instante de tiempo inmediatamente siguiente, no existiendo retroalimentación en el modelo, por lo que se estiman los valores sólo un paso adelante. En la estimación MPO, en un comienzo ingresan los valores iniciales de las variables de estado, luego el valor estimado por el modelo es retroalimentado a través de éste, utilizándose como la siguiente entrada. Así sucesivamente se utilizan todas las entradas estimadas para simular el proceso completo múltiples pasos adelante. Los esquemas de funcionamiento OSA y MPO pueden verse en la Figura 4 y la Figura 5, respectivamente.

4 Experimentación

El primer paso es la recolección de datos. Para esta labor, se consideran las consultas de usuarios realizadas en un período de 1 año a un buscador comercial. Además, sólo se estudian las consultas más frecuentes. La selección de las consultas más frecuentes es debida al comportamiento zipfiano de las consultas de usuario y al estudio de las consultas en ráfaga. Finalmente, cada consulta frecuente es representada como una serie de tiempo, en la cual cada punto es la frecuencia de la consulta en un minuto. Ésta será la única entrada a estudiar de las consultas.

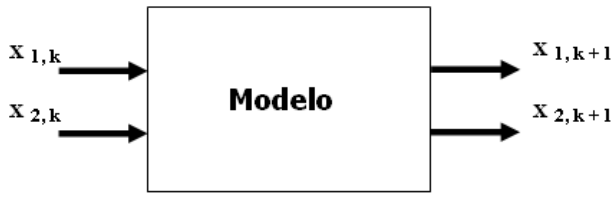

Figura 4: Esquemas de funcionamiento OSA.

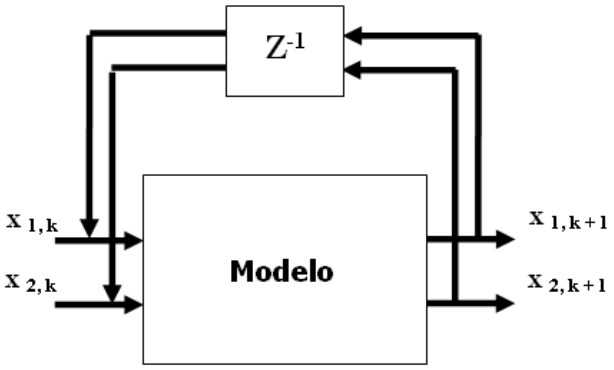

Figura 5: Esquemas de funcionamiento MPO.

De los 12 meses estudiados, se puede obtener la consulta más frecuente mes a mes. Para nuestro caso particular, la consulta más frecuente es 'facebook'. Esta consulta nos servirá para establecer cotas inferiores y superiores en términos de frecuencia. La idea es la siguiente: sólo seleccionar como primer filtro aquellas consultas que superen, en términos de frecuencia, a 'facebook' en un intervalo de tiempo definido. Como fue mencionado anteriormente, el intervalo de tiempo definido es un minuto.

Como conjunto resultante del procedimiento anterior, sólo quedarán consultas que son muy frecuentes (alguna vez superaron a la consulta histórica más frecuente). Cada una de estas consultas es representada como una serie de tiempo. Finalmente, se decide dejar dentro del grupo de consultas en ráfaga sólo las consultas que presentan cero en su serie de tiempo.

En conclusión, en el conjunto de entrenamiento para la red neuronal, sólo se encuentran consultas que en algún instante de tiempo son más frecuentes que 'facebook', y que además tuvieron frecuencia cero en algún intervalo de tiempo, conformando dichas acciones el criterio empleado para la identificación de ráfagas.

Según nuestra experiencia, el procedimiento anterior produce buenas consultas candidatas que pueden ser consideras como ráfagas. En la Figura 6 se observa un ejemplo de la consulta en ráfaga 'seal team six' obtenida con el procedimiento descrito. En la Tabla 1 se observa el conjunto final de consultas a considerar, y además la separación que se realizó en conjunto de entrenamiento (C_E) y validación (C_V).

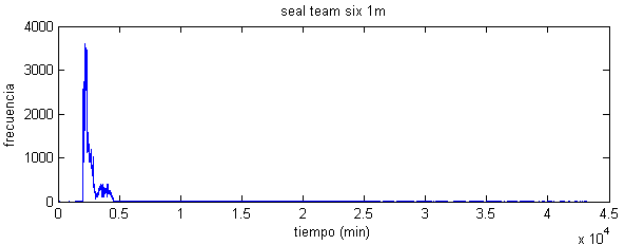

Figura 6: Frecuencia consulta en ráfaga 'seal team six' para un mes completo de ejecución. El intervalo de medición es a nivel de minutos.

Tabla 1: Consultas seleccionadas para el estudio.

Consulta	Conjunto
'sheen dumped'	C_V
'bin laden wives'	C_V
'seal team six'	C_E
'casey anthony trial'	C_V
'selena break up'	C_V
'amanda knox appeal'	C_E
'steve jobs'	C_E
'monica lewinsky'	C_E
'leah remini fired'	C_E
'sam wopat dies'	C_V

Con respecto a la red neuronal, se utilizó el método de Coeficientes de Lipschitz para determinar el orden del sistema o el número de mediciones pasadas para el modelo. Para este propósito fue utilizado el toolbox NNSYSID [9]. Los resultados obtenidos indican que el sistema es de orden 6, requiriéndose por tanto igual número de mediciones pasadas como entradas al modelo.

Para la construcción del modelo neuronal se considera el Perceptrón Multicapa con 14 neuronas en la capa oculta. Las métricas a evaluar son:

- Índice de adecuación (*Index of Agreement*, IA). Un valor aceptable es IA > 0,9.
- Raíz del error cuadrático medio (*Residual Mean Square*, RMS). Para esta métrica, un valor aceptable corresponde a RMS < 0,1.
- Desviación estándar residual (*Residual Standard Deviation*, RSD). Un valor RSD < 0,1 se considera aceptable.

En nuestra experimentación, se consideraron modelos lineales y no lineales, en los modos de simulación OSA (*One Step Ahead*) y MPO (*Model Predictive Output*). Sin embargo, los resultados con modelos no lineales (Tabla 2 y Tabla 3) no fueron auspiciosos, por lo que se optó por dejarlos fuera de este trabajo. Para los modelos lineales se utilizaron los modelos AR y ARMA implementados en el toolbox NNSYSID. Además, no se consideran entradas exógenas para establecer la serie de tiempo. Las neuronas de la capa oculta utilizan funciones de activación lineal, al igual que las neuronas de la capa de salida. Por último, el método de optimización utilizado es Levenberg-Marquardt.

Tabla 2: Simulación OSA para NAR para C_V.

Consulta	IA	RMS	RSD
'sheen dumped'	0.4312	0.8523	0.9676
'bin laden wives'	0.8154	0.6676	0.8094
'casey anthony trial'	0.9452	0.0714	0.0655
'selena break up'	0.5463	0.9227	1.1477
'sam wopat dies'	0.8967	0.3353	0.3591
Promedio	**0,7270**	**0,5699**	**0,6699**

Tabla 3: Simulación OSA para NARMA para C_V.

Consulta	IA	RMS	RSD
'sheen dumped'	0.2993	0.9690	2.8819
'bin laden wives'	0.4298	0.8953	3.6340
'casey anthony trial'	0.7828	0.1569	0.1464
'selena break up'	0.4017	0.9643	2.7649
'sam wopat dies'	0.5811	0.6224	1.0687
Promedio	**0,4989**	**0,7216**	**2,0992**

Dado que el fenómeno estudiado es el de las consultas más frecuentes no tiene sentido construir un modelo individual para cada consulta, por tanto, se desarrolla un estimador grupal en vez de uno individual. Para el entrenamiento se crea una única señal formada por la composición de las consultas seleccionadas para dicho fin. La composición de casos para el entrenamiento ha sido utilizada en una investigación de otra área [10]. Las ráfagas sólo se consideran desde el preámbulo de la subida hasta el máximo valor de la frecuencia ya que el resto de la señal no es de interés para una predicción.

En la Tabla 4 y Tabla 5 se observan los resultados de la simulación OSA para las series de tiempo lineales. Se observa un excelente valor promedio para IA, y valores promedios aceptables para RMS y RSD. Los mejores resultados para IA corresponden a la consulta 'selena break up', mientras que el mejor comportamiento RMS y RSD es para la consulta 'casey anthony trial'.

Tabla 4: Simulación OSA para AR para C_V.

Consulta	IA	RMS	RSD
'sheen dumped'	0.9947	0.0984	0.0904
'bin laden wives'	0.9947	0.1483	0.1061
'casey anthony trial'	0.9962	0.0191	0.0174
'selena break up'	0.9979	0.0773	0.0684
'sam wopat dies'	0.9935	0.0708	0.0657
Promedio	**0,9954**	**0,0828**	**0,0696**

Tabla 5: Simulación OSA para ARMA para C_V.

Consulta	IA	RMS	RSD
'sheen dumped'	0.9947	0.0990	0.0909
'bin laden wives'	0.9946	0.1492	0.1067
'casey anthony trial'	0.9962	0.0192	0.0175
'selena break up'	0.9978	0.0775	0.0685
'sam wopat dies'	0.9935	0.0707	0.0656
Promedio	**0,9954**	**0,0831**	**0,0698**

A continuación, en las Figuras 7 y 8 se muestran los resultados para las simulaciones AR y ARMA de la consulta 'selena break up'. Visualmente no se aprecian diferencias.

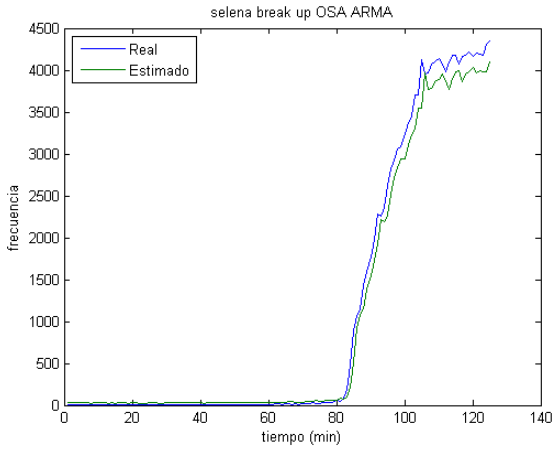

Figura 7: Predicción OSA ARMA para consulta '*selena break up*'.

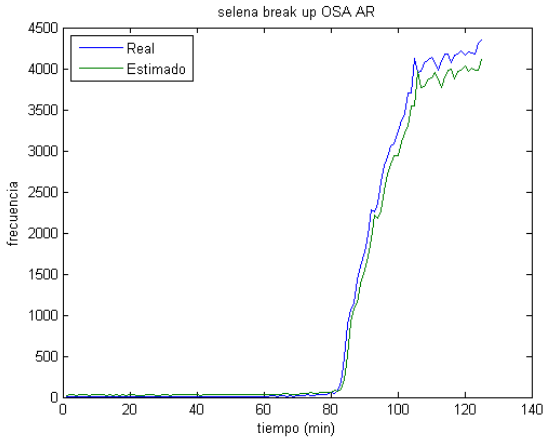

Figura 8: Predicción OSA AR para consulta '*selena break up*'.

Respecto a la comparación entre modelos lineales, ambos obtienen resultados prácticamente idénticos. Sin embargo, la elección en este caso es el modelo AR ya que ARMA es más complicado y no obtiene una mejora notoria.

El paso siguiente consistió en verificar la validez de las predicciones del modelo en simulación MPO en un horizonte de 10 pasos adelante, situación que equivale a tomar las 6 últimas mediciones para estimar las 4 siguientes. De acuerdo a los resultados presentados en la Tabla 6 y Tabla 7, se observó un rápido deterioro de las métricas.

Tabla 6: Simulación MPO 10 para AR para C_V.

Consulta	IA	RMS	RSD
'sheen dumped'	0.2219	0.7895	0.4387
'bin laden wives'	0.0905	0.5997	0.4208
'casey anthony trial'	0.7451	0.0180	0.0156
'selena break up'	0.0428	0.8756	0.5444
'sam wopat dies'	0.0189	1.2705	0.9230
Promedio	**0,2547**	**0,7651**	**0,4755**

Tabla 7: Simulación MPO 10 para ARMA para C_V.

Consulta	IA	RMS	RSD
'sheen dumped'	0.1146	0.8139	0.5282
'bin laden wives'	0.0375	0.6263	0.4759
'casey anthony trial'	0.7408	0.0185	0.0160
'selena break up'	0.0297	0.8683	0.5893
'sam wopat dies'	0.0103	1.2561	0.9509
Promedio	**0,2186**	**0,8223**	**0,5224**

5 Conclusiones

En el presente trabajo se ha estudiado la creación de un predictor de frecuencia para las consultas más usuales recibidas por un motor de búsqueda. La idea de diseñar un predictor está relacionada con la detección anticipada de consultas en ráfagas, ya que considerando la frecuencia estimada se define qué consulta ha aumentado su frecuencia en forma abrupta y, por ende, es considerada ráfaga. Este fue un intento inicial y exploratorio para determinar la factibilidad de detectar consultas en ráfagas utilizando un predictor de frecuencias. Como una primera aproximación al problema se utiliza una serie de tiempo, de forma que con las 6 últimas mediciones se realiza una buena predicción de la frecuencia siguiente.

El toolbox NNSYSID utilizado para el entrenamiento y prueba del modelo, ha facilitado el análisis de linealidad al permitir usar neuronas lineales en la capa oculta del perceptrón multicapa. Además, ha ayudado en la representación gráfica de la información al momento del entrenamiento para conocer y analizar los resultados de este proceso.

Las aproximaciones utilizando los modelos AR y ARMA han entregado buenos resultados en un esquema de simulación OSA, lo cual se comprueba de manera gráfica, así como también en los buenos resultados obtenidos en los índices de calidad IA, RMS y RSD.

Los resultados en simulaciones MPO no han sido los esperados, como trabajo futuro, se espera conseguir la implementación del predictor de búsqueda, para lo cual se piensa fortalecer el modelo con variables que den indicios sobre el aumento de frecuencia. Una interrogante que sigue abierta es cómo diferenciar las consultas en ráfaga del resto de las consultas más frecuentes.

Referencias

[1] R. Baeza-Yates, B. Ribeiro-Neto. “Modern Information Retrieval: The Concepts and Technology behind Search”, (2011).

[2] H. Yan, S. Ding, T. Suel. “Compressing term positions in web indexes”, in Proceedings of the *32nd International*

ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2009), pp. 147-154, (2009).

[3] T. H. Haveliwala. "Efficient Encodings for Document Ranking Vectors", in Proceedings of the *International Conference on Internet Computing*, pp. 3-9, (2003).

[4] C. Gómez-Pantoja, D. Rexachs, M. Marin, E. Luque. "A Fault-Tolerant Cache Service for Web Search Engines: RADIC evaluation", In Proceedings *18th International European Conference on Parallel and Distributed Computing* (Euro-Par 2012), pp. 298–310, (2012).

[5] D. Karger, E. Lehman, T. Leighton, R. Panigrahy, M. Levine, D. Lewin. "Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web", in Proceedings of the *29th Annual ACM Symposium on Theory of Computing*, pp. 654-663, (1997).

[6] D. Karger, A. Sherman, A. Berkheimer, B. Bogstad, R. Dhanidina, K. Iwamoto, B. Kim, L. Matkins, Y. Yerushalmi. "Web Caching with Consistent Hashing", *Computer Networks*, **31 (11)**, pp. 1203-1213, (1999).

[7] F. Cruz, G. Acuña. "Indirect Training with Error Backpropagation in Gray-Box Neural Model: Application to a Chemical Process", In Proceedings *XXIX International Conference of SCCC*, pp. 265-269, (2010).

[8] G. Dreyfus. "Neural Networks: Methodology and Applications", *Springer*, Germany, (2005).

[9] M. Nørgaard, O. Ravn, N. K. Poulsen. "NNSYSID-Toolbox for system identification with neural networks", *Mathematical and Computer Modelling of Dynamical Systems*, **8 (1)**, pp. 1-20, (2002).

[10] C. Sanhueza. "Evaluación de las redes neuronales recurrentes y máquinas de vectores soporte para la estimación no invasiva de la presión intracraneal", *Master Thesis*, Universidad de Santiago de Chile, (2011).