Mejorando el Agrupamiento Solapado de Recursos Web para un Dominio Específico

María R. Romagnano and Martín G. Marchetta

***Abstract*—El agrupamiento de recursos web es una tarea significativa y difícil. En trabajos anteriores se sugirieron distintas formas de unificar fuentes de información web, tratando de solucionar inconvenientes como heterogeneidad de contenido, falta de estructura, disponibilidad, distribución, cantidad y calidad. No obstante, existen dominios más complejos que otros, donde encontrar la solución es un problema aún más engorroso, debido a la cantidad y variedad de información que se maneja y a que el conjunto de fuentes que se consultan es considerable. Un ejemplo concreto es el dominio del turismo. Si bien existen prototipos y aplicaciones comerciales que asisten al turista web, actualmente denominados sistemas recomendadores de paquetes turísticos, muchas fuentes de información se duplican y dispersan en la red. El desafío de una aplicación que brinde información concisa y unificada va más allá de ubicar dónde está esa información, procurando además saber de qué forma llegar a ella y cómo agruparla. En este trabajo se presenta una metodología para recuperar recursos web, de la cual forma parte un mecanismo para agrupar esos recursos, permitiendo el solapamiento entre grupos. Por consiguiente, con esta contribución se logra mejorar la precisión en las respuestas para un dominio establecido a priori.**

***Index Terms*— Clustering methods, Information analysis, Information retrieval, Web Search**

I. Introducción

La enorme y rápida acumulación de datos, el advenimiento de las tecnologías y el uso masivo de Internet han permitido obtener información de una gran cantidad de fuentes disponibles en la Web. La WWW se ha convertido en una de las mayores fuentes de información sobre prácticamente todas las áreas de interés, lo cual ha determinado el crecimiento exponencial del uso de la Web como repositorio de información. El apogeo y la moda de usar Internet, la expansión de este tipo de fuente de información ofrecen una prometedora oportunidad de búsqueda y extracción de información. Así por ejemplo leer el diario, conocer el clima, programar un viaje, o simplemente hablar con otra persona, lo hacemos de manera más rápida, eficiente y en tiempo real.

En la sociedad de la información se destina un importante número de recursos para recuperar, procesar y recopilar enormes volúmenes de información y así poder obtener de ella el conocimiento que se necesita. Pero, ¿cómo podemos adquirir conocimiento de tanta información existente en la web? Si bien el ser humano está dotado de inteligencia es evidente que no puede procesar, manualmente o con la ayuda de simples herramientas estadísticas, millones de datos almacenados en grandes bases de datos. Además, resultaría altamente relativo, costoso y lento. La posibilidad de consultar información proveniente de distintas fuentes de información web requiere de estándares y herramientas que faciliten esta práctica. Actualmente, en la mayoría de los casos, el acceso a la misma se hace a través de buscadores. Si bien los motores de búsqueda actuales suelen ser muy eficaces, los resultados que proporcionan en la mayoría de los casos no resultan plenamente satisfactorios para los usuarios. La cantidad de información que se recibe es tan vasta que resulta difícil, casi imposible, de asimilar.

El almacenamiento y la posterior recuperación de la información, desde que se usaban las tablas de piedra hasta el uso actual de los medios digitales, representan uno de los problemas a los que la humanidad se ha tenido que enfrentar desde la invención de la escritura. Con la aparición de las nuevas tecnologías de información y comunicación (NTICs) este problema se ha resuelto parcialmente. Es más fácil producir datos que guardarlos, administrarlos y recuperarlos. Se cree que en el año 2020 la cantidad de datos mundial llegará a 35ZB [1].

Si bien se cuenta con herramientas de búsqueda, la Web crece a una velocidad mucho más rápida que la de cualquier tecnología actual para indexar páginas web [2]. Para tener una idea, por ejemplo, se puede mencionar que en el año 1998 Google indexó 26.000.000 de páginas y recibió 3.600.000 de consultas y en el año 2014 indexó 30.000.000.000.000 de páginas y recibió 2.095.100.000.000 de consultas [3], [4].

Por lo tanto, la Web puede considerarse como un repositorio digital, ya que permite contar con un "abanico de información" fácil y rápidamente. Sin embargo, esta accesibilidad se convertirse en un problema cuando se presentan miles de respuestas a una consulta y ninguna de ellas es satisfactoria o por el contrario no se encuentra respuesta.

El desafío de una aplicación que brinde información concisa y unificada va más allá de ubicar dónde está esa información, procurando además saber de qué forma llegar a ella y cómo agruparla. En trabajos anteriores se han sugerido distintas formas de agrupar fuentes de información, tratando de

María R. Romagnano es docente/investigador del Instituto de Informática y del Dpto. de Informática, FCEFN, UNSJ, San Juan, Argentina.
E-mail: maritaroma@iinfo.unsj.edu.ar.
Martín G. Marchetta es docente/investigador del Centro Universitario, FI, UNCu, Mendoza, Argentina.
E-mail: mmarchetta@fing.uncu.edu.ar.

solucionar inconvenientes como heterogeneidad de contenido, falta de estructura, disponibilidad, distribución, cantidad y calidad. No obstante, existen dominios más complejos que otros, donde encontrar la solución es un problema aún más engorroso, debido a la cantidad y variedad de información que se maneja y a que el conjunto de fuentes que se consultan es considerable. Un ejemplo concreto es el dominio del turismo. Si bien existen varios prototipos y aplicaciones comerciales que asisten al turista web en su búsqueda, estos siguen manteniendo el inconveniente de contar con un número limitado de información. Así por ejemplo, generalmente, una vez que estos sistemas han vendido toda la plaza de la que disponen de un cierto hotel, el turista se queda sin visualizar ese hotel y en realidad si él busca por su cuenta en la web o en otra aplicación turística probablemente encuentre una habitación disponible de dicho hotel.

Se pone de manifiesto la necesidad contar con un mecanismo que permita la clasificación y el agrupamiento solapado de recursos web, para un dominio específico y de acuerdo a un criterio preestablecido. Este trabajo propone un método que ofrece agrupar recursos web, permitiendo el solapamiento entre grupos. Adicionalmente, se presenta una metodología para recuperar información web, de la cual forma parte dicho método. Por consiguiente, con esta contribución se logra mejorar la precisión en las respuestas para un dominio establecido a priori.

El resto del trabajo se estructura de la siguiente forma: la sección 2 presenta trabajos relacionados, la sección 3 presenta la propuesta describiendo en detalle el método de agrupamiento enmarcado dentro de la metodología para recuperar recursos en la web, en la sección 4 se presentan la experimentación y los resultados obtenidos, en la sección 5 se ponen en discusión los resultados alcanzados. Finalmente, la sección 6 presenta conclusiones.

II. Trabajos Relacionados

Para llevar a cabo el posterior análisis de la información recuperada, los recursos web pueden agruparse de acuerdo a un determinado criterio. Este agrupamiento puede realizarse a través del aprendizaje automático utilizando técnicas de aprendizaje supervisado, como la clasificación, o técnicas de aprendizaje no supervisado, como el clustering [5]. El agrupamiento es útil en muchas técnicas exploratorias de análisis de patrones, minería de datos, toma de decisiones, etc.; es decir en el aprendizaje automático. Sin embargo, en muchos de estos problemas existe poca información previamente y a la hora de tomar las decisiones se deben hacer la menor cantidad posible de suposiciones.

La diferencia entre clasificación y clustering puede parecer insignificante al principio, debido a que en ambos casos tenemos una partición de un conjunto de recursos en grupos. Sin embargo, ambos problemas son fundamentalmente diferentes. En la clasificación el objetivo es reproducir una distinción categórica que un supervisor humano impone a los datos. En el clustering no se tiene tal maestro que establece las categorías [6]. Según Qi la clasificación es mucho más que una simple asignación de etiquetas, de categorías y una organización de información. La clasificación es tradicionalmente vista como un problema de aprendizaje supervisado en el cual un conjunto de datos etiquetados son usados para entrenar a un clasificador, el cual luego será usado para clasificar o etiquetar futuros ejemplos [7].

El clustering es el proceso de agrupamiento de datos en clases o clusters. Todos los objetos de un mismo cluster tienen alta similitud en comparación con objetos de otros clusters. El clustering obtiene grupos o clases desde un aprendizaje por observación y, alternativamente, puede usarse como un paso de pre-procesamiento de otros algoritmos, tales como selección de un subconjunto de atributos, caracterización, clasificación, etc. El clustering presenta el inconveniente de que hay diferentes formas de definir los grupos y la calidad de tales agrupamientos dependen del conjunto de datos de entrada y de la función objetivo. Por tal motivo existe una gran cantidad de algoritmos de clustering [5]. Por su parte, Manning, Raghavan y Schütze, plantean que el aporte fundamental para un algoritmo de agrupamiento es la medida de la distancia. Las diferentes medidas de distancia dan lugar a diferentes agrupamientos. Por lo tanto, la medida de distancia es un importante medio que puede influenciar en el resultado de la agrupación [8].

En [9] y en [10] se propone clasificar páginas web a través de algoritmos genéticos, usando las URLs de las páginas, n-gramas, teniendo en cuenta el contenido y la semántica de las páginas. En [11], [12], [13], [14], [15], [16] y en [17] se realiza clustering de páginas web considerando palabras claves, contenido semántico, ontologías y etiquetas como herramientas externas y mapas auto organizados, K-Means y C-Means como métodos estándares para realizar clustering hard y fuzzy respectivamente. En [18] y [19] además de permitir el solapamiento entre clusters proponen la idea de agrupar los resultados de búsqueda web ya generados por motores de búsqueda convencionales. En [20] se agrupan páginas web independizándose de los clásicos algoritmos de clustering, usando el diccionario de Internet para contextualizar las palabras claves. Por su parte, en [21] se propone usar una técnica estadística para clusterizar páginas web y en [22] además de una ontología y HowNet usa el modelo SVD para clusterizar. En [23] se usa lógica difusa para evaluar sitios web y en [24] se propone una representación basada en lógica borrosa para clusterizar páginas web. En [25] se plantea un método que recupera y agrupa fuentes de información web de acuerdo a los servicios que ofrecen, permitiendo al usuario obtener respuestas precisas, reduciendo el tiempo y la complejidad en la búsqueda. En [26] se propone un método el cuál mediante un agente de filtrado localiza fuentes de información en la web, las agrupa y luego brinda al usuario información precisa, acorde a sus necesidades. Además, para determinar la relevancia de una página han usado lógica difusa. Para agrupar las páginas similares proponen un algoritmo basado en los métodos K-means y Fuzzy C-means. En [27] se esboza una metodología para asistir al usuario web en su búsqueda de información en un

dominio de aplicación determinado.

III. PROPUESTA

A. *Introducción*

Habitualmente, para lograr recuperar e integrar recursos desde diferentes sitios web, se requiere de aplicaciones especializadas, complejas y con dificultades en tiempo de desarrollo y permanente mantenimiento. Lo ideal sería poder recuperar estos recursos como si se estuviese trabajando en bases de datos, con la misma facilidad y transparencia de contenidos. Cuando se realiza una búsqueda, más allá de establecer palabras claves, usar comillas u operadores booleanos, existe la posibilidad de encontrarse con recursos que nada tienen que ver con la búsqueda deseada (Fig. 1). Por otra parte, existen recursos web que se repiten más de una vez, y aún más significativo, recursos que contienen información pertinente a la búsqueda realizada y los mismos se encuentran en las últimas páginas del buscador. Un ejemplo de este caso puede observarse en Fig. 2, donde el documento del Parque Nacional San Guillermo se encuentra en la página Nº 15 y puede ser de interés para un turista debido a su abundante contenido en información turística del lugar. Obviamente, este recurso raramente será visitado por un interesado, seguramente abandona la búsqueda antes de llegar a él.

Si bien en este trabajo se plantea la metodología para recuperar recursos web, principalmente se aborda el método de agrupamiento y su respectivo algoritmo.

Fig. 1. Resultado incorrecto de una búsqueda, en una posición considerable de ranking.

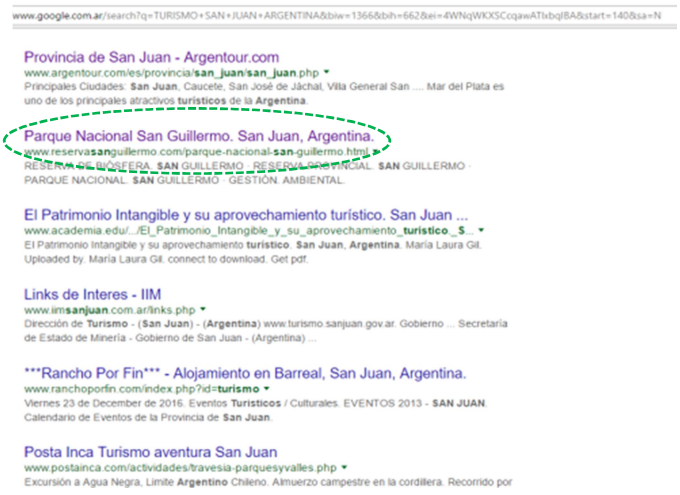

Fig. 2. Resultado correcto de una búsqueda, en una posición no considerable de ranking.

B. *Etapas de la Metodología*

La metodología sugiere seis etapas, abarcando desde el relevamiento, procesamiento e indexado de los recursos web hasta resolver la consulta del usuario.

1) *Buscar*

Cada cierto período de tiempo (cuya frecuencia variará dependiendo del dinamismo con el cual cambie el dominio en cuestión), automáticamente y a través de las APIs provistas por los buscadores generales e índices temáticos, se realiza la búsqueda de la información con las palabras claves DOMINIO, PROVINCIA y PAÍS. Podemos mencionar a Google, Yahoo o Bing como ejemplos de buscadores e índice más populares que proveen un conjunto de funciones y procedimientos, que ofrecen una biblioteca determinada para ser usados por otra aplicación como una capa de abstracción en sus búsquedas web.

2) *Pre-procesamiento de recursos relevantes*

En este hito de la metodología se realiza un análisis preliminar de los resultados obtenidos en la etapa anterior, seleccionando aquellos recursos que sean relevantes al dominio en cuestión. Se establece que se tendrán dos clases bien definidas: relevantes y no relevantes. Se sugieren dos técnicas para determinar la relevancia de un recurso web:

a) Aprendizaje supervisado. Un experto provee ejemplos de recursos relevantes y no relevantes del dominio. De esta forma se aprende qué términos clasifican los recursos en relevantes y no relevantes. Además, se usa una ontología preexistente del dominio de aplicación para determinar sinónimos y/o relaciones entre términos y así poder contemplar la semántica. Esta tarea se puede realizar haciendo uso de una herramienta que soporte el aprendizaje automático. Así por ejemplo se puede mencionar a Orange, Weka, RapidMiner, entre otras.

b) Análisis discriminante. A través de un experto en el dominio se establece cuáles serán las variables de entrada. Se analiza el código fuente de cada uno de los recursos para establecer la frecuencia de dichas variables en cada recurso. Con esta información se arma una base de recursos candidatos para determinar en qué clase se ubicará cada uno. Se obtiene una función discriminante que ayuda a determinar o discriminar en qué clase se ubicará cada recurso. Esta tarea se puede hacer con ayuda de un software estadístico tal como SPSS o R, por ejemplo.

Seguidamente, todos aquellos recursos que hayan quedado en la clase de relevantes se seleccionan para continuar con las siguientes etapas de la metodología.

3) Seleccionar términos relevantes

En esta etapa se determinan cuáles serán los términos relevantes, los que posteriormente serán nombres de los agrupamientos. Para seleccionar dichos términos se emplean herramientas externas tales como bolsa de palabras (producto de la consulta a un experto del dominio), DBpedia, WorNet y la ontología del dominio.

Manualmente, el experto en el dominio es quién determina qué términos serán los candidatos a ser seleccionados y posteriormente en forma automática se realiza una comparación de estos con los términos de las restantes herramientas. Es decir, se realiza un análisis semántico, estableciendo relaciones y/o diferencias semánticas (sinonimia o antonimia, polisemia u homonimia, hiperonimia o hiponimia, holonimia o meronimia) entre cada subdominio de los propuestos por el experto y los términos de cada una de las restantes herramientas externas. Aquellos términos dados por el experto que sean similares o que coincidan con los términos de al menos dos de las tres herramientas restantes serán considerados como relevantes y por consiguiente serán establecidos como nombres de los futuros grupos (Fig. 3). Este requerimiento asegura una mejor cobertura de la semántica.

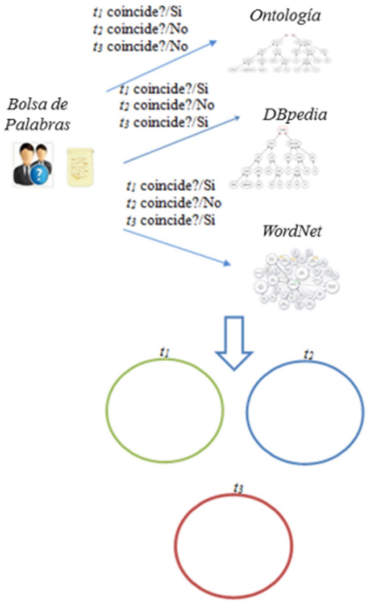

Fig. 3. Elección de términos relevantes.

4) Determinar el valor de cada término en cada recurso

La tarea de analizar y obtener información que se halla en los recursos web, a veces, resulta ardua. Así por ejemplo si ese recurso es un documento web dicha tarea no es similar a la de un documento de texto. En la web se encuentra diversidad de lenguajes, gran cantidad de datos heterogéneos, distribuidos, ubicuos, redundantes, y/o no estructurados.

El objetivo de esta etapa es obtener la cantidad de información que ofrece un recurso sobre los términos discriminantes, es decir los términos relevantes al dominio comprometido.

Los recursos seleccionados como relevantes fueron establecidos y almacenados en una base en la etapa anterior. Esta base se representa en la Tabla 1, donde R_j representa la j-ésimo recurso, t_i representa el i-ésimo término relevante y w_{ij} representa la normalización del número de veces que el término ti aparece en el recurso R_j.

El procedimiento consiste en analizar el contenido de cada recurso almacenado en esta base y a través de minería web remover las stopwords y realizar stemming para determinar la frecuencia de aparición de cada término relevante y sus variantes. Nuevamente, se usan las herramientas externas propuestas para establecer la correlación semántica entre los términos de las diversas fuentes de información web y los términos definidos como relevantes en la etapa anterior. Luego, se usa el esquema TF [28] para calcular el peso w_{ij} de los términos relevantes en cada recurso según la ecuación:

$$w_{ij} = \frac{f_{ij}}{\max\left\{f_{1j}, f_{2j}, \ldots, f_{nj}\right\}} \qquad (1)$$

f_{ij}: frecuencia del término relevante t_i en R_j.
n: cantidad de términos.

TABLA I. TÉRMINOS CON SUS PESOS PARA CADA RECURSO RELEVANTE, EN UN DETERMINADO DOMINIO

Rj	***URL***	t_1	t_2	t_3	…	t_n
R_1	URL_1	w_{11}	w_{21}	w_{31}	…	w_{n1}
….						
R_m	URL_m	w_{1m}	w_{2m}	w_{3m}	…	w_{nm}

5) Agrupar

En esta etapa la metodología propone que aquellos recursos que presenten información de un mismo término se agrupen, con el objetivo de concentrar en cada grupo aquellos recursos que cuenten con un término (o sus sinónimos) establecido como relevante. La intensión de agrupar consiste en disminuir los tiempos y aumentar la precisión en las respuestas.

Los recursos pueden presentar información de varios términos relevantes. Necesariamente la metodología debe permitir que un recurso pueda corresponder a uno o a más grupos con un cierto grado de pertenencia. La idea de esta etapa se muestra en la Fig. 4, donde G_j representa el grupo al cual pertenecen recursos similares, agrupados en función del término t_i que resulte como centro al aplicar el algoritmo y w_{ij} es el valor numérico que representa el peso del término t_i en el recurso R_j; calculado en la etapa anterior.

Algoritmo

Para desarrollar el algoritmo se tomaron como base métodos de clustering particionales y soft [29]. De los métodos particionales se adopta la simplicidad en cuanto a la idea de elegir un centro para cada grupo y agrupar los restantes

elementos en función de la distancia de ellos con el centro. Pero como dichos métodos no permiten que un elemento pueda pertenecer a más de un grupo, se adopta esta posibilidad de un método soft.

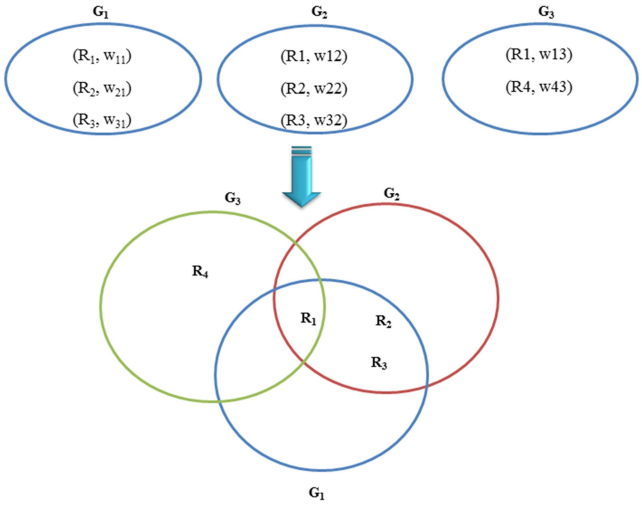

Fig. 4. Agrupamiento de recursos web.

Como puede observarse la propuesta toma la idea principal de estos dos métodos populares, pero además como se tienen definidos de antemano cuáles serán los grupos que se desean tener, los cuales coinciden con los términos preestablecidos como relevantes, también se adopta la idea principal de los métodos de clasificación.

Los grupos se eligen una única vez, con un criterio que se define luego, y se mantienen durante todo el proceso. La pertenencia de cada elemento a cada grupo se registra en una matriz y en una única iteración. La dimensión de cada elemento coincide con la cantidad de grupos, disminuyendo considerablemente la cantidad de operaciones a realizar por el algoritmo. Se asume que a cada grupo G_i se le asignará el nombre de cada columna t_i de la Tabla 1 y que el mismo tendrá un valor central al cual se lo llama centro.

A continuación se desglosa en partes el algoritmo, en función de cada una de las actividades principales que se llevan a cabo.

Algoritmo A. SeleccionCentros()

```
⋮
// m cantidad de recursos
// n cantidad de grupos
⋮
//Calcular el máximo de cada columna
Para i=1 hasta n con incremento +1
  max-w[i]=0; max-f[j]=0;
  Para j=1 hasta m con incremento +1
    Si  w[i,j]> max-w[i]
     entonces a max-w[i] asignar el contenido de w[i,j]
     sino Si w[i,j] = max-w[i]
            entonces
              // Calcular la máxima frecuencia del término ti si hay empate
              asignar a p el valor de i // p contiene la posición del máximo
             Si f[i,j] > max-f[p,j]
               entonces asignar a max-f[p,j]el valor de f[i,j]
               sino Si  f[i,j]=max-f[p,j]
                      entonces
                       // Calcular la máxima densidad del documento
                        Si den[i,j] > den[p,j]
                             entonces asignar a centro[i] el valor de w[i,j]
                             sino Si den[i,j] < den[p,j]
                                    entonces asignar a centro[i] el valor de  w[p,j]
                                    sino el sistema elige un recurso aleatoriamente.
                                FinSi
                        FinSi
                    FinSi
          FinSi
    Fin Si
  Fin Para
Fin Para
⋮
//Calculo de la frecuencia de cada término en cada documento y la máxima frecuencia de términos
f[i,j]=0; max_f[i,j]=0;
Para j=1 hasta m con incremento +1
 Para i=1 hasta n con incremento +1
  Mientras ∈ términos (o sus sinónimos) por analizar
   Si término ti= Cluster[i]
     entonces
       Incrementa f[j,i]
       Si f[j,i] > max_f[i,j]
         entonces asignar a max_f[i,j] el valor de f[j,i]
       FinSi
    FinSi
  FinMientras
 FinPara
FinPara
⋮
//Calcular el peso de cada término en cada documento
Para j=1 hasta m con incremento +1
 Para i=1 hasta n con incremento +1
   asignar a w[j,i] el valor de f[j,i]/max_f[j,i]
 FinPara
FinPara
⋮
//Calcular la densidad de cada documento
Para j=1 hasta m con incremento +1
 den[j,i]=0; den_total[j,i]=0;
 Para i=1 hasta n con incremento +1
   Si  f[j,i]> 0
    incremento +1 den[j,i];
   FinSi
 FinPara
```

```
  den_total[j,i]= den[j,i]/n;
FinPara
⋮
```

Es decir que para cada grupo G_i se calcula el centro c_i como el máximo valor de cada columna. Si dos o más recursos tuviesen un valor máximo se desempata con el recurso que tenga máxima frecuencia del término t_i. Si siguen siendo candidatos en una tercera instancia se calcula su densidad De_j como la cantidad de términos relevantes que son cubiertos por este recurso.

$$\max\left\{De_j\right\} \quad (2)$$

Por último, ante la igualdad entre recursos, el sistema elige aleatoriamente uno de los recursos en competencia.

Algoritmo B. Similitud()

```
⋮
//Calcular la similitud entre cada recurso y cada centro, para todos
los centros.
// m cantidad de recursos
//n cantidad de grupos
// wc[i] es un vector que almacena los pesos de los centros
Para i=1 hasta n con incremento +1
  similitud = 0;
  Para j=1 hasta m con incremento +1
        sim[i, j] = w[i, j] / wc[i] * 100;

        Similitud = similitud + sim[i,j];
  FinPara
FinPara
⋮
```

Es decir que para cada recurso por analizar se calcula la similitud de cada recurso con cada centro c_i. La similitud viene dada por la siguiente expresión:

$$S_{r_j c_i} = \frac{w_{ij}}{w_{ci}} * 100 \qquad (3)$$

Sr_jc_i: similitud entre el recurso r_j y el centro c_i
w_{ij}: peso del término t_i en el recurso r_j.
w_{ci}: peso del centro c_i.

Esto dará la idea de qué porcentaje del máximo (centro) está cubriendo el recurso analizado, para un término en particular.

Además se determina el umbral de similitud como se establece en la fórmula:

$$U_S = \frac{1}{n} \sum_{j=1}^{n} Sr_j c_i \qquad (4)$$

U_s: umbral de similitud
Sr_jc_i: similitud entre el recurso r_j y el centro c_i
n: cantidad de recursos

Algoritmo C. Agrupar()

```
⋮
//Analizar por grupo todos los recursos para determinar en qué
grupo se ubican.
// m cantidad de recursos
//n cantidad de grupos
Para i=1 hasta n con incremento +1
  Para j=1 hasta m con incremento +1
    Si sim[i,j] >= Usi
          entonces ubicar el recurso Rj con su respectivo grado
          de pertenencia (wij) en el grupo Gi
    FinSi
  FinPara
FinPara
⋮
```

Es decir, si la similitud del recurso R_j con el centro c_i es mayor o igual a U_S entonces dicho recurso se selecciona y ubica en el grupo G_i con su respectivo grado de pertenencia, es decir el peso w_{ij}. Caso contrario, se considera poco similar y se descarta como miembro ese grupo.

6) Responder a la consulta

En esta etapa la metodología plantea que ante la consulta de un usuario, en lenguaje natural, se deberán realizar las siguientes tareas:

1. Remover los sotpwords.
2. Realizar stemming.
3. Analizar el dominio. Como el sistema tiene preestablecido que dominios puede cubrir, se debe determinar cuál de ellos es el que se encuentra en cuestión. Para realizar esta tarea el sistema deberá comparar los términos de la consulta y la ontología de cada uno de los dominios cubiertos.
4. Instanciar la aplicación con el dominio comprometido.
5. Establecer, a través de herramientas externas (bolsa de términos, ontología, DBpedia, WordNet), relaciones semánticas entre los términos de la consulta y el nombre del o los grupos a los cuales se debe acceder para responder a la consulta.
6. Otorgar un listado de URLs.

Para establecer el orden (o ranking) en el cual se van a mostrar las direcciones web se propone la siguiente función de relevancia y cada una de sus partes integrantes, como se muestra en las fórmulas:

$$Rr_j = dr_j + A_j + dc_j \quad (5)$$

Rr_j: relevancia del recurso r_j.

$$dr_j = \sum_{i=1}^{n} \frac{w_{ij}}{wc_i} ; \; 0 < dr_j <= 1 \qquad (6)$$

dr_j: densidad relativa del recurso r_j.
w_{ij}: peso del término t_i en el recurso r_j.
wc_i: peso del centro c_i.
n: cantidad de términos relevantes del recurso r_j.

$$A_j = \begin{cases} 0 \text{ si el documento no aparece en el historial de consultas} \\ 1 \text{ si el documento aparece en el historial de consultas} \end{cases}$$

A_j: aparición del recurso r_j en el historial de consultas.

$$dc_j = \frac{ctcr_j}{ctc}\ ;\ 0 < dc_j <= 1 \qquad (7)$$

dc_j: densidad de cobertura del recurso r_j.
$ctcr_j$: cantidad de términos de la consulta que aparecen en el recurso r_j.
ctc: cantidad de términos de la consulta.

IV. Experimentación y Resultados

Para realizar la experimentación se eligieron sólo páginas web relacionadas con el dominio del turismo, por ser uno de los dominios con mayor cantidad de información a manejar y que se actualiza continuamente. En esta sección se expondrán resultados obtenidos de las experimentaciones realizadas en las etapas de clasificación (pre-procesamiento de recursos relevantes) y agrupamiento.

Para realizar la etapa de *pre-procesamiento de recursos relevantes* se consultó a expertos en el dominio los cuales otorgaron términos que hacen que una página sea relevante o no relevante. Además, del análisis empírico se pudo determinar que palabras como subsidio, emprendimiento, carreras, colegio, universidades y todos sus sinónimos hacían que una página no fuese relevante (Tabla 2).

Tabla II. Términos Relevantes y No Relevantes

Relevantes	No Relevantes
Alojamientos	Ministerio
Agencias	Municipio
Destinos	Gobierno
Turismo	Colegio
San Juan	Estudiar
Argentina	Subsidiar
	Emprendimiento
	Provincias
	Minería
	Carrera

Por otra parte, los expertos en el dominio proporcionaron una base de 157 páginas para poder realizar las pruebas experimentales (Fig. 5). Dichas pruebas fueron realizadas con Weka 3.8, bajo Windows 10 y procesador Core i7.

En Fig. 6 puede observarse que se trataba de un problema desbalanceado respecto de la clase Relevancia, debido a que se obtuvieron 114 relevantes y 43 no relevantes.

Luego se construyó el clasificador. Para analizar qué clasificador era el más conveniente se probaron varias técnicas de clasificación basadas en reglas, en árboles, en Bayes, etc. La Fig. 7 revela que la técnica Funciones/Logistic clasifica mejor.

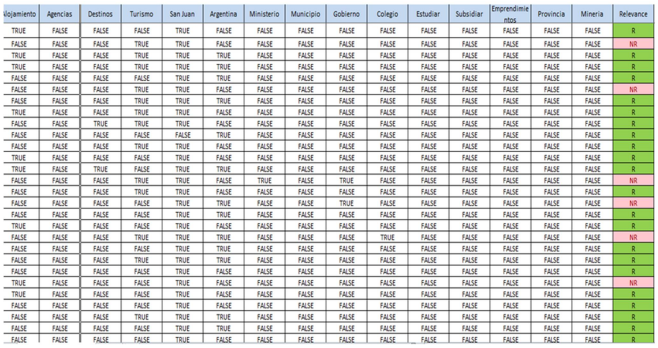

Fig. 5. Base de páginas para realizar la experimentación.

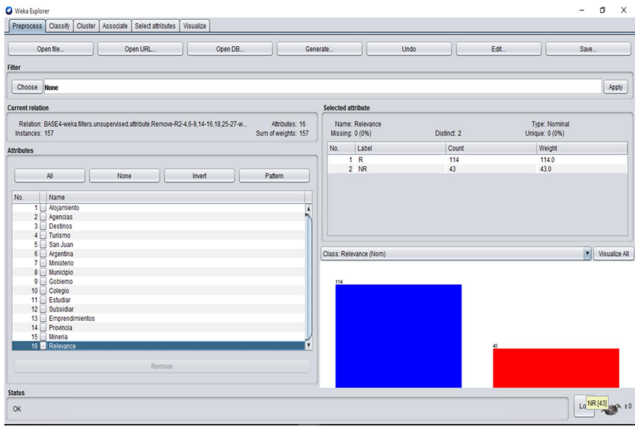

Fig. 6. Pre-procesamiento de la base en Weka 3.8.

	REGLAS					ARBOLES			BAYES		FUNCIONES	METAS	
	ZeroR	PART	JRip	Decision Table	OneR	JR48	LMT	Random Tree	Naives Bayes	Bayes Net	Logistic	Simple Logistic	Cost Sensitive Classifier	Attribute Select Classifier
% CC	72,61	84,71	83,44	80,89	73,89	82,80	89,81	84,71	84,08	85,35	92,36	89,81	84,71	84,08
% IC	27,39	15,29	16,56	19,11	26,11	17,20	10,19	15,29	15,92	14,65	7,64	10,19	15,29	15,92

Fig. 7. Comparación de técnicas de clasificación en Weka 3.8.

Para llevar a cabo la tarea de agrupar, en la etapa de *Agrupamiento,* y en función de la distancia entre cada página y los centros se hicieron pruebas con la similitud del coseno, con la distancia euclídea, calculando la diferencia entre dos páginas en función de un término específico y con la similitud propuesta por la metodología; codificadas en PHP. En Fig. 8 puede observarse que podemos elegir la similitud o distancia y el agrupamiento deseado y se muestra un listado de páginas web que se encuentran en ese grupo.

En la Fig. 9 y Fig. 10 puede observarse como la similitud del coseno y la distancia euclídea, respectivamente, permiten que páginas que no tienen información sobre un grupo estén ubicadas en ese grupo. Por ejemplo, la similitud del coseno, en el grupo Gastronomía donde el centro es la página 25 (0.72500 CO) consideró como similar al centro las páginas 3, 4, 5, 11,

12,…, 17 que no tienen información al respecto (0.00000 CO). Los mismo sucede con la distancia euclídea. Sin embargo con la similitud propuesta puede observarse que no sucede esto, ya que si bien en la columna del grupo Gastronomía pueden observarse valores en cero sólo los valores que figuran con RG son los que se encuentran dentro del grupo (Fig. 11).

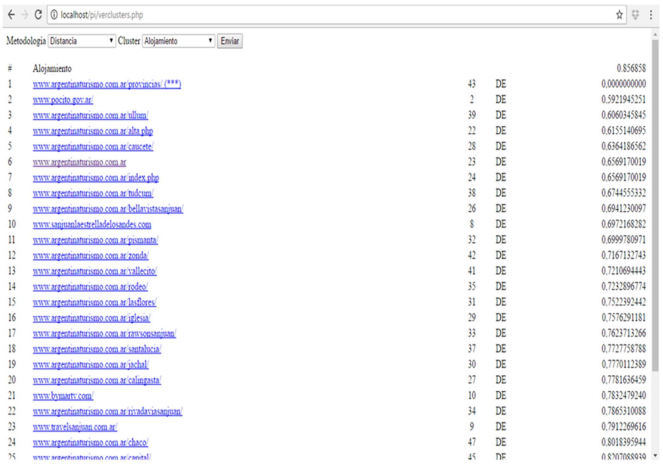

Fig. 8. Selección de la métrica para calcular la distancia o similitud y grupo.

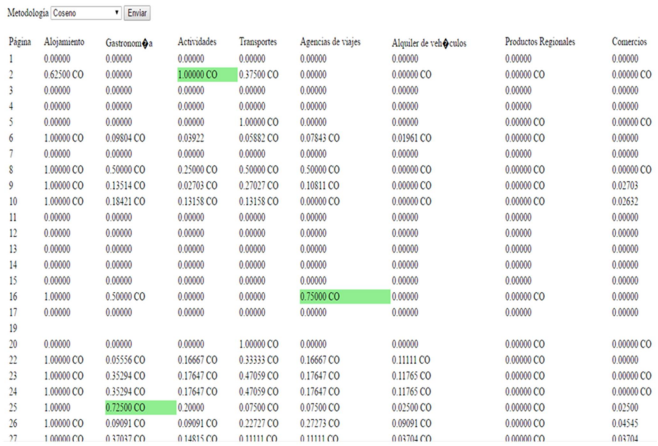

Fig. 9. Ubicación de las páginas en los grupos con su respectivo peso en función de la similitud del coseno.

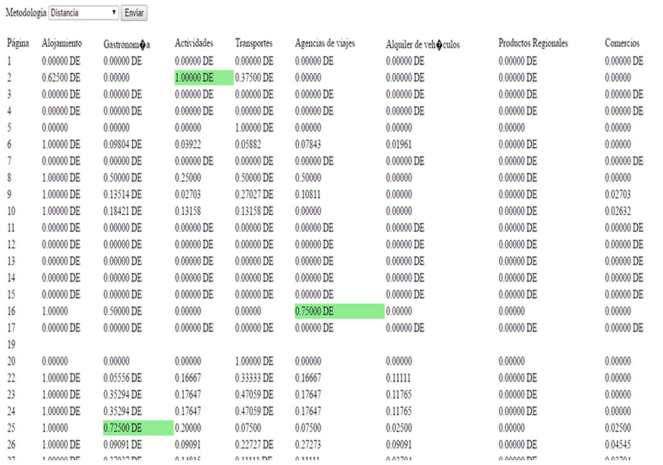

Fig. 10. Ubicación de las páginas en los grupos con su respectivo peso en función de la distancia euclídea.

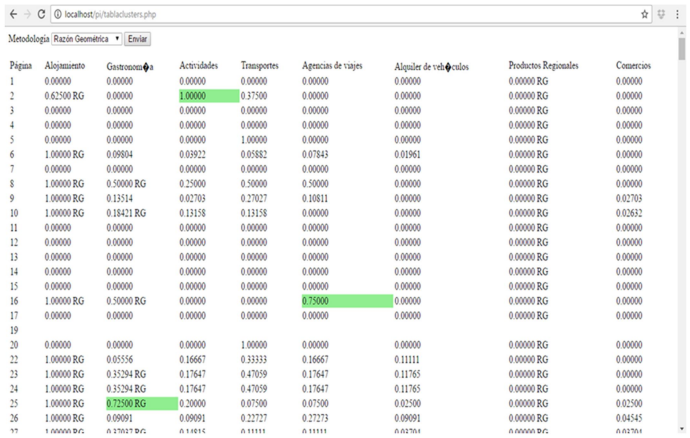

Fig. 11. Ubicación de las páginas en los grupos con su respectivo peso en función de la similitud propuesta.

V. Discusión

Según Lui en el clustering de documentos web los documentos pueden ser considerados como bolsas de palabras, donde la secuencia y la posición de las palabras no se tienen en cuenta. Cada documento es visto como un vector, es decir usando el modelo de espacio vectorial (VSM). Generalmente, se calcula la similitud entre documentos a través de la similitud del coseno, más bien que la distancia entre los vectores que representan esos documentos [30]. Sin embargo algunos autores han encontrado algunas falencias al aplicar esta métrica. Así, en [31] propusieron extender la medida de similitud del coseno, considerando la distancia Mahalanobis.

Por su parte, en [32] plantean que la similitud del coseno tiende a sesgarse a aquellos términos con alta frecuencia. En este trabajo los autores proponen como alternativa el cálculo de la distancia de la medida ponderada del coseno usando la distancia de Hamming con lo cual se cuenta cuantos términos de dos vectores no comparten, en cuyo caso se baja la similitud entre ambos. Por último, en [33] proponen modificar la forma de calcular la similitud entre cada par de características considerando una matriz de similitudes entre ellas, la cual es agregada al VSM.

En este trabajo se consideraron las páginas web para establecer similitud/distancia entre ellas. Para determinar la similitud (o distancia), en cuanto a cantidad de información que ofrecen las páginas, cualquiera de los métodos presentó problemas al ser aplicados. Proveían una similitud (o distancia) general entre dos vectores, por lo tanto cuando se necesitaba puntualizar la similitud (o distancia) para un término específico entre dos páginas no se obtenían verdaderos resultados. Se pensó en una primer alternativa calculando la diferencia entre dos páginas en función de un término específico, tomando el peso w_{ij} de cada término en cada página y compararlo con el peso del centro correspondiente. Si bien esta forma proporcionaba resultados más reales, no subsanaba el inconveniente de contar en un grupo con páginas que no tengan información de un término determinado, debido a que al calcular la diferencia entre ambos ésta podría ser menor o igual a la del umbral y sin

embargo la página en cuestión no contenía información de ese grupo. La segunda alternativa, calcular la razón geométrica entre el peso del término t_i de la página que se está analizando y el centro c_i, resulta más apropiada debido a que cuando se busca información y se calcula la similitud sólo interesa la información de ese término y no el resto que tiene esa página.

Fundamenta, aún más, esta última propuesta el hecho que puede darse el caso que dos páginas resulten similares por la cantidad de información que presentan en promedio, pero que para un término en particular sean muy poco o nada similares, con lo cual se puede cometer el posterior error de mostrar una página que no presente nada o casi nada de información para determinado término determinado.

VI. Conclusión

En este trabajo se propone un método que permite agrupar recursos web, aún con solapamiento entre grupos.

Adicionalmente, se presenta una metodología para recuperar información web, de la cual forma parte dicho método.

La investigación fue motivada por la posibilidad de aportar precisión, simplicidad y rapidez ante la búsqueda de recursos web relacionados con un dominio establecido a priori.

Proponer un mecanismo que concentre información precisa y renovada se considera un aporte importante debido a que en la actualidad la mayoría de los casos la búsqueda se vuelve una tarea compleja para los usuarios web. La propuesta no pretende competir con los actuales buscadores; hace uso de los mismos como parte de la metodología para indexar páginas web. La mejora que se promueve es hacer análisis, recolección, clasificación y agrupamiento automático con solapamiento de información, aplicando un método sencillo que sólo tienen en cuenta la similitud entre el término y el centro, sin considerar una similitud general como lo hacen los métodos estándares. De esta forma la tensión, angustia y agotamiento del usuario al buscar en una inmensa biblioteca digital se ven reducido a buscar sólo en nuestras bases de datos, haciendo la búsqueda más eficiente y efectiva. A su vez, se aprovecha la capacidad de la búsqueda e índices de los motores de búsqueda existentes.

La experimentación se realizó en el dominio del turismo por ser uno de los dominios con mayor cantidad de información a manejar y que se actualiza continuamente. Al proponer grupos solapados, es decir admitir que una página pueda encontrarse en la intersección de dos o más grupos, permite escoger rápidamente aquellas páginas que sólo brindan información explícita de un término solicitado por el usuario o seleccionar aquellas páginas que ofrecen información de más de un término ubicando la intersección de los grupos comprometidos.

El desafío actual de una aplicación que pretenda brindar recursos concisos y unificados va más allá de ubicar ese recurso y agruparlo, procurando establecer un agrupamiento que brinde información precisa y acorde a las necesidades del usuario.

Referencias

[1] M. Li, and S. Cao, “A serie method of massive information storage, retrieval and sharing”. Mechatronics and Automation (ICMA), 2014 IEEE International Conference on, pp. 1171-1175, 2014.

[2] J. Edosomwan, and T. O. Edosomwan, “Comparative Analysis of Some Search Engines”. South African Journal of Science, vol. 106, no 11-12, pp.1-4, Oct. 2010.

[3] B. Satatistica. “Total Number of Pages Indexed by Google”. http://www.statisticbrain.com/total-number-of-pages-indexed-by-google. 2016.

[4] B. Satatisticb. “Google Annual Search Statistics”. http://www.statisticbrain.com/google-searches/, 2016.

[5] P. Baldi, P. Frasconi, and P. Smyth, “Modeling the Internet and the Web. Probabilistic Methods and Algorithms”. West Sussex PO19 8SQ, England: John Wiley & Sons, Ltd. The Atrium, Southern Gate, Chichester, p. 20-22, 2003.

[6] C. D. Manning, P. Raghavan, and H. Schütze, “Introduction to Information Retrieval”, Cambridge: Cambridge university press, vol. 1, no. 1, p. 496, 2008.

[7] X. Qi, “Web Page Clasification and Hierarchy Adaptation". http://wume.cse.lehigh.edu/pubs/qi-dissertation.pdf, 2012.

[8] C. D. Manning, P. Raghavan, and H. Schütze, “Introduction to Information Retrieval”, Cambridge: Cambridge university press, vol. 1, no. 1, p. 350, 2008.

[9] S. A. Özel, “A Web Page Classification System Based on a Genetic Algorithm Musing Tagged-Terms as Features”. Expert Systems with Applications, vol. 38, no 4, pp. 3407-3415, April 2011.

[10] E. Baykan, M. Henzinger, and I. Weber, “A Comprehensive Study of Techniques for URL-Based Web Page Language Classification”. ACM Transactions on the Web (TWEB). Vol. 7, no 1, article 3, March 2013.

[11] R. Chen, C. Bau, and M. Tsai, “Web Pages Cluster Based on the Relations of Mapping Keywords to Ontology Concept Hierarchy”. International Journal of Innovative Computing, Information and Control, vol. 6, no 6, pp. 3407-3415, 2010.

[12] D. Ramage, P. Heymann, C. D. Manning, and H. Garcia Molina, “Clustering the Tagged Web”. In Proceedings of the Second ACM International Conference on Web Search and Data Mining. February 9-12, Barcelona, Spain: ACM, 2009.

[13] M. Shelke, K. Sadavarte, R. Dhurjad, and N. Pandit, “Improved Web Page Clustering Using Words and Tags”. 1° International Conference on Recent Trends in Engineering & Technology. Special Issue of International Journal of electronics, Communication & Soft Computing Science & Engineering, pp. 25-28, March, 2012.

[14] D. Patel, and M. Zaveri, “A Review on Web Pages Clustering Techniques”. In Trends in Network and Communications, SpringerBerlin Heidelberg, pp. 700-710, 2011.

[15] J. Liu, C. Yu, W. Xu, and Y. Shi, “Clustering Web Pages to Facilitate Revisitation on Mobile Devices”. In Proceedings of the 2012 ACM international conference on Intelligent User Interfaces, pp. 249-252, 2012.

[16] S. Ghosh S, and D. S. Kumar, “Comparative Analysis of K-means and Fuzzy C-means Algorithms”. International Journal of Advanced Computer Science and Applications (IJACSA), vol. 4, no 4, pp.35-39, 2013.

[17] A. M. Sote, and S. R. Pande, “Web Age Clustering Using Self-Organizing Map”. International Journal of Computer Science and Mobile Computing, vol. 4, no 1, pp. 78-84, 2015.

[18] T. Matsumoto, and E. Hung, “Fuzzy Clustering and Relevance Ranking of Web Search Results with Differentiating Cluster Label Generation”. In Fuzzy Systems (FUZZ), 2010 IEEE International Conference on, pp. 1-8, July 2010.

[19] L. Xiao, L., and E. Hung, “Clustering Web-Search Results Using Transduction-Based Relevance Model”. In IEEE 1st pacific-asia workshop on web mining and web-based application, 2008.

[20] V. Hegde, "Web Pages Clustering: A New Approach". International Journal of Innovate Technology & Creative Engineering, vol. 1, no 4, pp. 42-44, April, 2011.
[21] I. Hernández, C. Rivero, D. Ruiz, and R. Corchuelo, "CALA". Journal Knowledge-Based Systems, ACM, vol.115, pp. 130-143, May 2016.
[22] L. Yue, W. Zuo, T. Peng, Y. Wang, Y., and X. Han, "A Fuzzy Document Clustering Approach Based on Domain-Specified Ontology". Journal Data & Knowledge Engineering, v. 100, PA, pp. 148-166, November 2015.
[23] R. Rekik, and I. Kallel, "Fuzz-Web: A Methodology Based on Fuzzy Logic for Assessing Web Sites". International Journal of Computer Information Systems and Industrial Management Applications, vol. 5, pp. 126-136, 2013.
[24] A. P: García-Plaza, V. Fresno, and R. Martínez, "Web Page Clustering Using a Fuzzy Logic Based Representation and Self-Organizing Maps". In Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, vol. 01, pp. 851-854, IEEE Computer Society, December 2008.
[25] M. R. Romagnano, S. V. Aciar, and M. G. Marchetta, "Method to Reduce Complexity and Response Time in a Web Search", International Journal of Information Technologies and Systems Approach, vol. 8, no 2, pp. 32-46, July-December 2015.
[26] M. R. Romagnano, P. I. Dominguez, M. G. Marchetta, and S. V. Aciar, "Reduciendo la Complejidad de Búsqueda Web en Base a las Necesidades del Usuario", http://conaiisi2015.utn.edu.ar/memorias.html, 2015.
[27] M. R. Romagnano, and M. R. Marchetta, "WIREE. Propuesta de una Metodología de Recuperación de Información Web Eficaz y Eficiente para un Dominio Específico", http://www.ucasal.edu.ar/conaiisi2016/book/memorias.html, 2016.
[28] B. Liu, "Web Data Mining – Exploring Hyperlinks, Contents and Usage Data". ISBN-10 3-540-37881-2. Springer-Verlag Berlin Heidelberg, p.189, 2007.
[29] J. Han, and M. Kamber, "Data Mining: Concepts and Tech-niques". Segunda Edición. Elsevier, pp. 402-408, 2006.
[30] B. Liu, "Web Data Mining – Exploring Hyperlinks, Contents and Usage Data". ISBN-10 3-540-37881-2. Springer-Verlag Berlin Heidelberg, p.138, 2007.
[31] K. Mikawa, T. Ishida, and M. Goto, "A proposal of extended cosine measure for distance metric learning in text classifica-tion". In Systems, Man, and Cybernetics (SMC), IEEE International Conference on, pp. 1741-1746, 2011.
[32] B. Li, and L. Han, "Distance weighted cosine similarity measure for text classification". In International Conference on Intelligent Data Engineering and Automated Learning, Springer Berlin Heidelberg, pp. 611-618, 2013.
[33] G. Sidorov, A. Gelbukh, H. Gómez-Adorno, and D. Pinto, "Soft similarity and soft cosine measure: Similarity of features in vector space model". Computación y Sistemas, vol, 18, no 3, pp. 491-504, 2014.

María R. Romagnano es Docente/Investigador del Instituto de Informática de la Facultad de Ciencias Exactas, Físicas y Naturales, Universidad Nacional de San Juan. Ha recibido el título de Licenciada en Sistemas de Información en el año 2002, en la Universidad Nacional de San Juan, Argentina y su título de Magister en Informática en el año 2010 en la Universidad de la Matanza, Argentina. Actualmente es doctoranda del Doctorado en Ingeniería de la Universidad Nacional de Cuyo, Mendoza, Argentina. Ella ha sido beneficiada con tres becas, iniciación y perfeccionamiento (CICITCA) y doctoral (Agencia Nacional de Promoción Científica y Tecnológica). Ha participado en diez proyectos de investigación y desarrollo y uno de extension. Además, ha participado en veinticinco publicaciones (doce como primer autor). Sus intereses en investigación abarcan las áreas de Inteligencia Artificial e Ingeniería de Software.

Martín G. Marchetta es Docente/Investigador en la Facultad de Ingeniería, Universidad Nacional de Cuyo, Mendoza, Argentona. Ha recibido el título de Ingeniero en Sistemas, en el año 2002, en la Universidad Tecnológica Nacional, Facultad Regional Mendoza y su título de Master Design Global - Mention Recherche en Innovation et Conception Intégrée,en el año 2009, en ENSGSI - Institut National Polytechnique de Lorraine, France. Se ha doctorado como Doctor en Ingeniería en el año 2009, en la Universidad Nacional de Cuyo, Mendoza. Sus intereses en investigación son: Inteligencia Artificial. Investigación Operativa. Optimización. Informática Industrial. Sistemas de manufactura. Logística y Supply Chain Management. Gestión e ingeniería de la innovación. Gestión de Conocimiento. Sistemas mecatrónicos.