Recuperación de información basada en técnicas de minería Web

Fabián Favret[1], Raúl Montiel[2], Victor Alvarenga[1], Matias Barboza[1], Leandro Witzke[1],
[1]*Univer. Gastón Dachary. Av. López y Planes 6519, Posadas, Misiones.*
[2]*Univ. Tecnológica Nacional, Facultad Regional Resistencia. French 414, Resistencia, Chaco.*
{fabianfavret, raulmontiel, matias.mbz , leanwitzke}@gmail.com, alva_victor@hotmail.com

Abstract

Obtener información exacta y relevante a través de los motores de búsquedas no supone una tarea sencilla debido a la existencia de factores tales como la cantidad de información en páginas y documentos Web que no cuentan con una estructura única y fija. Por otro lado, existen circunstancias del lado del usuario que por el momento un motor de búsqueda de información no lo puede resolver en su completitud como la subjetividad y la dificultad de precisar la necesidad de información en el lenguaje de la consulta.

En éste trabajo se desarrolla el módulo de minería web para la recuperación de información y documentos web y obtener resultados relevantes en un orden correcto de acuerdo a su contenido.

Palabras clave: grandes volúmenes de datos, minería de Web, recuperación de información, clasificación de documentos

1. Introducción

La aparición de Internet y los enormes avances tecnológicos implicaron grandes cambios en las áreas de almacenamiento, búsqueda y recuperación de información, debido al desarrollo de nuevas técnicas y métodos que dieron lugar a herramientas de gran potencia y utilidad. Una de ellas muy conocida y utilizada en la actualidad son los motores de búsqueda, creados con la finalidad de recuperar información almacenada en la Web y presentarla de acuerdo a las necesidades del usuario.

Más allá de lo extensivo de la utilización de los motores de búsqueda, obtener información exacta y relevante no es una tarea sencilla debido a la existencia de diferentes factores tales como (a) la cantidad de información desorganizada, es decir, documentos Web que no cuentan con una estructura única y fija; (b) información publicada en diferentes tipos de medios (texto, audio, gráficos) con diferentes formatos, en diferentes idiomas y alfabetos; (c) la priorización de visualización de resultados que tienen patrocinio; y (d) el modelo de consulta–respuesta que requiere redefinir la búsqueda constantemente [1]. Por otro lado, existen circunstancias del lado del usuario que por el momento un motor de búsqueda de información no puede resolver como son la subjetividad y la falta de precisión en las consultas[2][3].

Mediante diversos estudios se demostró que el 71,33% de los clics realizados en el buscador de Google son efectuados en los primeros diez resultados de la página[4]. Esto demuestra la importancia del proceso de clasificación de los documentos Web dado un requerimiento de búsqueda. Por ello, las empresas intentan posicionar sus páginas entre los primeros resultados de las búsquedas. Para lograra esto, existen básicamente dos maneras de posicionar una página en las primeras posiciones sin tener en cuenta la calidad de su contenido. La primera es utilizar enlaces patrocinados que consisten en páginas que tienen un cierto privilegio para un determinado segmento mediante un aporte monetario. La segunda es utilizar técnicas denominadas Black Hat SEO[5], que tiene como objetivo intentar engañar a los motores de búsquedas para adquirir las primeras posiciones en una determinada búsqueda. Dichas técnicas varían en el tiempo debido a las actualizaciones de los algoritmos. Entre las técnicas más destacadas podemos nombrar:

(a) ***Texto oculto***: consiste en crear contenido oculto en la misma página y el texto está del mismo color que el fondo de la página y así el usuario no puede ver el texto pero los buscadores sí y consideran que es contenido de la página;

(b) ***Redes de enlaces***: se pueden considerar como redes de portales Web que se intercambian enlaces unas con otras con la única finalidad de incrementar la popularidad de los enlaces y así aumentar el puntaje PageRank;

(c) ***Contenido duplicado***: se trata de crear varios portales Web con el mismo contenido o con contenido que es muy parecido, y lo que más cambia en estas páginas duplicadas son los nombres con los que se realizan los enlaces.

Claramente entonces, existe la posibilidad de que las páginas devueltas por un motor de búsqueda y clasificadas como las más importantes no tengan la información solicitada y con el grado de relevancia adecuado.

Otro factor que se suma a estas cuestiones es que la mayoría de los usuarios no están familiarizados con las búsquedas mediante restricciones[6][7] desaprovechando las herramientas brindadas para mejorar la relevancia de los resultados.

Para hacer frente a estas cuestiones se estudian y desarrollan métodos para buscar información de valor en grandes volúmenes de datos. Esta área se conoce como Minería de Datos (MD) y se enfoca en la generación de algoritmos que permiten extraer conocimiento desde los datos[1]. De esto surgen las técnicas de Minería de Web, una metodología de recuperación de información que permite procesar y capturar información útil de páginas Web y documentos en Internet[8].

Este trabajo muestra el desarrollo e implementación de mecanismos de clasificación de documentos Web relevantes de acuerdo a requerimientos específicos de búsqueda.

El artículo se estructura de la siguiente manera: en la Sección 2 se describen los conceptos básicos de Minería de Web; en la Sección 3 se explica el modelo propuesto; en la Sección 4 de exponen los resultados obtenidos y finalmente, en la Sección 5, se mencionan algunas conclusiones obtenidas.

2. Minería Web

Uno de los sistemas con mayor publicación de datos sin restricciones de su contenido y de acceso libre es la Web. Ésta, tiene características únicas[1] como (a) la existencia de distintos tipos de datos (audios, videos, textos, etc), (b) la información en las páginas Web es muy variada, dinámica y tiene ruido, (c) una pequeña cantidad de información está enlazada y (d) es de servicios y también es una sociedad virtual en la cual extraer información útil conlleva a una serie de problemas de índole multi-disciplinaria.

La Minería de Web (MW) es el conjunto de técnicas que permiten abarcar la problemática de explotación de la información que se encuentra en la Web[1]. El proceso de MW puede ser definido formalmente como “el proceso global de descubrir información o conocimiento potencialmente útil y previamente desconocido a través de los datos de la Web”[9]. La MW tiene como objetivo descubrir información útil o el conocimiento tanto del contenido de documentos Web, como también de la estructura de hipervínculos Web y los datos de uso.

Las tareas de MW se clasifican en tres categorías: MW de contenido, MW de estructura y MW de uso. La primera extrae información del contenido en los documentos Web. La segunda, MW de estructura, descubre un modelo a partir de la topología de enlaces de la red. Este modelo puede ser útil para clasificar o agrupar documentos. Y por último VM de uso extrae información (hábitos, preferencias, etc. de los usuarios o contenidos y relevancia de documentos) a partir de las sesiones o registros de uso en la Web[9].

Este proceso de MW no es trivial por lo cual se plantean diversas teorías establecidas por áreas como la recuperación de la información (RI), la extracción de información (EI), el procesamiento del lenguaje natural (PLN) o la Minería de Texto (MT), entre otras

La búsqueda en la Web necesita de diferentes técnicas de RI cuyo campo de estudio ayuda al usuario a encontrar la información necesaria dentro de una gran colección de documentos de texto[9]. RI consiste básicamente en encontrar un conjunto de documentos relevantes para la consulta hecha por el usuario. Mientras que por otro lado también suele ser utilizado un Ranking que puntúa de acuerdo a la relevancia que posee el documento evaluado respecto a la consulta realizada. Técnicamente, RI estudia la adquisición, organización, almacenamiento, recuperación y distribución de información [1].

Un sistema de RI Web (SRI) incluye un analizador de consultas que interpreta texto plano y separa una cadena de texto en tokens de palabras individuales y elimina aquellas palabras que no se consideren relevantes respecto a la consulta o necesidad de información requerida. Utiliza modelos de matemáticos cuya función es la de efectuar una puntuación de documentos a modo de lograr crear un índice ordenado, desde la mejor puntuación a la peor (Ranking), según su grado de relevancia. Lo cual implica que los primeros documentos sobre los que se comienza un análisis de contenido son aquellos que posean las mejores puntaciones[1].

Una vez que se hallan sitios Web con contenido de acuerdo a requerimientos específicos de búsqueda, el sistema debe ser capaz de obtener todo lo relevante utilizando mecanismos de extracción de contenido Web (Web Scraping), y al mismo tiempo adjuntar información de indexación a los registros para mantener una estructura que facilite futuros accesos a los recursos Web de una manera más directa, sin la necesidad de analizar ni ponderar contenido de una página que ya ha sido considerado relevante[1]. Para guiar la navegación de enlaces con el objetivo de localizar de manera eficiente las páginas de destino de gran relevancia, se utilizan técnicas de Web Crawling; donde, cada página que es escaneada es dada a un cliente que guarda las páginas, crea un índice para las páginas o guarda y analiza los contenidos de las páginas [10].

3. Modelo propuesto

El modelo propuesto tiene dos módulos bien definidos. El primero es un proceso que da soporte a la recopilación de los requerimientos de usuario y el segundo genera un proceso de búsqueda continua de recursos en la web mediante MW. Ambos procesos trabajan coordinadamente en un mismo flujo. La idea del primer proceso

es orientar al usuario en el armado de varias claves que serán tomadas como punto de partida de la búsqueda y la del segundo es proveer al usuario recursos a medida que se vayan encontrando, estos recursos son sincronizados entre ambos módulos poder presentárselos al usuario en una interface del sistema. Luego, para determinar la relevancia de los documentos Web recuperados se implementan cuatro algoritmos: C-Rank, enfoque ponderado, espacio vectorial y Okapi [11][12].

3.1. Esquema general

El esquema general del sistema puede observarse en la Fig. 1. El usuario comienza interactuando con la interfaz del sistema de recopilación de requerimientos (SRR), aquí se define el tema principal, luego mediante una guía de preguntas estándares se refina el concepto a buscar, de esta manera el sistema construye las claves de búsqueda.

Esas claves de búsquedas se al proceso de generación de URL's, que consulta a las interfaces públicas de cuatro buscadores: Google, Bing, Intelligo (un buscador de patentes) y Msmlx Excite (un metabuscador). Las primeras diez URLs de cada uno son enviadas al módulo unificación y ordenamiento de URL's. Éstas son categorizadas y ordenadas en una sola lista en base a los requerimientos de usuario (de esa lista se quitan las URLs repetidas). A partir de este punto son enviadas al módulo de MW que comienza un proceso continuo de inspección de enlaces y recursos tomando como base cada una de las URLs de la lista.

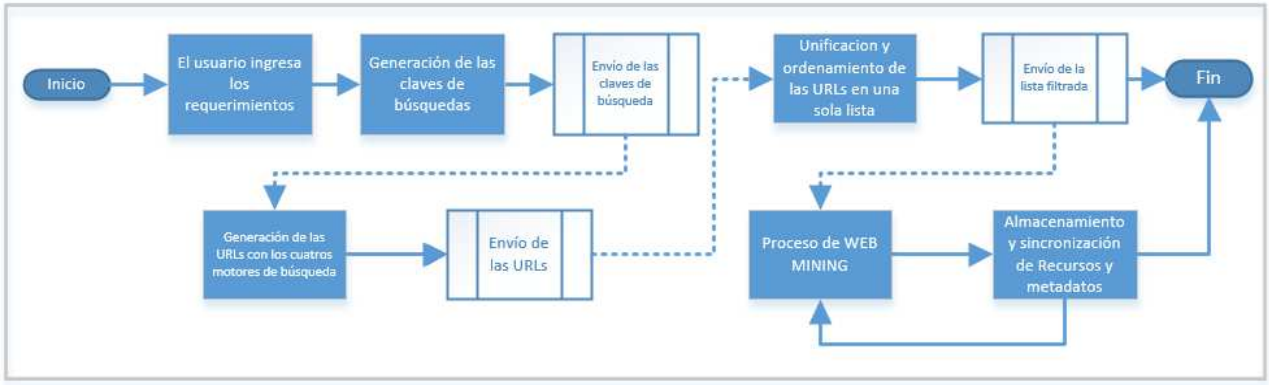

Fig. 1 - Esquema general del sistema

A medida que se encuentran recursos que se consideran relevantes para el usuario, se los almacena en un directorio junto con los metadatos del recurso y luego ambos son sincronizados. Por las características de este proceso continuo (lo que genera un gran volumen de información), se seleccionan solamente cincuenta recursos que se le muestran al usuario para que los valore.

3.2. Componentes del modelo

Un proceso muy importante en el modelo propuesto es la ponderación de los documentos web recuperados a fin de obtener una lista ordenada de acuerdo a su nivel de relevancia. Para esto, se evalúa la lista de documentos web obtenidos mediante cuatro algoritmos de ranking diferentes, generando una lista rankeada separada por cada uno de los mismos. Por cada documento web, se computa la posición en la que está ubicado en cada lista, y luego se suma para obtener un puntaje final. Una vez calculados todos los puntajes, se confecciona una única lista resultante ordenada de menor a mayor por el puntaje obtenido. A continuación, se explican en qué consiste cada uno de los algoritmos implementados.

3.2.1 Modelo de espacio vectorial

Es el modelo de IR quizás más conocido y el más utilizado. Plantea la necesidad de utilizar una función de similitud entre el documento y la consulta introduciendo un ranking en los documentos recuperados[1]. Cada documento de la colección está representado por un vector t-dimensional, donde t es la cardinalidad del conjunto de términos del corpus de documentos y cada elemento del vector posee un peso del término asociado a esa dimensión. Debido a esto, cualquier documento puede ser representado por un vector en este espacio vectorial. Al igual que los documentos, la consulta también es representada como un vector en este espacio vectorial. Tanto la asignación de los pesos a los términos en los documentos como el cálculo de similitud se puede realizar de distintas maneras[11]. Debido a que en este modelo un documento se representa como un vector de pesos, el primer paso es obtener estos valores, donde el peso es calculado mediante cálculos de frecuencias, generalmente usando el esquema $TF * IDF$, que plantea establecer una relación entre la frecuencia de un término dentro de un documento y su frecuencia en los documentos de la colección. Básicamente, obtiene la frecuencia pura del término t_i en el documento $d_j(TF)$ y lo multiplica por la inversa de la cantidad de documentos de la colección donde aparece $t_i(IDF)$. Los pesos de cada elemento del

vector no son únicamente 0 y 1 como en el conocido Modelo Booleano, sino que puede ser cualquier valor [1][11]. A continuación, para asignar una puntuación numérica a un documento para una consulta determinada que represente su relevancia, el modelo mide la similitud entre el vector consulta y el vector documento. El ángulo entre dos vectores se utiliza como una medida de la divergencia entre los vectores y la similitud por medida del coseno del ángulo es el más popular de las medidas de semejanza (ya que tiene la propiedad útil que es 1 para vectores idénticos y 0 para vectores ortogonales). Con los valores de similitud obtenidos de cada documento de la colección se arma una lista descendente que representa el ranking de los documento recuperados en orden de relevancia con respecto a la consulta del usuario[1][11]. El cálculo de las medidas de similaridad para cada documento de la colección es:

$$Sim\,(Q, D_i) = \frac{\sum_{j=1}^{t} w_{q_j} * w_{d_{ij}}}{\sqrt{\sum_{j=1}^{t} (w_{q_j})^2 * \sum_{j=1}^{t} (w_{d_{ij}})^2}} \qquad (1)$$

donde:
$D_i = W_{di1}, W_{di2}, \ldots W_{dit}$, es el vector que representa un documento.
$Q = W_{q1}, W_{q2}, \ldots W_{qt}$, es el vector de la consulta.
t, es el número de términos en la colección.
w_{ij}, es el peso del término j en el documento i.
w_{qj}, es el peso del término j en la consulta q.

3.2.2 Modelo Okapi

El modelo Okapi es un modelo probabilístico que incorpora la frecuencia de términos y la longitud de los documentos para su cálculo, cuyo esquema de pesos denominado BM25, se encuentra entre los más efectivos y junto al $TF * IDF$ es punto de referencia para el desarrollo y evaluación de nuevos modelos y nuevos esquemas de pesos. Esta función es una variación del modelo $TF * IDF$ usando un modelo probabilístico. Se basa en el concepto de "bolsa de palabras" (vocabulario de palabras de los documentos) en lugar de vectores como lo hace el Modelo Vectorial[1][13]. El cálculo para obtener el valor de relevancia de cada documento es:

$$Okapi\,(d_j, q) = \sum_{t_i \in q, d_j} \ln \frac{N - df_i + 0.5}{df_i + 0.5} \times \frac{(k_1 + 1) f_{ij}}{k_1 \left(1 - b + b \frac{dl_j}{avdl}\right) + f_{ij}} \times \frac{(k_2 + 1) f_{iq}}{k_2 + f_{iq}} \qquad (2)$$

donde:
t_i es un término.
$f_{ij,}$ es la frecuencia del término t_i en el documento d_j.
$f_{iq,}$ es la frecuencia del término t_i en la consulta q.
$df_{i,}$ es el nro. de documentos que contienen el término t_i.
$dl_{j,}$ es la longitud del documento (en bytes).
N, es el nro. total de documentos en la colección.
$avdl$, es la longitud media de todos los documentos de la colección.
k_1 (valor entre 1.0 y 2.0), b (usualmente 0.75) y k_2 (valor entre 1 y 1000) son parámetros constantes.

3.2.3 Modelo enfoque ponderado

El modelo Enfoque Ponderado tiene como característica principal el uso de un diccionario de dominio. Consta de tres etapas, donde en primer lugar se realiza un preprocesamiento de todos los documentos a rankear. Si bien este proceso en la mayoría de los casos es indistinto al método de ranking aplicado, se lo incluye debido a que contribuye a obtener mejores resultados. Aquí intervienen tres pasos, eliminación de palabras de parada, stemming y tokenization[14]. La segunda etapa consiste en calcular la frecuencia de todos los términos de cada documento y construir un diccionario de dominio aplicado a un campo determinado, en este caso, que tenga relación con los documentos a clasificar.

La última etapa consiste en obtener todos los términos que aparecen en el documento y verificar si se encuentran en el diccionario de dominio y/o en la clave de búsquedas. Para aquellos términos que aparecen tanto en el documento como en la claves de búsqueda, se le suma su frecuencia a un acumulador llamado Clave_Acierto, en cambio, si dicho termino aparece solamente en el diccionario de domino se suma a un acumulador llamado Acierto_Positivo y por el contrario, si el termino solo aparece en la clave de búsqueda, se suma la frecuencia a un acumulador llamado Acierto_Negativo. El puntaje final se calcula de la siguiente manera:

$$DRi = \frac{[ca(i) * \propto] + [ap(i) * \beta] + [an(i) * \gamma]}{ca(i) + ap(i) + an(i)} \qquad (3)$$

donde:
ca representa clave_acierto,
ap como acierto_positivo y
an como acierto_negativo.

Los valores de las constantes son los siguientes [14]:$\alpha = 1$, $\beta = 0{,}75$ y $\gamma = 0{,}50$.

3.2.4 Modelo C-Rank

El modelo de C-Rank está orientado a la colaboración de contenidos entre documentos con contenidos similares. Se basa en la idea de que el autor de un documento Web agrega enlaces hacia otros documentos para com-

plementar la falta de información (exceptuando banners de publicidad). Por ende, si un documento Web esta enlazado a varios documentos, está demostrando que posee valor en ese tema en particular.

El C-Rank de un término en un documento es definido por la suma de las puntuaciones de relevancia y una proporción de la puntuación de contribución del término en otros documentos.

$$CR_t(\mathrm{p}) = \lambda R_t(p) + (1-\lambda)\sum_{d}\sum_{q\in D(p,d)} \alpha_t^d(p,q) R_t(q) \qquad (4)$$

donde:

$CR_t(p)$ representa el C-Rank del termino t en un documento p,

$R_t(q)$ es el puntaje de relevancia de la página p para el término t,

$\propto_t^d (p,q)$ es el porcentaje de contribución del término t en la página p para la página q.

Es importante destacar que, a diferencia de otros modelos de colaboración como el PageRank, este modelo calcula el puntaje de colaboración con otros documentos teniendo en cuenta solo los términos de la clave de búsqueda, y no solamente por poseer una relación[12].

4. Pruebas y resultados

Para probar si los recursos encontrados tienen relación con los temas buscados se han realizado distintas pruebas. Los temas principales de búsqueda están relacionados con la seguridad informática en las aplicaciones Web en la que se incluyó: Cross-site scripting (XSS), Injection SQL, Illegal HTTP Version, Illegal Byte Code Character y Cookie Injection.

Para el proceso de validación se ha contado con un grupo de expertos en seguridad informática quienes han evaluado la relevancia de los documentos que son retornados por el sistema a partir de diferentes consultas.

Se ha verificado la pertenencia de cada documento con respecto a la consulta ingresada, teniendo en cuenta la relación directa entre sus términos y aquellos resultantes a partir del proceso de búsqueda. A partir de la combinación de esos factores se ha determinado qué porcentaje de los resultados serían considerados como efectivamente fiables para el usuario, sirviendo tal métrica como medida de desempeño del buscador.

Como las búsquedas son un proceso continuo, se realizaron durante 48 horas de iniciado el proceso y la cantidad máxima de recursos listados son los primeros cincuenta.

Se utilizó la interfaz del SRR con las siguientes claves de búsqueda que se muestran en la tabla 1. La fuente de información principal para obtener los recursos son los documentos (no se restringe la búsqueda y pueden ser cualquier clase de recursos como ser texto, audios, videos, etc.) y papers (restringe la búsqueda a publicaciones de investigaciones científicas).

Tabla 1. Claves de búsquedas

Tema	Consigna
XSS	Cross-site scripting, paper, documents ,Reflect, persistent
Injection SQL	SQL INJECTION, paper, documents blind
Illegal HTTP Version	Illegal HTTP Version, paper, documents, malformed version
Illegal Byte Code Character	Illegal Byte Code Character in Header, paper, documents, header
Cookie Injection	Cookie Injection, paper, documents, Tampering

El porcentaje de recursos obtenidos, ver la tabla 2, indican que se recuperaron más sitios cuya fuente de información son documentos. Estos resultados se deben al dominio de búsqueda ya que actualmente la cantidad de información "informal" es mucho mayor al a información que puede estar publicada en sitios de difusión académica.

Tabla 2. Porcentaje de recursos obtenidos

Tema	Fuente información	
	Porcentaje Documentos	Porcentaje Papers
XSS	96 %	4%
Injection SQL	98 %	2%
Illegal HTTP Version	98%	2%
Illegal Byte Code Character	100%	-
Cookie Injection	100%	-

En la Tabla 1 puede observarse que los documentos relacionados con el tema de búsqueda para todos los tópicos es elevado (al menos el 96% de los resultados coincide con la temática). Específicamente, las fuentes obtenidas para el tema XSS e Injection SQL son de carácter de páginas estándares, blog personales, organización como OWASP que es un proyecto abierto de seguridad para aplicaciones web y referente en el ámbito de la seguridad informática, W3SCHOOL (un sitio para desarrolladores web con tutoriales).

Para Illegal HTTP Version e Illegal Byte Code Character las fuentes obtenidas son de carácter de blog personales y páginas estándares. En cambio para Cookie Injection las fuentes obtenidas son de páginas estándares.

Claramente, relacionado con lo antes dicho sobre el dominio de las búsquedas, la mayor cantidad de fuentes

de información son de carácter no académico y muestra la ductilidad del modelo propuesto.

En cuanto al análisis cualitativo de los recursos obtenidos, los resultados se muestran resumidos en la Tabla 3 (específicamente, si son documentos que aportan contenido fiable). Por ejemplo si tomamos el tema XSS, del 96% de documentos obtenidos de la tabla 2, solamente el 92% son documentos pertinentes al tema buscado como se puede ver en la tabla 3.

Tabla 3. Análisis cualitativo de resultados

Tema	Fuente información	
	Porcentaje Documentos fiables	Porcentaje Papers fiables
XSS	92%	1%
Injection SQL	94%	1%
Illegal HTTP Version	85%	1%
Illegal Byte Code Character	84%	0%
Cookie Injection	0%	0%

En cuanto al primer tema (XSS), los documentos encontrados contienen: descripción en general, XSS reflect y persistent, ejemplos de explotación XSS estándares y avanzado y protección ante éstos ataques. Cómo el objetivo de la búsqueda en este tema era tener una descripción general y conocer los ataques XSS reflect y persistent los resultados son pertinentes ya que se obtuvo documentos de cómo protegerse ante estos ataques y forma de explotarlos.

Entre los resultados obtenidos se destaca una página de owasp.org (proyecto abierto de seguridad en aplicaciones Web) ya que contiene es una descripción del tema buscado, distintos tipos de ataques, cómo revisar el código, cómo proteger los sistemas de este tipo de vulnerabilidad y ejemplos de ataques más comunes.

Otro resultado interesante encontrado es un recurso de IBM que menciona como prevenirse de éstos ataques, mejores prácticas para desarrolladores Web y ejemplos de este tipo de vulnerabilidades.

Además, la página acunetix (un escáner de vulnerabilidades de aplicaciones web) tiene disponible información relacionada con los dos tipos de ataques buscado.

Por otro lado, del sitio de perl se pudo acceder a información sobre que es XSS y las soluciones para prevenir éstos ataques en perl. Claramente la relevancia de esta información es alta ya que son fuentes fiables y de consulta permanente en este área.

El segundo tema Injection SQL, los documentos obtenidos tratan de: una descripción general, ejemplos de inyección de SQL en distintas base de datos, inyección del tipo BLIND y formas de evitar inyección SQL. Nuevamente como el objetivo de ésta búsqueda era tener una descripción general del tema y conocer el tipo de ataque BLIND se puede apreciar que los resultados en ésta búsqueda también son pertinentes ya que se obtuvo información adicional de inyección de SQL en distintas bases de datos, formas de explotarlos y protegerse ante estos ataques.

Los resultados obtenidos en ésta sección también son de relevancia ya que se obtuvieron, por ejemplo, recursos de la página Web de Cisco que presentan una introducción y una explicación de la problemática de Injection SQL y como defenderse ante estos ataques.

Otros recursos encontrados están relacionados a los sitios owasp.org y acunetix, mencionados anteriormente. Oswap describe el tema, el modelo de amenazas, actividades relacionadas con éste tipo de vulnerabilidades, ejemplos y ataques relacionados con Injection SQL, esta información es muy útil para abordar el tema de seguridad. Acunetix también describe el problema y como funciona y, además, que puede hacer un atacante con Injection SQL.

Una aspecto interesante es que se ha obtenido como resultado un video de Youtube que explica que es y como atacar páginas Web con este tipo de vulnerabilidades. Claramente, la obtención de este tipo de recursos distintos a páginas web muestra la versatilidad del sistema.

El tercer y el cuarto tema estaban relacionados a la búsqueda de información sobre Illegal HTTP Version e Illegal Byte Code Character. El objetivo general de ésta búsqueda era tener una descripción general del tema y formas de explotarlos, observándose que los resultados son pertinentes debido a que se obtuvo información adicional como formas de protección.

Nuevamente los recursos obtenidos fueron principalmente de owasp.org y otras páginas Web que hacen una descripción general de los temas, las formas de tratarlos y evitar estos tipos de ataques utilizando Firewall Application Web.

Para el último tema (Cookie Injection) los documentos obtenidos no son de relevancia debido a que no tratan el tema en particular. Esto puede deberse a diferentes factores como ser el análisis del contenido dentro de la etiqueta HTML HEAD, la ambigüedad de la consulta o la falta de términos a excluir.

5. Conclusiones

Este trabajo presenta un modelo de búsqueda de recursos que se basa en técnicas de Minería de Web, compuesto por un módulo de Recopilación de Requerimientos de usuario y un módulo de exploración.

Esta separación en ambos módulos permite la independencia de las actividades relacionadas con dar soporte al usuario en la determinación de requerimientos y la implementación de los algoritmos y estrategias de búsqueda.

El modelo propuesto no hace una búsqueda tradicional consulta - respuesta, sino que los resultados obtenidos se refinan continuamente y categorizados según los requerimientos del usuario. En este sentido, los resultados obtenidos en las pruebas realizadas demuestran que el modelo recupera información de acuerdo a las definidas por el usuario de manera adecuada.

En general los recursos devueltos han sido satisfactorios porque además de obtener una idea general del tema y forma de explotarlo se obtuvo información de cómo protegerse ante estos ataques. Sin embargo en algunos casos (como el caso Cookie Injection) el sistema tiene inconvenientes para encontrar recursos relevantes.

Trabajos futuros

Actualmente se está trabajando en la depuración de los algoritmos presentados en este artículo. Además se ha comenzado el análisis de algoritmos semánticos y la administración de los resultados (para este último punto se tiene previsto implementar un indexador de archivos para brindar más calidad en la presentación de los resultados).

Agradecimientos

Este trabajo forma parte de los proyectos “Modelos de Análisis de Información para la Toma de Decisiones Estratégicas” (UGD) y “Análisis de Información en Grandes Volúmenes de Datos Orientado al Proceso de Toma de Decisiones Estratégicas” (cód: UTN4058).

Referencias

[1] Bing Liu, *Web Data Mining, Exploring Hyperlinks, Contents, and Usage Data*, First Edition. Chicago: University of Illinois, 2007.

[2] G. Tolosa and F. Bordignon, “Introducción a la Recuperación de Información,” *Univ. Nac. Lujan*, 2008.

[3] E. Abadal and L. Codina, “Recuperación de Información. Bases de Datos Documentales: Características, funciones y método,” 2005.

[4] P. Petrescu, “Google Organic Click-Through Rates in 2014,” 2014. [Online]. Available: https://moz.com/blog/google-organic-click-through-rates-in-2014. [Accessed: 08-Jan-2016].

[5] A. Michael and B. Salter, *Marketing Through Search Optimization*, 2nd ed., vol. 1. Elsevier Ltd, 2008.

[6] R. W. White and D. Morris, “Investigating the querying and browsing behavior of advanced search engine users,” *Proc. 30th Annu. Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, p. 255, 2007.

[7] F. Cacheda and Á. Viña, “Understanding how people use search engines : a statistical analysis for e-Business,” *CiteSeerx*, vol. 319, 2001.

[8] O. Etzioni, *The World Wide WEB: Quagmire or gold mine*, 11th ed., vol. 39. Commun. ACM, 1996.

[9] Jeria Victor and Heughes Escobar, “Minería Web de Uso y Perfiles de Usuario: Aplicaciones con Logica Difusa,” *Universidad de Granada*, 2007.

[10] Menczer Filippo, Pant Gautam, and Srinivasan Padmini, *Topical Web Crawlers: Evaluating Adaptive Algorithms*, vol. V. ACM Transactions on Internet Technology, 2005.

[11] Faustina Johnson and Santosh Kumar Gupta, *Web Content Mining Techniques: A Survey*, vol. 42, 11 vols. International Journal of Computer Applications, 2012.

[12] D.J. Kim, L. Sang-Chul, H.-Y. Son, S.-W. Kim, and J. B. Lee, *C-Rank and its variants: A contribution-based ranking approach exploiting links and content*, J. Inf. Sci., vol. 40, 6 vols. 2014.

[13] Baeza Yates and Berthier Ribeiro, *Modern Information Retrieval: The Concepts and Technology behind Search*, 2nd ed. 2010.

[14] S. S. Bama, M. S. I. Ahmed, and A. Saravanan, *Enchancing The Search Engine Results Through Web Content Ranking*, Int. J. Appl. Eng. Res., vol. 10, 5 vols. 2015.