Revista en Telecomunicaciones e Informática, Vol. 3, No. 5 p. 55 - 73
Medellín - Colombia. Enero - Junio de 2013, ISSN 2215-8200

Sistema multiagente para el filtrado de pornografía mediante la evaluación del contenido multimedial de las páginas web

Ana Isabel Oviedo, Catalina Andrea Manco, Juan Esteban Guerra
Facultad de Ingeniería Informática, Universidad Pontificia Bolivariana Medellín, Colombia
anaisaoviedo@gmail.com

Resumen
Las Tecnologías de la Información y las Comunicaciones (TIC), han contribuido al progreso de las telecomunicaciones, pero han generado peligros como la pornografía. Ante esta situación, los sistemas de filtrado permiten identificar los contenidos pornográficos de Internet. Sin embargo, estos sistemas no consideran toda la información multimedial de las páginas web, generalmente se basan sólo en el texto de las páginas. En este trabajo se propone realizar la evaluación de la información multimedial de las páginas web (imágenes, texto y estructura) mediante un sistema multi-agente (SMA's). El diseño del sistema multiagente se realiza por medio de la metodología MAS-CommonKADS y la implementación del prototipo se realiza utilizando la plataforma JADE. Finalmente, la evaluación del prototipo se realiza en términos de eficacia, usando las métricas: precisión, cobertura y exactitud. De la evaluación se concluye que, un sistema de filtrado que evalúe los diferentes contenidos multimediales tiene mejores resultados que los filtros web que evalúan sólo un tipo de contenido multimedial.

Palabras clave: Clasificación de páginas pornográficas, JADE, MAS-CommonKADS, Sistemas de filtrado, Sistemas multiagente.

I. Introducción

Existe actualmente una necesidad creciente de filtrar los contenidos nocivos e inapropiados de Internet, dada la cantidad de información y diversidad de contenido que este manipula. La integración de las Tecnologías de la Información y las Comunicaciones (TIC), ha contribuido al progreso de las comunicaciones, el acceso a la información y la movilidad, entre otras; pero también, debido a su naturaleza inherentemente distribuida y de difícil control ha generado peligros como el acceso a contenido nocivo y la aparición de nuevas tipologías delictivas, como la pornografía [1].

Un filtro de información proporciona una separación física entre los contenidos nocivos que están en Internet y los usuarios que están navegando y se activa para que los usuarios no puedan acceder a contenidos tales como: pornografía, racismo, sectas, drogas, entre otros [2]. Para realizar esta separación del contenido nocivo, los sistemas de filtrado utilizan diferentes métodos de clasificación de información como: filtros bayesianos, redes neuronales, árboles de decisión, método de Rocchio, máquinas de soporte vectorial, entre otros.

Ningún sistema de filtrado de información es perfecto, ya que a veces existe un filtrado excesivo (contenido bloqueado por error) o en otros casos un filtrado defectuoso (contenido inconveniente no bloqueado) [3]. Una de las causas de dichos problemas es que los sistemas de filtrado no consideran toda la información multimedial existente en las páginas web (texto, imágenes, audio, video).

Ante estas dificultades, la inteligencia artificial y los sistemas multi-agente (SMA's), permiten utilizar agentes para el filtrado de la información mediante su evolución a través del tiempo y su aprendizaje del entorno, optimizando los resultados obtenidos [4].

Para aportar a esta problemática, este trabajo propone el desarrollo de un prototipo de software de un sistema multiagente para el filtrado de páginas pornográficas mediante el análisis del contenido multimedial de las mismas, ya que la unión de varios agentes facilita la evaluación de diferentes recursos hipertextuales en los SMA's [5]. Con este fin se tienen en cuenta los resultados de diferentes proyectos, donde se han abordado las tareas de filtrado basado en el texto [6], filtrado basado en las etiquetas HTML [7], filtrado basado en las imágenes [8] y ensamble de los resultados obtenidos [6].

El resto del artículo está organizado de la siguiente manera: La sección II describe algunos conceptos sobre los sistemas de filtrado. En la sección III conceptualiza algunos métodos de clasificación de contenidos. En la sección IV se realiza una revisión literaria sobre los sistemas multiagente. En la sección V se realiza una caracterización del modelo de filtrado. En la sección VI se describe el diseño realizado para el sistema multiagente. En la sección VII se presenta la evaluación del prototipo y finalmente en la sección VIII se presentan las conclusiones y trabajos futuros.

II. Sistemas de filtrado

Los filtros son programas que evitan que los usuarios accedan a contenidos nocivos que están en la web. Algunas técnicas utilizadas actualmente para la selección y el filtrado de información son las herramientas de control de acceso y monitorización para el usuario y en el proveedor de Internet, con el uso del análisis semántico para el bloqueo de palabras claves aprovechando técnicas de inteligencia artificial, el control de accesos según perfiles y horarios de acceso, la clasificación y filtrado de contenidos, entre otras [9].

Los sistemas de filtrado web se clasifican en: basados en el conocimiento cuyos perfiles son definidos explícitamente por y para el usuario, basados en el comportamiento del individuo cuyos perfiles son definidos por el sistema según los comportamientos del usuario, y el filtrado colaborativo cuyos perfiles son definidos por el sistema para personas afines usando técnicas de minería de datos [10].

Diversas compañías han desarrollado sistemas de filtrado web, algunas de las deficiencias que se han encontrado en los modelos es una difícil categorización de sitios permitiendo que se bloqueen más o menos información de la necesaria y algunas diferencias idiomáticas que dificultan el filtrado [11].

Algunos proyectos de I+D que se han trabajado recientemente son POESIA y TEFILA. POESIA (Public Opensource Environment for a Safer Internet Access) es un sistema de código abierto para el filtrado de información inapropiada en ambientes escolares. TEFILA (Técnicas de Filtrado basadas en Ingeniería del Lenguaje, Aprendizaje automático y agentes) orienta al filtrado en el puesto de trabajo [12].

Para la identificación de contenido riesgoso, los sistemas de filtrado implementan métodos de clasificación de contenidos que permiten identificar patrones en la información evaluada. En la siguiente sección se presenta una revisión de los métodos de clasificación de contenidos más utilizados para el filtrado de páginas web.

III. Métodos de clasificación de contenidos

Los métodos de clasificación de contenidos permiten la categorización de las páginas Web mediante el aprendizaje realizado en un conjunto de documentos previamente preclasificados, llamado conjunto de entrenamiento. De este conjunto de entrenamiento los métodos de clasificación de contenidos aprenden las características de las clases para evaluar nuevos documentos como pertenecientes o no pertenecientes a las clases consideradas en el entrenamiento [13]. A continuación se presentan algunos métodos de clasificación de contenidos usados para el filtrado de páginas web.

A. Métodos bayesianos

Están basados en el teorema de Bayes, una de sus principales aplicaciones es determinar si un correo electrónico es *spam* o no. El filtro debe disponer de una base de datos para poder actuar. El funcionamiento es el siguiente, al recibir un nuevo correo, se realiza un análisis descomponiendo el texto en palabras y seleccionando las más relevantes, para que sean procesadas por el filtro bayesiano calculando la probabilidad de que el correo recibido sea inapropiado o no. Si la probabilidad supera un umbral establecido se considera como información inapropiada [14].

B. Redes neuronales

Son algoritmos de aprendizaje automático basados en el funcionamiento del cerebro humano; se encuentran conformadas por varias capas de unidades de procesamiento o neuronas interconectadas, en las cuales las unidades de entrada reciben los términos del documento, mientras que las unidades o neuronas de salida representan las categorías de interés y los pesos en las interconexiones de las neuronas expresan la mayor o menor fuerza de la conexión. Las redes neuronales son entrenadas mediante el ajuste de los pesos de las conexiones, para realizar la clasificación de los términos de un documento o página web determinada [8].

C. Árboles de decisión

Se pueden describir como un conjunto de condiciones organizadas en forma de estructura jerárquica, de tal modo que la decisión que se tomará como resultado, se puede determinar siguiendo las condiciones cumplidas desde la raíz del árbol hasta algunas de sus hojas. En los árboles de decisión, los nodos internos son etiquetados como atributos, las ramas salientes de cada nodo representan pruebas para los valores del atributo, y las hojas del árbol identifican a las categorías [7].

D. Método de Rocchio

Construye los patrones de las diferentes clases de contenido partiendo de una colección de entrenamiento que se clasifica manualmente con anterioridad. El método identifica vectores prototipo para cada una de las clases predefinidas, que contienen ejemplos positivos y negativos de cada categoría. Una vez se han identificado los vectores prototipo para cada una de las clases, se estima la similitud entre un nuevo documento y cada uno de los vectores prototipos, de esta manera se estima la pertenencia a una clase según la similitud del nuevo documento con los vectores[7].

E. Máquinas de soporte vectorial

Los SVM (Support Vector Machine o Máquinas de Soporte Vectorial) son unos clasificadores muy utilizados en la identificación de patrones en texto e imágenes. Una SVM construye un hiperplano de separación óptima, el cual separa un conjunto de muestras positivas de un conjunto de muestras negativas maximizando el margen de separación. Dado que las características pueden almacenarse en forma de vectores n-dimensionales, se utiliza un hiperplano [14].

IV. Sistemas Multiagente (SMA)

Los sistemas multiagente presentan grandes ventajas para el desarrollo de prototipos inteligentes para realizar actividades que requieren aprendizaje, por este motivo, en esta sección se presenta una introducción al tema. Un agente es una entidad física o abstracta que puede percibir su ambiente, evaluar estas percepciones, tomar decisiones y comunicarse con otros agentes para obtener información [15]. La comunicación es la forma de interacción entre agentes, esta puede darse mediante comunicación directa (con el paso explícito de mensajes) o comunicación indirecta (con el censado de otros agentes o el medio) [15]. Los sistemas multiagente son un conjunto de agentes organizados que interactúan en un ambiente común. Los agentes se clasifican de acuerdo a la función que cumplen y su dominio o sus servicios, los cuales pueden ser: búsqueda de información, filtrado de datos, monitorización de condiciones y alertas, entre otros [16]. Una de las áreas más estudiadas en los sistemas multiagente, es el desarrollo de metodologías que permitan diseñar sistemas exitosos. A continuación se presenta una revisión de las metodologías de análisis y diseño de sistemas multiagente y de las plataformas que soportan este tipo de desarrollos.

A. Metodologías de análisis y diseño de SMA

La programación orientada a agentes presenta diferencias respecto a la orientada a objetos, ya que los objetos modelados son elementos tangibles mientras que los agentes son abstracciones intangibles, cuyos elementos más importantes son sus metas y su adaptabilidad. Dadas las necesidades específicas de los sistemas multiagentes nace la Ingeniería de Software Orientada a Agentes (ISOA o AOSE por sus siglas en inglés –Agent-Oriented Software Engineering) [17] con el desarrollo de diversas metodologías que se presentan a continuación.

1) AUML

Extiende los conceptos y elementos de los modelos orientados a objetos para satisfacer las necesidades de los agentes, es decir, es una extensión del UML clásico para el

modelado de agentes [17], esto facilita la implementación de esta metodología, ya que parte del estándar de modelado orientado a objetos UML pero lo extiende en conceptos propios de los agentes.

2) Vowel Engineering (Ingeniería de Vocales)

El proceso de esta metodología se basa en la conjunción de cuatro unidades denominadas mediante vocales sobre las cuales se aplican diferentes técnicas de modelado, donde: con la vocal A se denominan los Agentes; con la vocal E se denomina el Entorno; con la vocal I se denominan las Interacciones, y con la vocal O se denomina la Organización. Lo que se pretende es el desarrollo de componentes individuales para cada unidad, permitiendo la reutilización de componentes [18].

3) BDI (Beliefs, Desires, Intentions)

Se encuentra inspirada en el modelo cognitivo del propio ser humano. Un agente recibe una serie de estímulos procedentes de su entorno a través de sus sensores, los cuales modifican el modelo del mundo que tiene el agente, es decir, sus creencias (beliefs) acerca del mundo. Además, el agente tiene que tomar decisiones para interactuar con el mundo, basadas en sus creencias y guiadas por sus deseos (Desires) u objetivos y para ello se vale de una serie de intenciones (Intentions) o acciones que el agente va realizando [18].

4) Gaia

En esta metodología la especificación de los requerimientos es independiente del proceso de análisis y diseño del sistema, además se dividen la fase de análisis de la fase de diseño, lo cual indica que se separan los conceptos que son entidades puramente abstractas, que no deben tener una imagen directa sobre la implementación; de los que son conceptos concretos, los cuales son entidades que tienen un reflejo directo en la implementación del sistema. Esta metodología se usa para aplicaciones grandes cuyos agentes son heterogéneos [18].

5) MaSe (Multiagent System Engineering)

Es una metodología basada en el ciclo de vida clásico del software, con un entorno propio de desarrollo, llamado agentTool para analizar, diseñar y construir sistemas multiagente heterogéneos. En esta metodología, los agentes no son considerados como entes autónomos, proactivos y sociales, sino como simples procesos que se comunican para conseguir el objetivo global del sistema [18].

6) Zeus

ZEUS consta de una herramienta y una metodología, de forma similar a agenTool y MaSE. Esta metodología propone un desarrollo en cuatro etapas: el análisis del dominio, el diseño de los agentes, la realización de los agentes y el soporte en tiempo de ejecución. De estas, la herramienta soporta las etapas de realización de los agentes y soporte en tiempo de ejecución. Esta metodología incluye conceptos propios de los agentes, como: planificación, ontologías, asignación de responsabilidades, relaciones sociales entre agentes [19].

7) MESSAGE/UML

MESSAGE (Methodology for Engineering Systems of Software Agents) parte de UML y lo extiende con conceptos de nivel de conocimiento y diagramas con notaciones sobre agentes. Una de las razones para considerar a UML como punto de partida, es que este es aceptado como estándar de modelado orientado a objetos y los paradigmas objetual y orientado a agentes son altamente compatibles. Además, UML se basa en un metamodelo lo que lo hace extensible [19].

8) Tropos

Es una metodología que parte de los conceptos propios de los agentes, tales como: metas, planes y capacidades para lograrlos, y de sus requerimientos, incluyendo a los actores y separando las funcionalidades del sistema. Después de tener definido lo que el sistema debe hacer, se identifican las capacidades necesarias para alcanzar los planes exigidos por las metas y luego agrupar dichas capacidades para generar los tipos de agentes necesarios para implementar el sistema. Tropos es independiente de la plataforma [17].

9) MAD-Smart

El enfoque de la metodología para el análisis y diseño de sistemas multiagente robóticos MAD-Smart está fundamentado en dos principios: es independiente de las técnicas de implementación y define un proceso metodológico ascendente. Esta metodología recopila elementos de otras metodologías, tales como: GAIA, Mas-CommonKADS y MaSe. Los ocho pasos que constituyen la metodología están orientados a definir la comunicación entre agentes, describir el hardware y software de los agentes individuales y distribuir tareas entre los agentes [15].

10) MAS-CommonKADS

Esta metodología es una extensión orientada a agentes de la metodología CommonKADS, cuyo enfoque principal es la construcción de sistemas expertos.

Plantea la integración del SMA con el modelo del ciclo de vida de software espiral dirigido por riesgos. Plantea la definición del sistema empleando 7 modelos diferentes: agente, tareas, experiencia, coordinación, comunicación, organización y diseño [18].

B. Plataformas de desarrollo de SMA

Hoy en día existen plataformas completas para desarrollar sistemas multiagente. Estas plataformas tratan de brindar a los programadores todos los elementos necesarios para liberarlos de la necesidad de diseñar librerías de comunicación o movilidad. Incluso existe una organización dedicada a desarrollar estándares y normas sobre los elementos e interfaz que deben implementar las plataformas de agentes inteligentes [17]. A continuación se presenta una revisión de las plataformas.

1) JADE

Es una plataforma que acoge los estándares de la FIPA (Foundation for Intelligent Physical Agents) que propone estándares abiertos para la interoperabilidad de los sistemas orientados a agentes. Está implementada en Java, tiene licenciamiento libre (LGPL o Licencia pública general blanda), es una librería que se instala sobre máquinas virtuales Java y forma una plataforma distribuida. Jade tiene una interfaz gráfica que permite monitorear el sistema, incorpora varios mecanismos de comunicación y modelado de información entre agentes [17].

2) AgentTool

Es el software que implementa los siete pasos del proceso MaSE con soporte para transformar modelos de análisis en modelos de diseño. Permite crear visualmente todos los diagramas del proceso de desarrollo, realizando cualquiera de los 7 pasos en el orden que se prefiera, pero respetando las dependencias entre ellos. Esta herramienta dispone de una base de conocimiento persistente, un verificador de conversaciones y generación automática de código [18].

3) ZEUS

ZEUS consta de una herramienta y una metodología, de forma similar a agenTool y MaSE. La herramienta se ha convertido en referencia de cómo debe ser una herramienta para el desarrollo de SMA, por la forma en que combinan los resultados de investigación en agentes (planificación, ontologías, asignación de responsabilidades, relaciones sociales entre agentes) en un sistema ya ejecutable [19].

4) Jack

Plataforma comercial que toma como base la metodología Tropos, cuyos constructores de implementación, es decir, sentencias en un lenguaje especial extendido de Java, se transforman directamente con los elementos de un modelo final de Tropos, de tal forma que un diseño de Tropos se puede traducir directamente a un programa para esta plataforma. Jack es una marca registrada de la empresa Agent-Software [17].

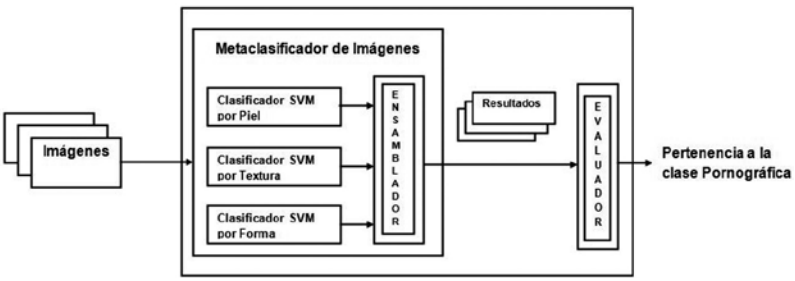
V. Caracterización del modelo de filtrado

Para el desarrollo de un sistema de filtrado de pornografía mediante la evaluación del contenido multimedial de las páginas web se necesitan filtros independientes para cada recurso hipertextual (texto, imágenes y estructura HTML). Una vez se tienen los filtros individuales se deben unir los resultados mediante un método de ensamble. A continuación se describen los filtros y el método de ensamble.

A. Filtro basado en las imágenes

Algunas investigaciones han utilizado el contenido de las etiquetas "img" de HTML extrayendo el título, la etiqueta "alt" y el nombre del archivo para realizar el filtrado. Otras investigaciones se basan en el contenido de la imagen y sus características tales como: el color, la textura o las características estadísticas que compartan las imágenes [6]. La solución implementada en este artículo evalúa el contenido de la imagen para realizar el filtrado. A continuación, en Fig. 1 se presenta la solución propuesta en [8] para el filtro de imágenes.

Figura 1. Estructura del filtro de imágenes, tomado de [8].



El filtrado basado en las imágenes recibe como entrada las imágenes de una página web y su salida es la evaluación indicando si el contenido es pornográfico o no. Internamente, el filtro está conformado por un metaclasificador y un evaluador [8].

El metaclasificador recibe las imágenes descargadas y realiza tres tipos de clasificaciones con máquinas de soporte vectorial previamente entrenadas que evalúan características de piel, características de textura y características de forma. Las características de la piel son extraídas identificando las regiones de piel de la imagen y organizándolas en un vector. Las características de textura se obtienen tomando la imagen y transformándola a niveles de grises para calcular la matriz de coocurrencia; luego, con estas características, se forma el vector con el cual se indexa la imagen para iniciar el proceso de clasificación. Las características de forma son los descriptores de forma (conjunto de números que tratan de describir un objeto) que permiten clasificar una imagen según la forma de su contenido. Internamente el metaclasificador tiene un módulo ensamblador que recibe los resultados de los tres clasificadores y es el encargado de determinar si una imagen corresponde o no a la clase pornográfica en base a la política de que una imagen es considerada como pornográfica siempre y cuando al menos uno de los clasificadores arroje como resultado que la imagen es de este tipo. El resultado del metaclasificador es entregado al módulo evaluador [8].

El evaluador se encarga de decidir cuándo una página web es pornográfica. La entrada del módulo está conformada por los resultados de la evaluación de cada una de las imágenes contenidas en la página; para determinar si una página tiene contenido pornográfico se implementa la siguiente política: si al menos 30% de las imágenes son pornográficas, entonces la página se considera pornográfica también. Para realizar dicha evaluación, este módulo espera el resultado de todas las imágenes presentes en la página [8].

B. Filtro basado en las etiquetas HTML

El filtro recibe como entrada la estructura HTML de una página web y retorna la clasificación de la página: porno o no porno; para realizar este proceso se tiene una estructura conformada por un módulo indexador de estructura HTML y un módulo clasificador. En Fig. 2 se ilustra el funcionamiento general del filtro propuesto en [7].

Figura 2. Estructura del filtro basado en etiquetas HTML, tomado de [7].

El módulo indexador recibe la estructura HTML para extraer características relevantes de la estructura HTML que permitan identificar contenido pornográfico. Las características son: cantidad de etiquetas de enlaces estáticos con la etiqueta <a href>, cantidad de etiquetas de imágenes con la etiqueta <img>, cantidad de

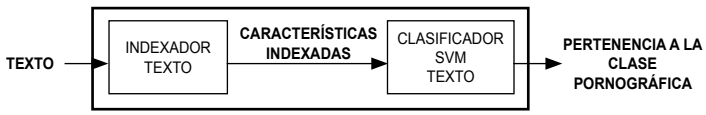
imágenes animadas con extensión ".gif", cantidad de documentos con extensión ".doc", ".txt" y ".rtf", cantidad de documentos con extensión ".pdf" y ".ps", cantidad de videos con extensión "wmv" y "avi", cantidad de objetos flash con extensión ".swf", cantidad de popups en javaScript con el código "(.open)", cantidad de tablas con la etiqueta <table>, cantidad de palabras obscenas en todas las etiquetas HTML ("sex", "xxx", "porn", "hardcore"), cantidad de etiquetas <script> en el código HTML y cantidad etiquetas <meta> en el código HTML. Las características son contabilizadas para encontrar su frecuencia de aparición y formar los vectores para la clasificación [7].

El módulo clasificador SVM asigna la categoría porno o no porno a la página web de entrada, evaluando las características extraídas; el entrenamiento del clasificador SVM es realizado por medio de máquinas de soporte vectorial que son entrenadas previamente al proceso planteado en la figura 2.

C. Filtro basado en el texto

Para la construcción de un filtro de texto se requiere inicialmente un preprocesamiento del contenido de la página web para extraer sólo el contenido textual; luego, éste debe ser transformado a una representación adecuada para el clasificador construido dependiendo del dominio de estudio. Para realizar este tipo de clasificación se debe asumir que el conjunto de palabras y su frecuencia de aparición describen la clase a la cual pertenece la página, lo cual no ocurre en todos los dominios [6]. En Fig. 3 se presenta una estructura para el filtro de texto.

Figura 3. Estructura del Filtro basado en texto, tomado de [6].



El módulo indexador recibe como entrada el texto de la página web y la indexa con pesos reales, los documentos son representados como vectores de características con pesos reales, que son asignados utilizando la función *tf*idf*, donde *tf* es la frecuencia de la característica en un documento e *idf* es la frecuencia inversa de la característica en el conjunto de documentos, esta plantea que entre más veces una característica ocurre en un documento es más representativa de su contenido y entre más veces ocurra en los documentos es menos discriminante. En la fórmula de la *idf* (1), D es el total de documentos y $df(C_i)$ es el número de documentos donde ocurre la característica C_i [6].

$$idf(C_i) = log\left(\frac{D}{df(C_i)}\right) \qquad (1)$$

El módulo clasificador SVM asigna la categoría porno o no porno a la página web de entrada, evaluando las características extraídas; el entrenamiento del clasificador SVM es realizado por medio de máquinas de soporte vectorial que son entrenadas previamente al proceso planteado en la figura 3.

D. Método de ensamble de los filtros

Los métodos de ensamble se basan en el principio "divide y vencerás", combinando un conjunto de soluciones [6].

Un ensamble de filtros *f* obtiene mejores resultados que la utilización de filtros individuales $\{h_1, h_2, h_3\}$ porque estadísticamente al promediar los votos de los filtros se reduce el riesgo de seleccionar un filtro erróneo [6].

En el ensamble se define la pertenencia o no a la clase pornográfica dependiendo de las predicciones de varios filtros que evalúen diferentes recursos hipertextuales. Para unir los diferentes resultados de los filtros se utiliza una estrategia de combinación adaptativa propuesta en [6] mediante pesos estáticos y dinámicos para los filtros.

Los pesos estáticos $\alpha = \{\alpha_1, \alpha_2, \ldots \alpha_k\}$, indican el nivel de confianza que se puede tener sobre la predicción de cada uno de los filtros de manera individual. Estos pesos se definen a priori de la utilización de la estrategia, como resultado de un proceso de aprendizaje, asignando un mayor peso a los filtros con mejor desempeño en un grupo de datos. En (2) se presenta una descripción de los pesos estáticos, donde p es la precisión y r la cobertura [6].

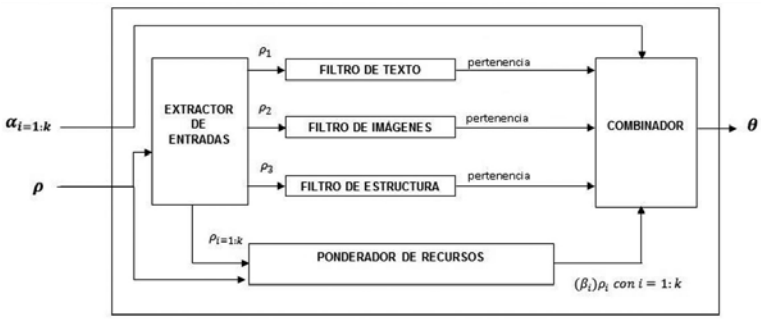
$$\alpha_i = \frac{2p_i r_i}{p_i + r_i} \tag{2}$$

Los pesos dinámicos $\beta = \{\beta_1(\rho_1), \beta_2(\rho_2), \ldots, \beta_k(\rho_k)\}$ indican la cantidad de información que tiene disponible cada filtro para realizar la clasificación, por lo cual se presentan en función de los recursos hipertextuales ρ_i. Estos pesos se calculan como la razón entre la cantidad de caracteres del recurso i $|\rho_i|$ y la cantidad total de caracteres de la página $|\rho|$ que se está filtrando [6]. Por ejemplo, el peso dinámico de un filtro de texto está dado por la razón entre la cantidad de caracteres de sólo texto y la cantidad total de caracteres de la página, como se presenta en (3).

$$\beta_i(\rho_i) = \frac{|\rho_i|}{|\rho|} \tag{3}$$

En Fig. 4, se presenta la arquitectura del método de ensamble. Se requiere como entrada la página web a clasificar ρ y los pesos estáticos α_i de los filtros; la salida θ está dada por un valor que indica la pertenencia o no pertenencia a la clase pornográfica [6].

Fig. 4. Arquitectura del Ensamble. Tomado de [6].



La arquitectura del ensamble se compone de un extractor de entradas, los diferentes filtros, un ponderador de recursos y un combinador. El extractor de entradas preprocesa la página web a clasificar y le entrega a cada filtro el recurso hipertextual ρ_i sobre el cual desarrolla la evaluación.

Los filtros reciben los recursos hipertextuales que van a evaluar y entregan su resultado al módulo combinador. Los filtros son construidos, entrenados y evaluados antes de llegar a la etapa de ensamble con cada una de las estructuras presentadas anteriormente.

El módulo ponderador de recursos tiene como entrada la página web ρ a clasificar y el resultado del extractor de entradas; con esta información se calculan los pesos dinámicos $\beta_i(\rho_i)$ de los filtros.

El módulo combinador recibe los resultados de los filtros, los pesos dinámicos calculados por el ponderador y los pesos estáticos que son calculados a priori. Por medio de (4), el módulo combinador ensambla los resultados de los filtros para dar un único resultado que indique si una página web es pornográfica; el resultado final es calculado como el promedio ponderado de las predicciones de los filtros con la suma de sus respectivos pesos estáticos y dinámicos [6].

$$\theta_1(\rho, porn, \alpha, \beta, \emptyset) = \frac{1}{k}\sum_{i=1}^{k}(\alpha_i + \beta_i(\rho_i))^* \emptyset_i(\rho_i, porn) \qquad (4)$$

VI. Desarrollo del SMA para el filtrado de pornografía

Para el diseño de un sistema multiagente que realice el filtrado de pornografía basado en el contenido multimedial de las páginas web se debe utilizar una metodología que permita el correcto diseño de los agentes. Para seleccionar una metodología de diseño de SMA se establecieron algunos criterios como: la metodología debe permitir modelar

el conocimiento de los agentes, la metodología debe tener independencia de la técnica de implementación y de una plataforma, la metodología debe ser robusta y finalmente la cantidad de documentación disponible sobre la metodología debe ser extensa y accesible. Teniendo en cuenta los criterios establecidos se selecciona como metodología de diseño MAS-CommonKADS, la cual tiene un ciclo de vida con diferentes fases que plantean el desarrollo de modelos. A continuación se presentan los resultados obtenidos en las fases de conceptualización, análisis y diseño de la metodología.

La fase de conceptualización tiene como objetivo alcanzar una mejor comprensión del sistema multiagente a desarrollar, identificando los objetivos del sistema, los actores que interactúan con el sistema y se realiza una descripción informal de los casos de uso. Como resultado de esta fase se identifican los actores activos y pasivos del sistema con sus respectivos objetivos. Los actores activos son los humanos que interactúan con el sistema: un administrador y un usuario. El administrador del sistema puede iniciar el sistema, realizar consultas, recibir resultados, registrar información para el filtrado de las páginas en archivo y finalizar el sistema. El usuario del sistema puede utilizar el filtro en línea. En esta fase se identificaron también los agentes necesarios para desarrollar todas las actividades del sistema.

La fase de análisis tiene como objetivo profundizar en la especificación del sistema; logrando delimitarlo y descomponerlo, con el fin de clarificar las características propias de los agentes a desarrollar.

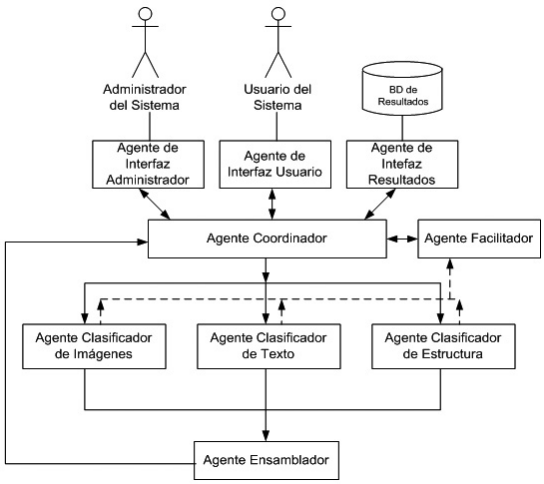
La fase de diseño tiene como objetivo definir las arquitecturas de los agentes y de la red, con el fin de transformar la información contenida en las fases anteriores de tal modo que se pueda plasmar en un lenguaje de programación.

Como resultado de la aplicación de las fases propuestas por MAS-CommonKADS se presenta en la figura 5 la arquitectura final del sistema multiagente donde se pueden ver por medio de las flechas las interacciones entre los actores del sistema y los agentes (Ver figura 5).

En la arquitectura de la figura 5 se presentan 2 agentes interfaz que interactúan con los usuarios para que estos puedan realizar peticiones al sistema, estos son: el agente *Administrador* y el agente *Usuario,* los cuales pueden enviar a evaluación una URL de una página web a un agente *Coordinador*.

El agente *Coordinador* se comunica con el agente *Resultados* para verificar en la base de datos de resultados si la página ha sido evaluada anteriormente, en tal caso, el agente *Resultados* le envía la evaluación de la clasificación anterior al agente *Coordinador* para que envíe la respuesta al Agente Interfaz que realizó la petición. Si no existen resultados anteriores de la evaluación de la página web, el agente *Coordinador* consulta a un agente *Facilitador* cuáles agentes están disponibles para evaluar recursos hipertextuales.

Fig. 5. Arquitectura del sistema multiagente



La función del agente *Facilitador* es estar en contacto con agentes *Clasificadores* que realicen la labor de evaluar recursos hipertextuales según los filtros presentados en la sección anterior. De esta manera, el sistema cuenta con un agente *Clasificador de Texto* que internamente tiene la estructura presentada en Fig. 3, donde recibe como entrada el texto de la página web y entrega como resultado la pertenencia o no pertenencia a la clase pornográfica según el texto. El agente *Clasificador de Estructura* tiene internamente la estructura presentada en Fig. 2, donde recibe como entrada la estructura HTML de la página y entrega como resultado la pertenencia o no pertenencia a la clase pornográfica según la estructura. El agente *Clasificador de Imágenes* tiene internamente la estructura presentada en Fig. 1, donde recibe como entrada las imágenes presentes en la página web y entrega como resultado la pertenencia o no pertenencia a la clase pornográfica según el contenido de las imágenes.

Los resultados de los agentes Clasificadores son entregados a un agente *Ensamblador*, que realiza la combinación de las evaluaciones de los clasificadores según la estructura interna presentada en Fig. 4. La evaluación final realizada por el *Ensamblador* es entregada al agente *Coordinador*, quien le envía el resultado obtenido al agente *Resultados* para almacenarlo en la base de datos y posteriormente entregarle los resultados al agente Interfaz que realizó la petición. El agente Interfaz *Administrador* o *Usuario* debe tomar la decisión de desplegar o bloquear la página web solicitada según los resultados obtenidos por el sistema.

VII. Evaluación del prototipo

El diseño realizado por medio de la metodología MAS-CommonKADS es evaluado por medio de un prototipo desarrollado en la plataforma JADE. Este prototipo es evaluado en un diseño experimental descrito a continuación. Los datos experimentales son tomados del Open Directory Project (ODP) <http://www.dmoz.com>. Para evaluar el prototipo del sistema multiagente construido se selecciona una muestra aleatoria de 5.000 páginas web, siguiendo la distribución de los datos existentes en Internet que se muestra en la tabla 1 publicada en el sitio web <http://www.internet-filterreview.com>.

El objetivo de la evaluación es la verificación de la siguiente hipótesis: ¿Un filtro web que evalué los diferentes contenidos multimediales de las páginas web puede tener mejor desempeño que un filtro web que evalué solo el texto? Para lograr la verificación de esta hipótesis, se realizan pruebas de la eficacia de un filtro web que evalúe sólo texto y otro que evalúe diferentes contenidos multimediales, en condiciones similares de tal manera que sea posible realizar una comparación del desempeño de los filtros en función de la eficacia.

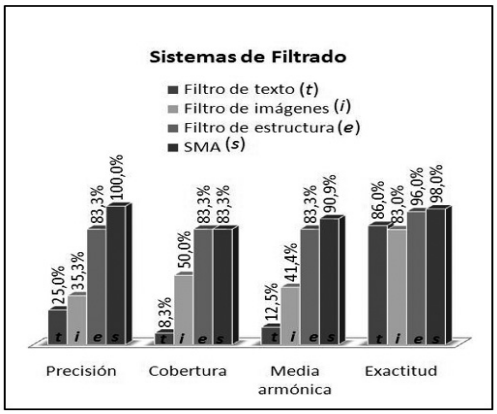
Tabla 1. Distribución de tipos de información en la web

Tipo de página web	Porcentaje de existencia en la web
Porno	12%
Artes	15%
Negocios	13%
Noticias	12%
Sociedad	11%
Niños	5%
Salud	3%
Compras	6%
Computadores	6%
Ciencia	5%
Recreación	6%
Deportes	6%

Las métricas de eficacia utilizadas para la evaluación del prototipo son las siguientes: Precisión (p), que es la razón entre la cantidad de páginas pornográficas clasificadas correctamente y el total de páginas clasificadas como pornográficas. Cobertura (r),

que es la razón entre la cantidad de páginas pornográficas clasificadas correctamente y el total de páginas pornográficas de la muestra. Media armónica (*f1*), que es la media armónica *f1* entre la precisión y la cobertura. Exactitud (*e*), que es la razón entre el número de documentos clasificados correctamente y el total de documentos. Los resultados de la evaluación de los filtros individuales y del sistema multiagente de filtrado son presentados en Fig. 6.

Figura 6. Métricas de eficacia



De los resultados de la Fig. 6, se puede concluir que en términos de eficacia un sistema de filtrado que evalúe los diferentes contenidos multimediales de las páginas web tiene un mejor desempeño que un filtro web que evalúan sólo el texto en todas las métricas evaluadas, respondiendo con esto la hipótesis formulada en la evaluación. Adicionalmente, el sistema desarrollado en este trabajo tiene un mejor desempeño que la evaluación individual de cualquiera de los filtros, excepto en la cobertura del filtro de estructura, donde obtienen el mismo nivel de desempeño.

VIII. Conclusiones

En este trabajo se presenta un sistema multiagente para el filtrado de pornografía mediante la evaluación del contenido multimedial de las páginas web. Para el diseño del sistema multiagente se utiliza la metodología de diseño orientada a agentes MAS-CommonKADS y la plataforma JADE. Para la evaluación del sistema multiagente se construye un prototipo, el cual fue probado con un conjunto de 5.000 páginas web. De la evaluación se obtiene que en término de las métricas precisión, cobertura, media armónica entre las métricas anteriores y exactitud el sistema de filtrado desarrollado en este trabajo tiene mejor desempeño que los convencionales clasificadores de texto utilizados en los sistemas de filtrado.

Diseñar y desarrollar el prototipo por medio de un sistema multiagente permite aprovechar las características de estos tipos de sistemas. Una de estas características es la autonomía de cada uno de los agentes al ejecutar sus tareas de forma simultánea para clasificar la página web. La metodología de diseño de SMA seleccionada (MAS-CommonKADS) es muy completa, ya que comprende aspectos y características propias de los agentes, que permiten clarificar la definición de los mismos y facilitar su posterior implementación.

La plataforma de desarrollo seleccionada (JADE) es de gran utilidad ya que es independiente del sistema operativo y no se encuentra ligada a alguna metodología de análisis y diseño. Además, JADE contiene herramientas propias orientadas a agentes que facilitan la implementación del modelo diseñado.

Se propone como trabajos futuros la adición de nuevos filtros al sistema multiagente como el análisis de enlaces o links en las páginas web y análisis de videos en las páginas web. También, se propone como trabajo futuro la implementación de las ontologías definidas en el modelo de MAS-CommonKADS en los diferentes agentes del sistema.

Agradecimientos

Los autores reconocen las contribuciones de los siguientes proyectos: proyecto de pregrado "WEPCIB - Clasificador de páginas web pornográficas basado en el contenido de las imágenes" de William Armando Ceballos Imbacuan y Luis Eduardo Salazar Taborda; proyecto de pregrado "Clasificador de páginas web basado en la estructura html - CLASES" de Sandra Patricia Tamayo Giraldo y proyecto de maestría "Método de filtrado de páginas web basado en el ensamble de clasificadores hipertextuales" de Ana Isabel Oviedo Carrascal.

Referencias

[1] María Ángeles Hernández Prados y Patricia López Vicent, "Contenido nocivo en la red. ¿Qué Hacer? ", Universidad de Murcia, 2008. Disponible en: http://www.congresointernetenelaula.es/virtual/archivosexperiencias/200806041531262008_COM_Contenido_nocivo.doc

[2] Ana Luisa Rotta Soares, "La protección de los niños y niñas en internet – Los sistemas de filtrado", I Congreso internacional sobre ética en los contenidos de los medios de comunicación en internet, Octubre de 2001. Disponible en: http://www.ugr.es/~sevimeco/biblioteca/etica/ana%20rotta.doc

[3] RED USI, "Preguntas Frecuentes - RED USI", Uruguay Sociedad de la Información, 2009. Disponible en: http://www.usi.org.uy/es/preguntas-frecuentes/index.html#faqs-5

[4] Francisco Serradilla García, "Agentes de información", Escuela Universitaria en Ingeniería de Sistemas y Automática, 2003. Disponible en: http://www.sia.eui.upm.es/grupos/Ainfo1.pdf

[5] Rubén Fuentes Fernández, Jorge Gómez Sanz y Juan Pavón, "Recuperación de información mediante adaptación automatizada de ontologías en Sistemas Multi-Agente", Universidad Complutense de Madrid, España, 2009. Disponible en: http://grasia.fdi.ucm.es/main/myfiles/IR_Adaptable_con_MAS.pdf

[6] Ana Isabel Oviedo Carrascal, Andrés Marín Lopera y Oscar Ortega Lobo, "Método de filtrado de páginas Web basado en el ensamble de clasificadores hipertextuales", Conferencia Latinoamericana En Informática CLEI 2007, San José, Costa Rica. Memorias de la Conferencia Latinoamericana en Informática 2007.

[7] Ana Isabel Oviedo y Sandra Patricia Tamayo Giraldo, "Clasificador de páginas web basado en la estructura html - "CLASES"", Universidad de Antioquia, 2009.

[8] Ana Isabel Oviedo Carrascal, William Armando Ceballos Imbacuan y Luis Eduardo Salazar Taborda, ""WEPCIB" Clasificador de páginas web pornográficas basado en el contenido de las imágenes", Colombia Revista Colombiana De Computación ISSN: 1657-2831 Ed: Universidad Autónoma de Bucaramanga v.10 fasc.N/A p.26 - 44 ,2009

[9] OPTENET, "Herramientas de filtrado - Seguridad en Internet", 2009. Disponible en: http://www.zonagratuita.com/servicios/seguridad/art-esp8.html

[10] Luz M. Quiroga, "Sistemas de filtrado: Un puente tecnológico entre oferta y demanda de información en línea al servicio de la toma de decisiones", Universidad de Hawai, 2009. Disponible en: http://www.cepal.org/dds/noticias/paginas/2/14632/ppt_LMQuiroga_Hawaii.ppt

[11] Marjorie Heins, Cristina Cho y Ariel Feldman, "Internet filters a public policy report", Brennan Center for Justice, 2006. Disponible en: http://www.freespeechonline.org/webdocs/filters2.pdf

[12] José María Gómez Hidalgo, Enrique Puertas Sánz, Francisco Carrero García y Manuel de Buenaga Rodríguez, "Categorización de texto sensible al coste para el filtrado de contenidos inapropiados en Internet", Departamento de inteligencia artificial, Universidad Europea de Madrid, Septiembre de 2003. Disponible en: http://www.sepln.org/revistaSEPLN/revista/31/31-Pag13.pdf

[13] Enrique V. Carrera, María del Cisne García y Fausto Pasmay, "Un algoritmo simple y eficiente para la clasificación automática de páginas Web", Noviembre de 2008. Disponible en: http://profesores.usfq.edu.ec/vinicioc/papers/andescon08.pdf

[14] Álvaro Gascón y Marín de la Puente y Miguel María Rodríguez Aparicio, "Clasificación jerárquica de contenidos Web", Universidad Carlos III de Madrid, 2009. Disponible en: http://www.it.uc3m.es/jvillena/irc/practicas/06-07/30.pdf

[15] Jovani Alberto Jiménez B., Marcela Vallejo Valencia y John Fredy Ochoa Gómez, "Metodología para el análisis y diseño de sistemas multi- agente robóticos: MAD-SMART", Universidad Nacional de Colombia y Universidad de Antioquia, 2007. Disponible en: http://www.colombiaaprende.edu.co/html/docentes/1596/articles-155887_archivo.unknown

[16] Clara Inés Peña de Carrillo, "Sistemas Multiagente para el tratamiento de la información en el Web: Agentes de interfaz, Agentes de información, Agentes de aprendizaje, Agentes intermediarios", I Congreso Internacional de Ingeniería de Sistemas - EISI 30 años, Universidad Industrial de Santander, Bucaramanga, Colombia, Noviembre 2000. Disponible en: http://eia.udg.es/~clarenes/docs/congreso-uis-2000-agentes-cip.pdf

[17] César A. Cabrera E. y Fredy A. Ramírez R., "Agentes inteligentes, Web Semántica Y PLN en Univirtual", Universidad Tecnológica de Pereira, Julio de 2006. Disponible en: http://www.cesarcabrera.info/proyectoGrado/

[18] Francisco José Gallego Durán, Faraón Llorens Largo y Ramón Rizo Aldeguer, "Breve análisis de algunas metodologías de diseño de SMA", Universidad de Alicante, Noviembre de 2004. Disponible en: http://www.dccia.ua.es/dccia/inf/asignaturas/AI/docs/SMA.pdf

[19] Jorge J. Gómez Sanz, "Metodologías para el desarrollo de sistemas Multi-Agente", Departamento de sistemas informáticos y programación, Universidad Complutense. Disponible en: http://cabrillo.lsi.uned.es:8080/aepia/Uploads/18/38.pdf

[20] Carlos Ángel Iglesias Fernández, "Definición de una metodología para el desarrollo de Sistemas Multiagente", 1998. Disponible en: http://www.gsi.dit.upm.es/tesis/pdf/tesiscif.pdf