

# FILTRADO DE CONTENIDOS WEB EN ESPAÑOL DENTRO DEL PROYECTO POESIA

Enrique Puertas  
*Universidad Europea de Madrid*  
[epuertas@uem.es](mailto:epuertas@uem.es)

Francisco Carrero  
*Universidad Europea de Madrid*  
[fcarrero@uem.es](mailto:fcarrero@uem.es)

José María Gómez Hidalgo  
*Universidad Europea de Madrid*  
[jmgomez@uem.es](mailto:jmgomez@uem.es)

Manuel de Buenaga  
*Universidad Europea de Madrid*  
[buenga@uem.es](mailto:buenga@uem.es)

## RESUMEN

Este artículo presenta el sistema para filtrado de contenidos de Internet POESIA, un proyecto de código abierto que utiliza técnicas de Procesamiento del Lenguaje y análisis de imágenes para filtrar contenidos inapropiados en Internet. Aunque el sistema incluye filtros para tratar 3 idiomas (Inglés, Italiano y Español), en el artículo nos centramos en el módulo de filtrado para el castellano, así como en los métodos utilizados para recopilar las colecciones que se han usado para entrenar y probar el sistema.

## PALABRAS CLAVES

Filtrado de contenidos, Procesamiento del Lenguaje, Recuperación de Información, Código Abierto

## 1. INTRODUCCIÓN

POESIA (Public Open source Environment for Safer Internet Access) es un proyecto financiado por la Unión Europea dentro del programa para una Internet más segura y que cuenta con la participación de diversas instituciones Europeas<sup>1</sup>. Su objetivo es proveer a centros de educación un servicio de filtrado de contenidos inapropiados (páginas con pornografía o lenguaje obsceno, por ejemplo) evitando que menores de edad puedan acceder a este tipo de material. POESIA es un sistema de código abierto, lo que quiere decir que cualquier persona puede usar, instalar o modificar el sistema libremente. En este artículo vamos a presentar una visión global de POESIA, con una breve descripción de sus distintos filtros para luego centrarnos en el filtro textual para el Español que hemos desarrollado en la Universidad Europea de Madrid. Comentaremos el método utilizado para recopilar la gran cantidad de páginas web que ha sido necesaria para entrenar y evaluar el sistema para terminar presentando y comentando los resultados de los experimentos realizados para evaluar el filtro.

## 2. EL SISTEMA POESIA

El enfoque usado en el sistema POESIA consiste en una serie de filtros que reciben información de elementos relativos a una página web y retornan una serie de evidencias de si la página web debe ser filtrada o no. El sistema cuenta con un mecanismo de decisión encargado de combinar las respuestas de los distintos filtros y decidir, de acuerdo a los resultados que éstos devuelven, la decisión de filtrar o no.

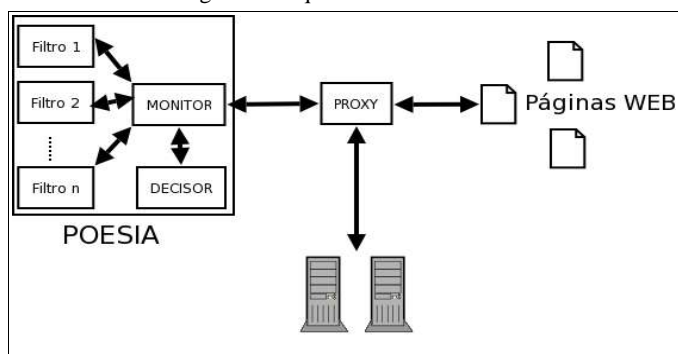
---

1 Puede verse una lista completa de las entidades participantes en <http://www.poesia-filter.org>

El elemento central del sistema es el Monitor, que es el módulo que se encarga de interactuar con los distintos elementos del sistema. Cuando se realiza una petición para ver una página web por parte de un cliente (navegador), éstos realizan una petición al servidor proxy que se va a encargar de recuperar los contenidos solicitados. Antes de devolver esos contenidos al navegador para que los muestre, se realiza una petición al monitor de POESIA, pasándole los contenidos obtenidos de la web, para ver si la página debe ser aceptada y mostrada o si debe ser rechazada.

Cuando el Monitor recibe la petición, envía los distintos elementos a los filtros para que estos den su opinión de si deben ser filtrados o no. Los resultados de los filtros se envían al mecanismo decisor que los combina (los resultados de los filtros se ponderan según el tipo de contenido) y da un valor que indica si los contenidos web solicitados deben ser aceptados o rechazados. En la figura 1 se puede ver la arquitectura del sistema.

Figura 1. Arquitectura de POESIA.



Entre los filtros de POESIA podemos encontrar un filtro que analiza imágenes basado en algoritmos de detección de piel y formas, un filtro reconocedor de idioma basado en n-gramas[CAVNAR94] y filtros que analizan el texto para los idiomas Italiano, Inglés y Español. Los enfoques seguidos en cada idioma son distintos con la idea de que cada filtro incorpore las técnicas y los recursos más adecuados y efectivos según las peculiaridades gramaticales de cada idioma. En las próximas líneas nos vamos a centrar en explicar el funcionamiento de los filtros en Español.

### 3. FILTRO PARA EL ESPAÑOL

Al igual que para el resto de idiomas, para el español tenemos un filtro ligero basado en aprendizaje estadístico que devuelve una respuesta en un tiempo corto, ya que el tiempo de respuesta es un factor crítico. El problema de filtrar páginas web puede abordarse como un problema de categorización automática de Textos [SEBASTIANI02]. Para decidir si una página en español se debe filtrar, construimos un clasificador usando Máquinas de Soporte Vectorial (Support Vector Machines en la literatura). Como atributos para la clasificación usamos las palabras del documento, que son obtenidas mediante un proceso de tokenización básica usando separadores habituales del español (espacios, puntos, comas, fines de línea, etc.). Una vez que hemos obtenido las palabras, aplicamos una lista de parada que elimina aquellas palabras que no aportan información y las sometemos también a un proceso de extracción de raíces que deja los términos en su forma morfológica básica (por ejemplo, “trabajo”, “trabajar” → “trabaj”). Finalmente representamos los documentos mediante el Modelo de Espacio Vectorial [SALTON89], con un vector de pesos binarios, con 1 si aparece la palabra y 0 si no lo hace. Para reducir el espacio de representación seleccionamos sólo aquellos atributos con mayor Ganancia de Información. En los experimentos presentamos resultados seleccionando conjuntos de atributos con el 0,5% y el 1% del total de atributos.

Usando esa representación entrenamos un clasificador basado en Máquinas de Soporte Vectorial, un algoritmo de clasificación que selecciona de cada clase un pequeño número de instancias límite llamadas

Vectores Soporte y construye una función lineal discriminante que las separa lo más posible. La implementación del algoritmo matemático que hay detrás de las máquinas de Soporte Vectorial queda fuera de los límites de este artículo. Para más información consultar [PLATT98].

Una vez que tenemos construido el clasificador, cada nueva instancia se clasifica aplicando el modelo, por ejemplo:

$$-1.99 * \text{sex} - 0.35 * \text{porn} + \dots > 0 \Rightarrow \text{Página segura}$$

Puesto que el filtro debe devolver un valor en forma de probabilidad (dato que utilizará el Mecanismo Decisor), y que las máquinas de Soporte Vectorial no nos proporcionan ese tipo de resultados, ajustamos el modelo mediante regresión lineal para poder obtener porcentajes.

## 4. CONSTRUCCIÓN DE LA COLECCIÓN

Para poder entrenar y evaluar el sistema, ha sido necesario construir una colección de documentos web con contenidos pornográficos y otra con contenidos seguros, todos en Español. La labor de recuperar y clasificar manualmente miles de documentos web resultaba inabordable, por lo que se optó por construir un robot que recuperase ese contenido de forma automática de la web.

Partiendo del Proyecto de directorio público DMOZ<sup>2</sup>, un directorio de páginas web clasificadas manualmente en categorías temáticas por cientos de voluntarios, recuperamos todas las páginas que colgaban de las categorías regionales en Español haciéndonos así con más de 35.000 documentos (>1Gb de información HTML) para poder entrenar y evaluar nuestro filtro. La colección de documentos recuperados fue postprocesada para eliminar páginas vacías (errores 404, frames, etc.) y aquellas que a pesar de estar categorizadas dentro de una categoría en español, su contenido tenía mayor cantidad de términos en otro idioma (se usó un reconocedor de idioma para realizar esta labor de filtrado).

## 5. EXPERIMENTOS Y RESULTADOS

Los experimentos realizados para evaluar nuestro filtro de páginas en Español se realizaron usando la colección anterior mediante validación cruzada. El clasificador se construyó usando el paquete Java de aprendizaje automático WEKA [WITTEN99]. En el experimento 1, cuyos datos se pueden ver en la Tabla 1, se seleccionó usó el 0,5% de los términos con mayor Ganancia de Información. La tabla 2 muestra los resultados obtenidos para el segundo experimento, con el 1% de los atributos con mayor Ganancia.

Tabla 1. Experimento 1 con 0.5% de atributos.

Tasa TP	Tasa FP	Precisión	Recall	F-Measure	Clase
0,936	0,001	0,987	0,909	0,964	positiva
0,999	0,064	0,980	0,997	0,992	negativa

Tabla 2. Experimento 2 con 1% de atributos.

Tasa TP	Tasa FP	Precisión	Recall	F-Measure	Clase
0,909	0,003	0,993	0,936	0,946	positiva
0,997	0,091	0,986	0,999	0,989	negativa

<sup>2</sup> [Http://www.dmoz.org](http://www.dmoz.org)

En las tablas podemos ver las tasas de Verdaderos Positivos (TP) y de Falsos Positivos (FP) así como los porcentajes de Recall y Precisión y de la medida F que combina ambos, obtenidos en los distintos experimentos. Puede observarse que al aumentar el número de términos utilizados en el segundo experimento, los resultados mejoraron en todas las medidas. El porcentaje de instancias clasificadas correctamente en el experimento 1 fue del 97,8003%, mientras que en el obtenido en el experimento 2 fue de 98,0784%. La Tabla 3 nos presenta a modo ilustrativo los 20 tokens con mayos Ganancia de Información que se obtuvieron para ambos experimentos:

Tabla 3. Lista de Experimento 2 con 1% de atributos.

Término	Término
-1.9983 sex	-2.152 voyeur
-0.3543 porn	-0.8738 transex
1.9585 amateur	-0.5636 xxx
-1.5137 desnud	-0.4037 lesb
1.2176 chic	0.7856 cul
-0.2054 gay	-0.3948 grat
-0.0036 fot	-1.5882 sad
-1.756 jovencit	0 asiatic
-1.2976 madur	-0.6195 webcam
-1.7792 tet	0.5225 corrid

## REFERENCIAS

[CAVNAR94]

W.B. Cavnar, W.B and Trenkle, J.M. *N-grambased text categorization*. In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval, pages 161-175, 1994.

[PLATT98]

Platt J. (1998). *Fast training of support vector machines using sequential minimal optimization*. In Schlkopf B., Burges C., and Smola A. editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press.

[SALTON 89]

Salton, G. 1989. *Automating text processing: the transformation, analysis and retrieval of information by computer*. Addison- Wesley.

[SEBASTIANI02]

Sebastiani, F. 2002. *Machine learning in automated text categorization*. ACM Computing Surveys, 34(1):1-47.

[WITTEN99]

Witten, I.H. y E. Frank. 1999. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann.

