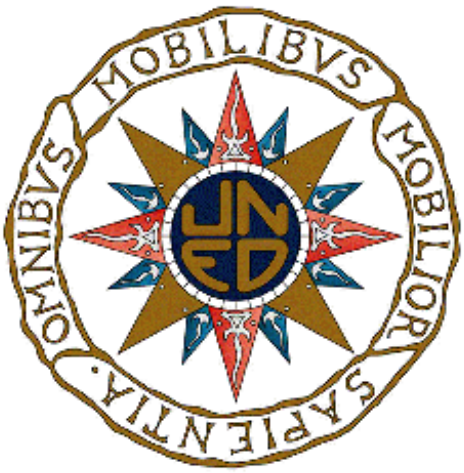
UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA
UNED



Clasificación de páginas Web en dominios específicos

MEMORIA DE PROYECTO PRESENTADA COMO PARTE DE LOS REQUISITOS PARA LA OBTENCIÓN DEL TÍTULO DE MASTER EN LENGUAJES Y SISTEMAS INFORMÁTICOS

D. Francisco Manuel Rangel Pardo

Dirigido por Dr. D. Anselmo Peñas Padilla

A mis padres, que me lo dieron todo.
A mi hermanita, que siempre será mi hermanita.
A mi familia, que siempre me apoya.
A mis animales, que son mi alegría.
A mi mujer, que es mi mundo.
A mi enanita, que es mi vida.

Agradecimientos

A mis padres, por haberme dado la vida y por haberme aguantado durante tantos años de estudiante... y los que les quedan por aguantarme.

A mi hermana, por aguantar mis broncas, nervios y malos genios en épocas de examen.

A mi abuela, mi tía, y familia, por su cariño.

A mis amigos, por no olvidarse de mí, aunque pasen años sin vernos.

A mis caballos, por estar ahí en los peores momentos, en especial a mi Terry, por haberme dado tantas alegrías y celebraciones.

A mis perros, que Dios los tenga en su gloria, por haberme divertido tanto con sus recercas, y a mi Samba, que tanto me cabrea, y a la que tanto quiero

Al Chaval de la Peca, por entretener los últimos momentos de la elaboración de este trabajo.

A mi director de proyecto, Anselmo Peñas, por su inestimable ayuda y orientación en este proyecto, así como al resto de profesores del master por haberlo hecho posible.

Y por último a la UNED por dar la posibilidad de cumplir el sueño de estudiar con la realidad de trabajar.

A todos los nombrados, gracias, así como a todos los que se me haya podido olvidar y que saben que merecen de tales.

Y no, no me olvido, a mi mujer, por quererme, mimarme y aguantarme pese a mis cabreos, mis dolores de cabeza y mis temporadas encerrado elaborando esta tesis. A ella se lo debo todo.

Gracias

Resumen

El presente trabajo intenta dar solución al problema de la clasificación en diferentes categorías de páginas Web en dominios específicos, y como caso de estudio se propone el del teatro en habla hispana.

Partiendo de un conjunto de Urls sobre teatro previamente clasificadas, se construye una representación formal de las mismas que será utilizada para el entrenamiento y creación de un conjunto de clasificadores mediante un proceso de aprendizaje inductivo, para posteriormente ser utilizados en la predicción de las categorías de nuevas páginas Web.

La investigación se centra en obtener una representación novedosa y competitiva respecto al estado del arte actual, para lo cuál se proponen dos soluciones, una solución general, basada en la meta-información de la cabecera de las páginas y que contiene la intención del autor por comunicar información acerca de su sitio Web, y en la meta-información de los enlaces, que contiene la intención del autor por dar una estructura clara y lógica a su sitio de acuerdo a algún criterio, y una solución específica basada en la representación de las características propias de las páginas de tipo *Blog*, que permita su clasificación de manera precisa independientemente de su contenido y el idioma del mismo.

Se realiza un estudio comparativo de las clasificaciones propuestas con las existentes en el estado del arte y se describen los resultados obtenidos, bastante favorables en el caso de la representación general, y especialmente favorables en el caso de la representación específica de los *Blogs*, y se describen las líneas de investigación futuras para mejorar y trasladar los resultados a otros dominios de la Web.

Las líneas futuras de investigación se describen para mejorar y extender los resultados a otros dominios de la Web.

Abstract

The present work tries to solve the problem of classification in different categories from Web pages in specific dominions, and as case of study sets out the one of the theater in Hispanic speech.

Starting off of a set of Urls on theater previously classified, a formal representation of the same ones is constructed that will be used for the training and creation of a set of sort keys by means of a process of inductive learning, later to be used in the prediction of the categories of new Web pages.

The investigation is centered in obtaining a novel and competitive representation with respect to the present state-of-the-art, for which two solutions, a general solution, cradle in the meta-information of the head of the pages and that contains the intention of the author to communicate information about its Web site, and in the meta-information of the connections, that it contains the intention of the author to give a clear and logical structure to his site according to some criterion, and a specific solution based on the representation of the own characteristics of the pages of Blog type, that independently allows to its classification of precise way of its content and the language of it itself.

A comparative study of the propose classifications is made with the existing ones in the state-of-the-art and the results obtained, quite favorable in the case of the general representation, and specially favorable in the case of the specific representation of the Blogs.

The future lines of investigation are described to improve and to transfer the results to other dominions of the Web.

ÍNDICE

Tabla de acrónimos

ARFF: Fichero formato Weka (del inglés *Attribute Relation File Format*)
BoW: Bolsa de palabras (del inglés *Bag-Of-Words*)
GNU: Proyecto de sistema libre (del inglés *GNU is not unix*)
HTML: Lenguaje de marcas de hipertexto (del inglés *hypertext marckup language*)
IR: Recuperación de información (del inglés *Information Retrieval*)
IVKM: Máquina Virtual de Java para .Net
SVM: Máquinas de Vectores Soporte (del inglés *Support Vector Machines*)
SWL: Lista de palabras vacías (del inglés *Stop Word List*)
TC: Categorización de textos (del inglés *Text Categorization*)
TF-IDF: Frecuencia del término por frecuencia inversa del documento (del inglés *Term Frecuency-Inverse Document Frecuency*)
XP: eXtreme Programming o programación extrema

Tabla de figuras

CAPITULO 1 INTRODUCCIÓN

.

Internet, y en concreto la Web, es hoy por hoy la fuente de datos más consultada para obtener información sobre cualquier tema. La gran cantidad de información que posee y la rapidez con la que evoluciona, hace posible que cualquier información compartida en una parte del mundo automáticamente sea visible en la otra punta del mismo.

Pero toda esta información no sirve de nada si no está estructurada o clasificada de algún modo, permitiendo una navegación ordenada o una búsqueda en la misma

Por ello es de vital importancia la clasificación automática en el entorno Web, de manera que se dé orden a la información contenida y sirva tanto para facilitar la búsqueda y la navegación de los usuarios, como la presentación de resultados útiles y relevantes al usuario que los precisa.

1.1 NECESIDADES DE LA CLASIFICACIÓN WEB

La World Wide Web, o la Web como se la conoce comúnmente, es el mayor repositorio de información existente. Dicha información se almacena y distribuye a lo largo de todo el mundo en forma de documentos de hipertexto, que no son más que una colección de palabras que permiten además de definir un contenido, definir una estructura de visualización del mismo a través de lo que se denominan tags, o marcas de hipertexto, y además, su mayor potencia se encuentra en los hipervínculos, que son enlaces desde dicho documento hacia otros documentos web, formando un entresijo de relaciones que es lo que conforma la Web.

La gran cantidad de información que proporciona la Web no es comparable con ningún medio conocido hasta la actualidad. Pero toda esta información debe ser procesada de algún modo para que sea de utilidad al usuario que la requiera. Para ello desde sus comienzos han sufrido un tremendo desarrollo los motores de búsqueda, capaces de presentar información relevante a los usuarios a partir de una consulta facilitada por los mismos.

En los comienzos de la Web aparecieron una serie de portales proveedores de información clasificada en diferentes categorías, de manera que los usuarios ávidos de información recurrían a ellos y buscaban bajo la jerarquía de categorías aquélla información deseada.

Esta tarea de clasificación era realizada por grandes grupos de recursos humanos. Los usuarios que deseaban tener presencia en la web mandaban su solicitud de inclusión en estos directorios enviando su web junto con las categorías dónde querían que apareciese. Un grupo de editores humanos se encargaba de velar por la relevancia de los documentos y su adecuación a las categorías solicitadas.

En la actualidad el tamaño de la Web es varias veces el de sus comienzos y la gran cantidad de contenidos que contiene, y el rápido cambio que sufren, no pueden ser

catalogados de manera manual como en sus comienzos, por lo que las tareas de clasificación automática se hacen extremadamente necesarias.

La clasificación es un problema que ha venido unido al desarrollo cultural del ser humano, quien ha intentado poner orden en la naturaleza clasificándola en base a unos criterios que él mismo ha definido.

Con la aparición y evolución de la informática se han orientado muchas líneas de investigación hacia la búsqueda de soluciones a la automatización de las tareas de clasificación, desarrollando complejos algoritmos que llevasen a cabo las mismas. La clasificación automática ha sido una de las ramas más estudiada en la minería de datos.

Con el nacimiento de Internet y su rápida evolución, la clasificación ha encontrado un nuevo reto; clasificar los contenidos de la Web para un mejor suministro de la información, sirviendo de base para estructurar grandes directorios informativos anteriormente etiquetados a mano, así como para la construcción de los índices de los buscadores más avanzados, permitiendo una búsqueda directa de contenidos que se adecuen a un tema dado.

Pensar en la actualidad en una clasificación manual de contenidos es prácticamente un imposible; Sitios Web de noticias de diferentes ámbitos, portales de contenidos sobre diferentes temas... son sólo algunos de los muchos ejemplos que requieren de un trabajo exhaustivo de clasificación.

Además, con el acercamiento tecnológico cada día más a la sociedad en los países desarrollados, la necesidad de proteger de ciertos grupos sociales de un uso pernicioso de la web se ha hecho precisa.

Así como en el cine o en los video-juegos, los contenidos de Internet deben ser calificados como aptos o no para grupos sociales especialmente protegidos como los niños, de manera que estos no tengan acceso a contenidos sexuales, violentos o que generen algún tipo de perjuicio.

Todo este problema de protección se reduce a la clasificación de los contenidos dentro de ciertas categorías predefinidas en base a unos criterios prefijados, de manera que se proteja de aquellas categorías que no sean aptas para su edad o condición.

Es por todo ello que el problema de la clasificación automática de páginas web es un reto interesante para la investigación y además su resolución, o al menos mejora, es una necesidad en la sociedad actual.

1.2 DIFICULTADES EN LA CLASIFICACIÓN WEB

Pero la problemática de la clasificación Web va más allá del problema de la clasificación o de su resolución automática mediante computadoras; la clasificación Web, por las características de la misma, tiene una serie de dificultades añadidas y unos retos nuevos que hay que abordar desde nuevas perspectivas.

El contenido de la Web son las páginas web, definidas mediante el lenguaje de marcado de hipertexto html, que permite definir mediante una serie de etiquetas su contenido y el modo de visualización del mismo. Además permite definir enlaces a otras páginas web, conformando con todo ello el entramado de páginas que componen la Web.

La mezcla de contenido y estructura de visualización llega en la actualidad a tal punto en el que el uso de elementos visuales como imágenes, mapas de imágenes, vídeos o contenidos multimediales que enriquecen su presentación, hacen de las tareas de clasificación una labor compleja, pues los algoritmos automáticos no son capaces, en la mayoría de los casos, de indagar en el interior de estos elementos visuales para determinar su contenido.

Por otra parte, lo que se plantea como un método para dar estructura a las páginas, el conjunto de marcas de etiquetado, tiene como consecuencia que la web sea un medio poco estructurado por naturaleza. La flexibilidad del lenguaje para definir su contenido es tal que cada autor puede utilizarlo de la manera que más le convenga, creando de este modo contenidos que pueden ser únicos en su estructura, dotando con ello de gran variabilidad a la web.

Debido a su carácter público, la Web está compuesta por documentos escritos por todo aquél que desee hacerlo, con lo que la variedad de estilo introducido por los diferentes autores de las páginas puede introducir variaciones lingüísticas como sinonimias o polisemias, además de todo tipo de errores ortográficos, sintácticos y semánticos.

A todo lo anterior, y debido a su carácter global, se une la gran diversidad de idiomas que aparecen en la Web. Siendo el inglés uno de los idiomas más utilizados, tal y como muestran estudios de [O'Neill 1998-2002] el número de los mismos es elevado, por lo que una clasificación clásica basada en su contenido deberá limitarse bien a un único idioma, bien utilizar técnicas de recuperación multilingüe.

Derivado de todo lo anterior está el problema de la alta dimensionalidad de las representaciones utilizadas en la clasificación de textos clásica, como la bolsa de palabras. Las variaciones de estilo y vocabulario, el multilingüismo, los errores, las etiquetas, las palabras vacías... todo ello resulta en representaciones con muy alta dimensionalidad difíciles de tratar con los algoritmos actuales.

Es por todo ello que la minería de datos tradicional no se puede aplicar directamente a la minería de contenido web y es necesario adaptar sus métodos y algoritmos de manera que se saque el mejor partido posible a este tremendo repositorio de datos.

Aquí nace la minería web, que no es más [Hernández Orallo] que el uso de técnicas de minería de datos para descubrir y extraer información automáticamente desde la Web.

De entre las diversas ramas de la minería web (contenido, estructura y uso), nos interesa principalmente la de contenido y estructura, que es la que servirá para extraer las características interesantes para aprender un modelo de clasificación.

El proceso de minería web, de manera similar a todo proceso de minería de datos, requiere resolver las siguientes subtareas:

- Localizar los documentos en la web, tarea que principalmente consiste en localizar los documentos interesantes para las tareas de minería a llevar a cabo, su extracción desde la web y su almacenamiento en dispositivos locales en lo que se denominan índices de documentos web, muy utilizados en los motores de búsqueda, y que será sobre los que se realice el resto de tareas de minería. Esta etapa previa equivaldría a la etapa de reunir los datos desde diferentes datawarehouses para construir la vista minable, ya que la mayoría de algoritmos de minería trabajan con los datos de manera local (aunque en la actualidad existen tendencias a distribuir la minería de datos [Rangel 2007])

- Selección y pre-procesado de la información, que es la etapa primordial en la que se centra el estado del arte de la investigación [Forman][Guyon][Zheng] pues de ella depende el buen resultado del proceso. En esta etapa se obtendrá la representación formal de los documentos en un conjunto de características que permita tratarlos de manera uniforme y aprender con ellos un modelo. Por las técnicas generalmente aplicadas, así como por la gran cantidad de ruido existente (páginas que no existen, contenidos que no se corresponden con lo que dicen ser...) así como por la alta dimensionalidad que suelen presentar, la necesidad de un pre-preocesado de la información se hace especialmente necesaria en este tipo de tareas de minería.

- Generalización. Esta etapa es común a toda minería de datos y es la que se encarga, aplicando un algoritmo o método determinado, de obtener un modelo capaz de generalizar a partir de los datos presentados como ejemplos a la entrada.

- Análisis, validación e interpretación, dónde se comprobará la bondad de los modelos aprendidos y se interpretará los resultados obtenidos.

Es el segundo punto el que mayor diferencia tiene con respecto a la representación de otros problemas de minería de datos, y el que más dificultades incorpora y por lo tanto en el que se centra principalmente la investigación en clasificación Web.

1.3 OBJETIVOS

Los objetivos de la investigación se centran en intentar dar solución al problema de la categorización de páginas web en dominios específicos, en concreto de páginas web de teatro en habla hispana como caso de estudio, para lo cuál se debe conseguir una aproximación **competitiva** y **novedosa** a la representación de las páginas web que permita un aprendizaje supervisado lo suficientemente bueno como para conseguir una predicción o nueva clasificación de una calidad aceptable.

Vistas las tareas a resolver por un proceso de minería web de contenido como el de la clasificación que nos ocupa, los objetivos concretos de la investigación se centrarán en los siguientes puntos:

- Crear y determinar la validez de un repositorio de datos para las tareas de entrenamiento y validación de modelos de clasificación Web

- Determinar un marco de evaluación correcto de los modelos creados para asegurar su adecuación al problema de la clasificación Web

- Comprobar la importancia del preprocesado de las características, así como de la aplicación de técnicas computacionales a la resolución de problemas lingüísticos, para la consecución de modelos más manejables y representativos, y por lo tanto robustos

- Comprobar cómo se puede mejorar la clasificación clásica basada en contenido (BoW) añadiendo características contextuales y utilizando información de las Urls

- Mejorar la clasificación web existente mediante la representación de la meta-información aportada por el creador de la página en su objetivo por transmitir información acerca de la misma

- Obtener una representación específica para el tipo de páginas Blogs, independiente del idioma, que se apoye en sus características invariables y consiga una evaluación más precisa y robusta que las existentes en la investigación actual

- Determinar un marco de investigación futura que permita trasladar los resultados aquí obtenidos a otros dominios específicos de la Web, así como permitir sentar las bases para una clasificación general de algunos tipos de páginas, como en este caso concreto las de tipo Blog

1.4 ESTRUCTURA DEL TRABAJO

La memoria se estructura en cinco grandes bloques, a saber, un primer bloque con la introducción al proyecto, dónde se discute la importancia de la clasificación Web, los retos que plantea y los objetivos que se persiguen con la investigación. Este primer bloque marca la necesidad de la investigación, su motivación, y los objetivos planteados para dar respuesta al problema de la clasificación Web.

En un segundo bloque se realiza una introducción preliminar al problema de la clasificación Web. En primer lugar se describe en líneas generales los conceptos previos necesarios para entender la investigación, su desarrollo y evaluación, introduciendo al lector en el marco conceptual de la investigación

A continuación se describe el estado actual de la investigación en clasificación web, los diferentes proyectos llevados a cabo, las líneas de investigación seguidas y los resultados obtenidos, dando al lector un marco temporal sobre la investigación y que sirve de base comparativa con el trabajo realizado.

Tras ello se describe el conjunto de alternativas posibles a seguir para la consecución de los objetivos, analizando las implicaciones y tomando las decisiones apropiadas para su realización. En este punto no se marca la guía de investigación, sino un referente acerca de por dónde empezar en vistas a perseguir los objetivos planteados.

Por último se describe la metodología general utilizada en el desarrollo de la investigación, desde la metodología de desarrollo para crear las herramientas necesarias para la investigación, el entorno de desarrollo utilizado, hasta las herramientas de minería de datos utilizadas y los métodos de evaluación de modelos. Es en general una introducción a las metodologías y tecnologías utilizadas para el desarrollo de la investigación.

En el tercer bloque se realiza una serie de experimentos orientados a la consecución de los objetivos. Los experimentos se determinan sobre la base de pequeñas hipótesis que con su aceptación o rechazo irán conformando un conjunto de conclusiones que se acercarán a la resolución de los objetivos.

Los experimentos son concretos, orientados a obtener, a partir del experimento anterior, un acercamiento progresivo hacia los objetivos, por lo que cada objetivo se verá cumplido tras la realización y conclusión de varios de estos experimentos.

El cuarto bloque capitula el conjunto de conclusiones obtenidas a partir de los diferentes experimentos y analiza su implicación en la consecución de los objetivos de la investigación. En este bloque se comentan las líneas de investigación futuras a seguir a partir de los resultados y conclusiones obtenidas y concluye el estudio de la investigación con la aceptación de los objetivos planteados.

El quinto bloque presenta el conjunto de referencias bibliográficas utilizadas para la realización de la investigación y la confección de la memoria.

Por último, en el sexto bloque se adjunta una serie de anexos con el análisis (I) y diseño (II) del sistema informático que ha servido de utilidad para la realización de los experimentos de la investigación, ejemplos de pantallas de la aplicación (III), los resultados completos de los experimentos, con matrices de contingencia e intervalos de confianza del error (IV) y por último una serie de direcciones con los recursos básicos de la investigación como la aplicación y los ficheros necesarios (V)

CAPITULO 2 PRELIMINARES

2.1 CONCEPTOS PREVIOS

La clasificación web no es más que una variedad de la clasificación clásica de documentos, que al final no es más que una de las tareas más frecuentes de la minería de datos, pero con unas peculiaridades añadidas como se vio en la introducción.

Como toda tarea de clasificación automática, se requiere un proceso previo de aprendizaje que parte de una representación formal de los elementos a clasificar, construye un modelo y tras una validación que determine su idoneidad para el problema, queda listo para aplicarse a la clasificación de nuevos ejemplos.

Es por ello que lo primero que hay que considerar a la hora de crear un clasificador es la formalización de la representación de los documentos a tratar, pero antes que nada se introducirá en qué consisten y en qué se diferencian las tareas de clasificación y categorización.

2.1.1 Clasificación vs. Categorización

Tanto la clasificación como la categorización se pueden englobar dentro de las tareas de minería de datos denominadas predictivas.

Los problemas predictivos son aquellos en los que hay que predecir uno o más valores para un ejemplo dado. El aprendizaje se realiza de manera supervisada con ejemplos que van acompañados de una evidencia o valor de clase, clases, valor numérico u orden entre ellos.

Dependiendo de la correspondencia entre los ejemplos y los valores de salida se puede diferenciar entre diferentes tareas predictivas. Es aquí dónde se diferencian las tareas de clasificación de las tareas de categorización.

En la clasificación los ejemplos se presentan como un conjunto de pares de elementos de dos conjuntos, los ejemplos por un lado, y el valor de clase por el otro.

$$\delta = \{< e, s >: e \in E, s \in S\}$$

Donde E es el conjunto de ejemplos dados y S es el conjunto {c1, c2, ..., cn} de clases nominales posibles.

Para cada valor de ejemplo tenemos un único valor de clase, con lo cual al par <e,s> generalmente se le denomina ejemplo etiquetado, y con ello a delta se le denomina conjunto de datos etiquetados.

El objetivo es aprender una función clasificadora que represente la correspondencia única entre cada valor de E y su correspondiente valor de S.

$$\lambda : E -> S$$

La clasificación, también denominada discriminación en estadística, consiste en asignar una y sólo una clase a cada ejemplo.

Se denominan clasificadores binarios a los que discriminan entre dos posibles clases, o entre la pertenencia o no a una determinada clase.

La categorización por su parte consiste en aprender una correspondencia entre cada ejemplo y cada posible categoría

$$\lambda : E -> S$$

Dónde se pueden asignar varias categorías a un mismo ejemplo.

Como se puede apreciar, la diferencia entre clasificación y categorización es que ésta última consiste en asignar una entrada a tantas categorías como sea preciso, a diferencia de la primera que consiste en asignar una y sólo una clase al ejemplo dado.

Se puede construir fácilmente un clasificador a partir de un categorizador simplemente devolviendo la clase con mayor probabilidad, así como realizar un categorizador de N categorías mediante N clasificadores binarios que determinen si el elemento pertenece o no a dicha clase.

2.1.2 Representación/Caracterización formal de los documentos: el modelo vectorial

La primera acción a realizar tras la obtención de los documentos web es la de representar de manera adecuada el contenido de los documentos que formarán parte del entrenamiento. Los principales objetivos de esta representación son por un lado mantener de manera fiel la información que aporta el contenido del documento, y por otra ser adecuada a las especificaciones del algoritmo de aprendizaje a utilizar.

En el caso de las páginas web la información contenida está presente en diversos elementos, de los cuáles los dos más importantes son su contenido, es decir, el texto que finalmente vemos en un navegador, y su estructura de enlaces, que por su parte puede aportar gran cantidad de información como las páginas hacia las que apunta o desde las que es a apuntada cada documento. Respecto al contenido ciertas etiquetas pueden ser susceptibles de aportar información, como el título del documento, que en muchos de los casos hace de resumen del mismo, la información de visualización, que puede ensalzar algunos contenidos a modo de título, y un largo etcétera.

Sea cual sea la opción elegida para extraer la información se debe cumplir simultáneamente ambos objetivos nombrados anteriormente, fidelidad y adecuación.

Elegir una representación formal de la información contenida en los documentos implicará definir un espacio matemático, de manera que se abstraigan las características generales del mismo, y sobre el cuál poder operar y utilizar aproximaciones de aprendizaje automático.

Una de estas representaciones formales más utilizadas para caracterizar dominios textuales en los campos de la IR (*Information Retrieval*, Recuperación de Información) y la TC (*Text Categorization,* Categorización de textos) es la del modelo vectorial.

La representación de documentos en formato vectorial parte de la premisa de que el significado de un documento puede derivarse de las características o rasgos presentes en el mismo.

En las tareas de TC se suelen representar los documentos como vectores dentro de un espacio euclídeo, por lo que mediante la medición de la distancia entre dos vectores se puede determinar la similitud entre los dos documentos a los cuales representan dichos vectores. La medida de la distancia en un espacio euclídeo se realiza mediante el producto vectorial de los mismos, tarea que se puede desarrollar con un coste computacional bastante reducido.

Para tareas de TC e IR se asume una premisa, que aunque es muy fuerte y en la mayoría de los casos irreal, permite una reducción considerable del coste computacional. Esta es la del principio de independencia por el que se considera que las cadenas aparecidas en un mismo texto no tienen relación entre sí. Es una suposición incorrecta, pero en la mayoría de los casos no empeora los resultados obtenidos, y además permite la representación en forma de vector de los documentos, con lo cuál, como se introdujo en el párrafo anterior, permite medir su similitud como la distancia calculada mediante su producto vectorial, operación ésta que se pueden implementar de manera muy eficiente en cuanto a coste computacional se refiere, con lo que obtenemos lo comentado anteriormente.

Esta representación como vector de palabras se conoce en la bibliografía de IR como representación Bag-of-Words (BoW) En ella, cada palabra del documento, que pertenezca a un diccionario o corpus prefijado, corresponde a un atributo o dimensión del espacio. Un documento, por tanto, vendrá representado por los atributos o dimensiones que contiene. Como se puede comprobar, este modelo no tiene en consideración el orden de las palabras que contiene, por lo que al cumplirse la premisa indicada anteriormente permite el cálculo de la similitud como la medida de la distancia mediante producto escalar en el espacio euclídeo de representación.

Este es el tipo de representación clásica utilizada en los comienzos de la clasificación Web, y aunque ha quedado demostrado en diversos artículos que su precisión es bastante limitada, por los problemas que envuelven al contenido web, suele ser utilizada como línea base para comparar los resultados de mejoras propuestas a la clasificación Web.

Pero además de su representación en cuanto a espacio de características, se debe elegir entre una posible asignación de pesos a cada una de dichas características, con lo que habrá que discernir entre diferentes medidas o funciones de ponderación.

Función de ponderación

El conjunto de palabras que conforman el documento, y que pertenecen a su vez a un vocabulario dado o corpus, representan el conjunto de características o

dimensiones del documento en su representación formal, es decir, en el espacio vectorial que lo representa. Ahora bien, sus valores se pueden asignar de diversas maneras, lo que compone su función de ponderación.

Básicamente las funciones de ponderación pueden ser locales o globales. Una función de ponderación local es aquella que asigna valores a las diferentes características de manera local al documento en cuestión. En cambio, una función de ponderación global asignará el valor a las características teniéndose en cuenta algún tipo de medida respecto al conjunto de documentos global.

Dentro de las funciones de ponderación local tenemos como más frecuentes la función binaria, que consiste en asignar un valor booleano a la característica indicando si está o no presente en el documento en cuestión, la función de frecuencia que indica la frecuencia de aparición del término en el documento, y la función de frecuencia normalizada, que divide la frecuencia de aparición anterior entre el número de palabras del documento, de manera que la representación de diferentes documentos con este tipo de ponderación es más homogénea y no da mayor peso a los documentos con mayor número de palabras.

Dentro de las funciones de ponderación global aparece como más utilizada la función de frecuencia del término por la frecuencia inversa del documento (TF-IDF), la cual mide la frecuencia normalizada de aparición de la palabra en el documento teniendo en cuenta el número total de documentos y el número de documentos en los que la palabra aparece.

Existen multitud de funciones de ponderación, siendo las anteriores las más utilizadas en el tipo de problemas que nos ocupa.

Selección/creación del corpus

Como se describió anteriormente, los documentos a categorizar se representarán de manera formal definiendo un espacio vectorial, el cuál no es más que el conjunto de palabras que aparecen en un determinado documento y que serán las que definan su dimensionalidad y su espacio de características. Ahora bien, también se nombró que el conjunto de palabras o características que formarán parte de la definición del documento serán aquellas que a su vez pertenezcan a un vocabulario dado. Esto es lo que se conoce como corpus, o conjunto de palabras de un ámbito dado utilizado para tareas como esta.

El corpus puede venir determinado muchas de las veces a partir de un conjunto de textos que definen el ámbito sobre el cuál trabajar, o bien puede ser necesario crearlo. Para esto último se utilizará un conjunto de textos conocidos referentes al ámbito de trabajo, se extraerán sus palabras seleccionándolas mediante algún tipo de criterio y se generará el conjunto de características general o corpus a utilizar en la tarea de formalización de los documentos.

Selección de las palabras

Para la creación de un corpus se seleccionará del conjunto de textos de que se disponga aquellas palabras que mejor definan el ámbito de trabajo. Para realizar esta tarea existen diferentes aproximaciones, como la de extraer todas las palabras

contenidas y eliminar aquellas que tengan una frecuencia relativa de aparición extremadamente baja, ya que no serán discriminatorias, extremadamente alta, ya que al aparecer demasiado tampoco serán discriminatorias, las que aparezcan en muchos documentos de diferentes categorías, ya que por el mismo motivo no serán discriminatorias de las categorías, o cualquier otro tipo de proceso más elaborado.

Todo esto compone la primera etapa de preprocesado de los datos en el proceso de minería. A esta etapa se le pueden incluir procesos automáticos basados en algoritmos generales que utilizan medidas como la entropía, la ganancia de información, el análisis de componentes principales[SBC2], etcétera, para reducir por un lado la dimensionalidad (estamos tratando con problemas de dimensionalidad muy alta que pueden provocar importantes caídas en el rendimiento computacional de los clasificadores), y por otro lado aumentando el poder discriminatorio de las características seleccionadas.

A parte de lo anterior, que resulta en sí en un proceso semiautomático guiado en gran medida por la prueba y el error, se suelen aplicar dos procesos o etapas automáticas para aumentar el poder discriminatorio de las palabras utilizadas en el corpus, así como reducir la dimensionalidad eliminando aquellas que no se consideren importantes para su posterior tratamiento. Ambos procesos son los de eliminación de las palabras de fin o palabras vacías (*stop words*) y el del tratamiento o análisis léxico/sintáctico de las palabras (proceso de lemmatization, dissambiguation, stemmers...)

Palabras de fin/vacías (stop words)

Las palabras de fin/vacías constituyen el conjunto de palabras más comunes en un lenguaje y que no aportan información semántica al texto, como puedan ser las preposiciones, artículos, conjunciones, adverbios, adjetivos de uso frecuente, algunos verbos, sobre todo copulativos, etcétera, palabras que por sí solas no tienen capacidad discriminante.

Cabe hacer la puntualización de que no todas las palabras muy frecuentes en un lenguaje deben ser catalogadas como vacías ya que en ciertos ámbitos podrían ser palabras muy discriminantes a la hora de categorizar textos. Por otro lado, palabras que no se consideren vacías en un ámbito concreto sí que lo son, puesto que no aportan información ninguna. Así pues, el listado de palabras vacías se ha de construir con la supervisión adecuada para no eliminar de ellas palabras que puedan tener cierto poder discriminante.

Lematización, Stemming...

El conjunto de palabras en un ámbito dado suele ser muy elevado. A este problema se añade el de todas aquellas palabras que aunque diferentes tienen una raíz común y que por tanto pueden tener un significado léxico equivalente. También habría que tener en cuenta aquellas palabras que teniendo una raíz común tienen un significado diferente y que por tanto no se deben catalogar como una misma palabras. Y por último aquellas palabras que aún siendo diferentes, tienen un significado común, como es el caso de los sinónimos. El análisis e identificación de estos casos es complejo y requiere de ciertos conocimientos y recursos lingüísticos. Algunos de los procesos automáticos que se realizan en este punto son la lematización y el stemming (truncado)

El proceso de lematización consiste en asignar a cada palabra su correspondiente lema. De esta manera múltiples palabras con un lema común se agruparían bajo el mismo, reduciendo de esta manera no sólo la dimensionalidad sino también la dispersión, y mejorando, como múltiples estudios [Arregi] han puesto de manifiesto, los resultados de la clasificación. Este proceso requiere de analizadores morfológicos y de recursos lingüísticos complejos, por lo que es una tarea compleja en sí misma. Existen herramientas para ello como por ejemplo FreeLing.

El proceso de stemming consiste en el truncado de las palabras para reducir el número de rasgos o características del vocabulario. Para dicho truncado se suelen eliminar los prefijos y sufijos, plurales, femeninos... de manera que se realice una lematización de manera más sencilla y relajada, y sin necesidad de recursos lingüísticos extra. Existen algoritmos de truncado en diferentes lenguas, como el famoso algoritmo de [Porter 1980]

2.1.3 Aprendizaje de modelos inductivos

El último paso consiste en el entrenamiento del clasificador mediante aprendizaje inductivo en base a los ejemplos previamente clasificados dados para ello. Se está hablando de aprendizaje supervisado puesto que es sobre una base previamente clasificada sobre la que se entrena al clasificador, mediante supervisión de su comportamiento, a diferencia de métodos no supervisados como la construcción de mapas de Kohonen o similar, que agrupan los ejemplos por alguna medida de similitud entre ellos, sin tener en cuenta el valor de su etiqueta.

Se tiene el corpus, se tiene el conjunto de documentos y la categoría a la que pertenecen. El proceso consistirá en obtener la representación vectorial dado el corpus de los diferentes documentos y entrenar al clasificador con esta representación vectorial y su correspondiente clasificación.

Algunos de los clasificadores más utilizados en la práctica, tanto por sus buenos resultados, como por su idoneidad para trabajar con conjuntos de datos de alta dimensionalidad son las SVM, los modelos bayesianos y en especial su algoritmo Naïve Bayes y los árboles de decisión, principalmente la implementación del C4.

Naïve Bayes

El modelo Naïve Bayes es uno de los modelos más simples de clasificación con redes bayesianas, donde la estructura de la red es fija y sólo necesitamos aprender los parámetros o probabilidades. Se fundamenta en la suposición de que todos los atributos son independientes conocido el valor de la clase, suposición bastante fuerte y poco realista, pero que no afecta a su rendimiento de manera que es uno de los métodos más competitivos, y de los más usados en tareas reales como la identificación de correo basura.

Naïve Bayes a su vez permite tratar problemas con alta dimensionalidad, por lo que su idoneidad para el problema de clasificación web queda latente.

Árboles de decisión C4.5

Los árboles de decisión tienen un carácter voraz en su aproximación al aprendizaje por lo que se comportan especialmente bien con grandes volúmenes de datos de alta dimensionalidad.

El algoritmo C4.5 (y su versión propia de Weka, J4.8) es un método "divide y vencerás" basado en partición derivada de la ganancia (GainRatio) Realiza prepoda basada en cardinalidad, donde se eliminan los nodos con cardinalidad inferior a una constante dada, y postpoda mediante un criterio más sofisticado, lo que tendrá como consecuencia que cada nodo puede corresponderse con más de una clase, eligiéndose en todo momento la que mayor probabilidad tenga de ser la correcta.

Las principales características de este tipo de modelos son las ya citadas de tratar con grandes volúmenes de datos, así como de datos de alta dimensionalidad, y además ser tolerantes al ruido y a atributos no significativos, como es el caso del problema de la clasificación de textos, con lo cual son interesantes como acercamiento al problema que nos ocupa.

SVM

Las SVM han aparecido recientemente y se han aplicado típicamente a problemas de IR, TC, etcétera, demostrando, al margen de su solidez teórica, sus buenos resultados empíricos [Joachims 2002]

Las SVM pertenecen a la familia de los clasificadores lineales ya que inducen separadores lineales o hiperplanos en espacios de características de muy alta dimensionalidad.

Son por todo ello una opción interesante a investigar e implementar en un sistema de clasificación de textos.

2.2 SITUACIÓN DE LA INVESTIGACIÓN ACTUAL

La situación actual de la investigación muestra que se han explorado gran cantidad de alternativas en la extracción de características de la web para su posterior tratamiento, aprendizaje y modelización.

Las investigaciones han sido muchas y muy variadas, se ha investigado en diferentes representaciones formales de las páginas como el modelo clásico de bolsa de palabras (BoW), que se basa en obtener características a partir del contenido de la página, hasta modelos que explotan la estructura de enlaces y meta datos de la web, y la relación entre páginas, representándolas a las mismas como una serie de características referentes no sólo a su contenido.

2.2.1 Aproximaciones basadas en contenido textual (BoW)

Quizás el primer modelo de representación fue el conocido como Bag Of Words, o bolsa de palabras, heredado del área clásica de clasificación de documentos.

El método de bolsa de palabras [Hernández Orallo] consiste en definir cada palabra del documento que pertenezca a un diccionario prefijado o corpus de los documentos que constituyen la base de entrenamiento, como un atributo o dimensión del espacio de características.

Los valores de dichos atributos se pueden asignar mediante diferentes métodos. El más simple de ellos es el booleano, que indicará la presencia o ausencia de dicha característica (palabra) en el documento que se está analizando. Otras funciones de ponderación, como son llamadas, incluyen métodos más elaborados, que bien pueden ser locales, obtenidas a partir de características del propio documento, o globales, obtenidas a partir de características del documento frente al conjunto de documentos. En ambos casos la mayoría de métodos consisten en realizar un conteo del número de apariciones de la palabra y obtener la frecuencia mediante algún método local, como dividir por el número total de palabras del documento, o algún método global, como ponderar la frecuencia local por la frecuencia de aparición global.

En cualquier caso el orden de aparición de las palabras no se tiene en cuenta, simplemente su aparición. Esto es una suposición un poco fuerte [Hernández Orallo] pero que permite la comparación entre la representación de dos documentos mediante el producto escalar de los mismos, operación ésta bastante eficiente para ser realizada de manera computacional, lo que permite un tratamiento automático sin excesivo coste.

Si no es el primer método utilizado sí es al menos el más clásico en las tareas de clasificación de textos. Uno de los principales problemas es que por la utilización de las palabras como características genera representaciones de muy alta dimensionalidad, lo que provoca que no todos los algoritmos puedan trabajar con ella. Existen múltiples estudios sobre diferentes algoritmos y su eficiencia con este tipo de representaciones, y como se puede extraer de las conclusiones de algunos de ellos, existen algoritmos como las máquinas de vectores soporte (SVM) que muestran una clara superioridad[Joachims 2002]

Pero además del problema de la dimensionalidad el resultado obtenido por este método en clasificación web no obtiene muy buenos resultados por adolecer de un gran problema, que es la gran cantidad de ruido que se introduce en la estructura web y en la variedad de autores que la escriben.

Cada página web suele estar escrita por un autor, quién tiene su propio estilo tanto para describir el contenido, con su propio lenguaje y estilo, como para definir su estructura y apariencia. El html, a diferencia de otras representaciones textuales, mezcla la definición del contenido con la del estilo de visualización del mismo, mezclando por tanto la información textual con imágenes, scripts, elementos multimedia..., introduciendo gran cantidad de información no interesante a priori en una clasificación BoW, y considerándose como ruido en este tipo de modelos, ruido que si no es bien tratado por el algoritmo de clasificación puede provocar una merma importante en el mismo.

Pero quizás también es por todo lo anterior por lo que esta representación BoW suele servir de base (*baseline*) para la comparación de nuevas técnicas de caracterización de los documentos para el aprendizaje automático y la mayoría de trabajos se apoyan en ella para demostrar sus resultados comparativos.

2.2.2 Aproximaciones basadas en resúmenes

Una mejora interesante a la clasificación basada en contenido es aquella que utiliza un resumen del mismo para realizar el aprendizaje y la clasificación, de manera que no sólo se reduce la dimensionalidad del problema sino que se acota el mismo sobre un vocabulario más concreto acerca del contenido de la Web.

En estudios como [Shen 2004] se muestra cómo mejora significativamente la clasificación de páginas basadas en resúmenes hechos por humanos.

Como se ha indicado anteriormente la clasificación Web tiene el inconveniente añadido a la clasificación de documentos de la gran cantidad de ruido que se introduce por ser escritas por muchos autores diferentes que utilizan su propio estilo y vocabulario.

La aproximación a la clasificación de dichas páginas sobre un resumen de las mismas realizado por un equipo humano se asemeja mucho más a la clasificación de documentos textuales, ya que por un lado el estilo y vocabulario seguirá una plantilla común más o menos homogénea, y por otro lado, el resumen contendrá expresamente la información más concreta e interesante sobre el contenido de la Web.

Las mejoras obtenidas con dichos experimentos motivan a sus autores a implementar un sistema de resumen automático de textos basado en análisis semántico latente (LSA), que posteriormente utilizan para el aprendizaje y clasificación de las páginas.

Según resultados de sus propios autores se consigue una mejora de un 12,9% de la prueba F sobre la línea base de la clasificación por bolsa de palabras, e inciden en la necesidad de obtener mejores métodos de resumen de textos para conseguir mejorar la clasificación automática de los mismos.

2.2.3 Aproximaciones basadas en uso de características estructurales

Una línea de investigación que difiere de lo anterior por no hacer uso de las características textuales, únicamente de las derivadas de la estructura de la web es la propuesta por [Lindemann 2007]

En ella se parte de un conjunto de 8 categorías generales de páginas Web, a saber, *Académicas, Blogs, Comunidades, Corporativas, Informativas, Sin provecho, Personales y Tiendas,* y se propone un conjunto de 30 características meramente estructurales como la media de enlaces externos al sitio, el número de páginas conocidas del mismo, la fracción de documentos pdf/ps que aparecen, la fracción de páginas con scripts o la media de dígitos en la ruta de la url, entre otros.

Mediante un modelo de aprendizaje probabilístico como Naïve Bayes y por validación cruzada realizan un experimento de clasificación sobre cerca de cien millones de páginas obtenidas a partir de un crawl de dieciséis mil sitios web seleccionados de un directorio público y se compara con una clasificación basada exclusivamente en contenido.

Los resultados mostrados son interesantes cuando se eliminan algunas categorías muy relacionadas entre sí, como los *Blogs, Comunidades, Informativas y Personales*. El experimento concluye mostrando el interés de realizar una preclasificación por estos métodos para mejorar la clasificación basada en contenido, obteniendo según resultados de los autores una media el 92% en la prueba F.

Una aproximación no del todo diferente a la anterior es la propuesta por [Kan 2004] donde se basa únicamente en las características extraídas de la URL para realizar la clasificación, extrayendo para ello las palabras que aparecen en su ruta o que se pueden inferir mediante un autómata de estados finitos.

En contra de las recomendaciones del World Wide Consortium (W3C) sobre la opacidad de las Urls, la mayoría de diseñadores intentan tomar para sus páginas un dominio descriptivo de las mismas, de lo cual se puede llegar a pensar que si un humano de un simple vistazo puede saber a qué se refiere una url concreta, por algún prefijo en la misma, o alguna palabra en su ruta, se puede entrenar a una máquina para hacerlo de manera similar.

El autor para ello utiliza un autómata de estados finitos que le permite obtener los atributos de la url a partir de su semejanza con el espacio de características predefinido (su corpus), por ejemplo, si en el corpus aparecen las palabras "new york times" y en la url aparece "nytimes" dicho autómata encontrará una concordancia y expandirá la url a la característica adecuada.

Los resultados obtenidos son bastante interesantes, comparados con la clasificación sobre características obtenidas del texto completo, del título y de los anclajes, y una combinación de los mismos. En algunos casos muestra un valor superior, lo que demuestra su competitividad. Además, al no ser necesario descargar la página completa ni analizar su contenido la velocidad de análisis y clasificación es muy superior, siendo un método muy apropiado para la clasificación online en tiempo real.

[Shih 2004] siguen un poco la línea anterior de análisis de la url para el bloqueo de spot publicitarios en la Web, aunque variando ligeramente la forma de interpretar las Urls.

Los autores parten de la idea de que ciertos prefijos usados en la ruta de las Urls suelen ser comunes para los contenidos publicitarios, así como ciertas secciones de las páginas Web son utilizadas para contenerlos.

Para el análisis de las Urls los autores generan un árbol a partir de su ruta y realizan comparaciones emparejando los diferentes niveles generados. Su aplicativo está orientado más directamente con la operativa comercial, como el bloqueo publicitario para el cuál ha sido diseñado o el control parental de contenidos no deseados.

Los resultados para el bloqueo de publicidad basado en el análisis de la url muestran que se obtienen tasas competitivas con productos comerciales sin la necesidad de intervención humana para ello, pero la principal pega que tienes es la gran dependencia con el site concreto para el cuál se entrena, lo que se soluciona con un entrenamiento previo sobre el mismo.

En el mismo estudio se propone un método para analizar la estructura jerárquica de las páginas mediante el análisis de su estructura tabular, determinando mediante emparejamiento como en el caso anterior, por su situación en la jerarquía, qué enlaces pueden ser de interés para el usuario y qué secciones pueden ser contenedoras de elementos publicitarios.

Comparando el método propuesto con sistemas como NewsDude [Billsus 1999], que utiliza un modelo basado en contenido para clasificar las diferentes páginas destino y proponer la más interesante para el usuario, o Newblasters [Barzilay 2003] o Google [news.google.com] que buscan sobre múltiples sitios predefinidos y proponen un sumario de los contenidos recomendados, los autores muestran que su método además de recomendar contenidos no preclasificados lo hace de manera que mejora el rendimiento de los anteriores, aunque el principal problema del que adolece es de la necesidad de un entrenamiento completo sobre el sitio sobre el que va a trabajar, no pudiendo realizar ninguna predicción sin que este entrenamiento se complete.

2.2.4 Aproximaciones basadas en hipertexto

Pero la mayoría de líneas de investigación actuales buscan mejorar la clasificación añadiendo información contextual a la clasificación basada puramente en el contenido.

Desde este punto de vista se puede dividir la investigación en tres grandes líneas, la que se basa en el hipertexto, la que se basa en el análisis de los enlaces y la que se basa en la vecindad.

En la aproximación de hipertexto se extraen las características de los anclajes de los enlaces, de palabras extraídas de las páginas a las que apuntan, de los encabezados que los preceden o de las palabras de alrededor de los mismos.

Estudios como [Furnkranz 1999], [Glover 2002] y [Sun 2002] muestran un aumento considerable de los resultados usando la información de los enlaces frente a la clasificación textual base.

[Sun 2002] realizan un estudio comparativo entre los modelos aprendidos a partir de representaciones puramente textuales, y mejoras a las mismas mediante la introducción de información del título, las palabras de los anclajes y la combinación de ambas, utilizando para ello clasificadores basados en máquinas de vectores soporte (SVM)

Los resultados muestran una mejora considerable de la utilización de la información del texto de los anclajes frente a la clasificación estándar, así como que la utilización de la información del título, por sí sola, no es consistente y homogénea para todas las categorías, poniendo más de manifiesto que la combinación con el texto de los anclajes es una buena aproximación para aumentar el rendimiento.

2.2.5 Aproximaciones basadas en análisis de los enlaces

En la aproximación basada en análisis de los enlaces se combina el análisis de los anclajes anteriormente descrito con el análisis textual de las páginas referenciadas.

En los estudios de [Calado 2003] se muestra cómo al usar información de los enlaces y añadirla a la clasificación por métodos basados en contenido se alcanza una ganancia de hasta 46 puntos de F1 sobre ésta última, pero en cambio también muestran que se introduce mucho más ruido.

Para su estudio los autores tienen en cuenta las siguientes asunciones:

- Si dos páginas están enlazadas es de sentido común pensar que se refieren a un tema común.
- Si una página está enlazada o apuntada por muchas otras páginas, se podrá asumir que su contenido es importante.

La mayoría de motores de búsqueda actuales tienen en cuenta estas asunciones. Así por ejemplo HITS se basa especialmente en la segunda para categorizar como páginas puente o páginas autoritarias las páginas que cumplan respectivamente las características de apuntar a muchas páginas importantes y ser apuntadas por muchas páginas que son importantes.

Basándose en lo anterior y en la evidencia de que una clasificación puramente textual basada en el contenido obtiene unos resultados pobres en el contexto de la Web, debido al ruido introducido por el uso de pequeños textos independientes, las imágenes, los scripts y toda la información multimedia que no se puede analizar de manera textual, proponen un método para combinar la información textual con la información obtenida por análisis de los enlaces.

Para el análisis de los enlaces hacen uso de medidas de similitud de los mismos como la *co-citation*, que consiste en que dos páginas son co-citadas si una tercera página las cita a ambas, la *bibliographic coupling,* dónde dos páginas la cumplen si ambas citan a una tercera página, o la combinación de ambas en la medida *Amsler,* que se define de manera que dos páginas son relacionables si ambas están citadas por una tercera página, ambas citan a una misma página o una de ellas cita a una tercera página que a su vez cita a la otra página.

Los autores combinan mediante un modelo de red bayesiana la evidencia de la clasificación basada en contenido con la evidencia de la clasificación basada en la estructura de enlaces.

Los resultados muestran una mejora considerable frente a la clasificación textual estándar, pero ponen de manifiesto la necesidad de un análisis previo de la estructura de páginas para determinar y calcular las anteriores medidas de similitud, para lo cuál se necesita un conjunto de páginas bastante elevado e interrelacionado entre sí.

Así mismo la utilización de la información extraída del análisis de los enlaces, por sí sola, introduce un ruido considerable que ha de ser filtrado de algún modo, para lo cuál la combinación con un modelo basado en contenido ha demostrado ser una buena opción.

Otros estudios que se basan en el análisis de los enlaces son [Slattery 2000], dónde se hace uso del algoritmo HITS para explorar la topología de hiperenlaces, o [Joachims 2001] que hace uso de la combinación de funciones núcleo de las máquinas de vectores soporte para utilizar conjuntamente la información textual con el análisis de la *co-citation*

2.2.6 Aproximaciones basadas en vecindad

En el análisis de vecindad la estimación de la categoría de un nuevo documento se realiza sobre la base de los vecinos previamente clasificados.

[Chakrabarti 1998] utiliza la información sobre las categorías de los documentos utilizados para el entrenamiento como base para la estimación de la categoría de los nuevos documentos que se presentan, pero en lugar de realizarlo mediante la estimación directa de los documentos de la vecindad hace uso de estrategias basadas en la *co-citation*, demostrando que sus resultados son superiores de este modo.

Otros estudios como [Oh 2000] mejoran el proceso de clasificación basado en vecindad mediante el filtrado de los documentos utilizados para el entrenamiento y posterior clasificación.

2.2.7 Aproximaciones comerciales

Desde el ámbito comercial existen grandes productos orientados a la extracción de conocimiento desde textos. La mayoría de dichos productos, como [SAS Text Miner], conjunto de aplicaciones y herramientas para realizar tareas de clasificación, categorización y modelado a partir de conjuntos de documentos estructurados o no, y [Megaputer Text Mining Technology], que permite análisis semántico y lingüístico, clasificación, categorización y visualización de patrones, permiten la minería en documentos html, lo que permite la realización de esta extracción desde un medio como la Web, integrando el software de análisis directamente en el sitio web. Otro paquete comercial en esta línea es [ISYS Search Software]

Aunque la comercialización de productos de minería web se ha decantado más hacia el análisis de uso, como muestran gran parte de los paquetes del sitio [KDnuggets Web Mining], el interés por la minería web para clasificación se ha orientado especialmente al control parental y la monitorización tanto de empleados en las empresas, como de menores en el hogar, y gran prueba de ello es la ingente cantidad de productos que surgen con esta intención, algunos de los cuales son los siguientes [CYBERsitter], [GFi Web Monitor] o [Spector Soft]

En cualquier caso la investigación se ha centrado mayoritariamente en conseguir una clasificación de páginas web dentro de alguna de las categorías previamente definidas para el dominio general de la misma. Para ello por lo general se han entrenado

modelos que han discriminado entre las diferentes categorías, en lugar de permitir la categorización en más de una de ellas.

En nuestro proyecto hemos ido un poco más allá proponiendo un clasificador binario para cada tipo de categoría, permitiendo de este modo categorizar las páginas en más de una clasificación posible, pues en el caso que nos ocupa no son discriminantes entre sí, así como permitiendo la ampliación y extensión a diferentes dominios de la Web.

Es por lo anterior que existe una dificultad añadida y es precisamente la posibilidad de que ciertas páginas puedan pertenecer a más de una clasificación, lo que hace del espacio de valores subconjuntos no disjuntos, implicando conceptualmente que sus características, en algunos de los casos, tenderán a ser muy similares, con la consiguiente dificultad para mantener un bajo ratio de falsos positivos, y con ello un valor conjunto del estadístico F bastante bueno.

Además el conjunto de datos de entrenamiento es limitado y muy variable lo que condicionará en gran medida los resultados obtenidos, pero a su vez servirá de base para obtener una comparativa con métodos clásicos estudiados en el estado del arte, lo que permitirá inferir las características de los modelos propuestos para su extensión a otros dominios diferentes.

CAPÍTULO 3 EXPERIMENTOS DE CLASIFICACIÓN

3.1 MARCO GENERAL DE EXPERIMENTACIÓN

Para la consecución de los objetivos del proyecto se deberá realizar un estudio del problema desde el punto de vista de las diferentes alternativas que existen para abordarlo, eligiendo la que proporcione el método más eficaz para conseguirlo.

Para ello se determinará cada una de las posibles alternativas y se estudiará con detenimiento sus características, eligiendo aquella alternativa cuyas características, a priori, parezcan resultar beneficiosas con relación al trabajo a desarrollar.

El estudio de las alternativas estratégicas nos da un punto de partida para determinar qué línea de investigación seguir de entre todas las alternativas posibles, y definir a partir de ella los experimentos.

Puesto que la investigación es un proceso circular, en el que en más de una ocasión se vuelve al principio, la identificación de alternativas es un paso previo de vital importancia para saber por dónde empezar, así como un punto de retorno dónde anotar y evaluar nuevas alternativas cuando la vía seguida no haya dado sus frutos.

Es por tanto que en las siguientes líneas se identificarán las diferentes alternativas que se han planteado desde el principio, las que se han seguido y no han dado sus frutos y por qué, las que no se han seguido y por qué, y definitivamente cuál se ha llevado a cabo y es el resultado de esta memoria.

Es por ello que aunque el estudio de alternativas es el comienzo, no así queda en el olvido sino que se vuelve sobre él tantas veces como sea necesario hasta llegar al resultado esperado, momento en el cuál se vuelve aún atrás a determinar el éxito de la estrategia seguida.

En el estudio de alternativas se tiene en cuenta diferentes aspectos del proyecto, como los atributos a seleccionar y por tanto la representación de los documentos, los tipos de clasificadores utilizados y la decisión final o los métodos de evaluación posibles y sus ventajas e inconvenientes, así como la metodología y herramientas utilizadas.

3.1.1 La selección de atributos para la representación del documento

La selección de atributos, en el contexto Web, equivale a la representación de los documentos de manera formal y que sirva como base de comparación entre los mismos, permitiendo el aprendizaje de un modelo inductivo capaz de generalizar a partir de dichos atributos y predecir la clase de nuevos documentos que se le presenten.

Como se ha visto en el estado del arte, la investigación ha derivado por múltiples vías para representar los documentos Web, desde concepciones basadas únicamente en su contenido, como la BoW, hasta concepciones basadas únicamente en su estructura, como las basadas en conteo de aparición de determinados elementos Html, pasando por otras muchas intermedias.

En este apartado tomaremos en cuenta algunas de estas posibilidades, evaluaremos su utilidad, así como propondremos alternativas nuevas, y decidiremos qué implementar y por qué motivo.

- En primer lugar hay que considerar la alternativa BoW, explicada en la introducción, y tan utilizada como modelo base sobre el que realizar las comprobaciones.

  Utilizando esta representación, se puede obtener la representación desde diferentes partes del documento, y por tanto conjuntos de características. Así pues se puede leer sólo el contenido textual de la página, equivalente a lo que el usuario vería al navegar por ella con un navegador textual, se podría utilizar el contenido pero dando mayor importancia a los títulos, encabezados y demás textos marcados por el usuario, y se podría dar información al contexto de los enlaces.

  Todo ello compone las siguientes alternativas:

  - BoW del cuerpo, que es la que se suele utilizar como *baseline* para las comparaciones
  - BoW del cuerpo con mayor ponderación para el texto remarcado y/o los enlaces
  - BoW de las Urls

- La combinación de BoW con información estructural, o la utilización de información estructural únicamente, como número de páginas conocidas, número de enlaces, de imágenes, de documentos,... y todas aquellas que se nos puedan ocurrir.

  Se han hecho pruebas basadas en una combinación de características de este tipo y no han dado resultados demasiado interesantes. Por un estudio visual de las páginas parece que características como el número de imágenes, el número de palabras o demás no aportan significativamente información ya que diferentes categorías comparten muchas de ellas, e incluso páginas de una misma categoría no poseen unas ratios homogéneas de las mismas, por lo que es una línea de investigación que no se ha decidido seguir.

- BoW de características que consigan plasmar la intención del creador de la página Web por las cuáles se conozca la clase de contenido de la misma.

  Se consideran apropiadas por incluir la intención del usuario de remarcar los contenidos.

  Así pues las palabras que generalmente se encuentran en los meta-tags de la cabecera, como description o keywords, además del título, suelen incluir la información que el creador de la página desea que sea indexada por los motores de búsqueda y por tanto define el contenido de su página.

Así pues, una página de libros pretenderá que sea reconocida como tal por los motores y sea devuelta al buscar en ellos por algún concepto relacionado con los mismos.

La misma idea se puede aplicar a los festivales, las revistas o cualquier otra categoría.

La idea de tomar todo el contenido de los meta-tags no ha sido explotada, limitándose en la mayoría de los casos a incrementar la ponderación para las palabras del título. Así pues, sí se ha realizado experimentos de clasificación sobre resúmenes, cosa que viene a ser equivalente al contenido de los meta-datos de las páginas Web.

El título es el meta-tag por excelencia. Es el texto que generalmente ponen todos los motores de búsqueda en sus resultados, así como uno de los textos más importantes a la hora de generar el índice.

Pero no todos los creadores de páginas Web lo utilizan como debieran. En la mayoría de casos lo utilizan como un anuncio de marketing, con el nombre propio del producto en lugar de su descripción, lo que genera unos textos no útiles para clasificación.

En las campañas de SEO [Marckini] generalmente una de las primeras cosas que se realiza es la modificación de dicho título para incluir las palabras que mejor definan a la Web en cuestión, y aquellas por las que se desee que sea indexado en los motores, repitiendo palabras de los meta-tags.

Es por ello que la utilización de toda la cabecera en lugar de únicamente el título puede aportar información significativa para discriminar y aprender un modelo más generalizado.

Pero en numerosos estudios [Pierre 2000] se habla de la irrelevancia o incluso inconsistencia del contenido de los meta-tags. Muchas páginas ni siquiera los incluyen, por un mal diseño, o los incluyen pero no son consistentes con el contenido de la página.

En el caso de la clasificación que nos ocupa nos llevaría hacia una clasificación incorrecta del documento, pero, ¿a qué atenerse, a una posible clasificación incorrecta del documento por un encabezado que pueda estar mal definido, o a una clasificación pobre basándose en un contenido tan poco discriminatorio entre los diferentes tipos de categorías?

La categorización de páginas de teatro dentro de subcategorías como las que ocupan el proyecto es una tarea compleja si se lleva por la rama del contenido, ya que páginas de diferentes categorías pueden tener contenidos muy similares. Así pues, una página de formación dónde se hable de una asignatura de teatro contemporáneo tendrá un conjunto de palabras muy similar a un texto sobre teatro contemporáneo. Una página de Blog será prácticamente imposible saber que es un Blog por su contenido pues hablará de prácticamente cualquier cosa relacionada con el teatro.

Es por ello que todo lo que pueda añadir cierta discriminación entre diferentes categorías y añadir cierta homogeneidad entre las mismas es una vía interesante de investigación, y es por tanto que el presente trabajo se ha centrado en explotar esta vía, junto con la siguiente mejora.

- Cada vez más los motores de búsqueda o los filtros de Internet hacen uso de información contextual, obtenida de los enlaces, o inmediaciones de los mismos, que componen la página Web.

  Algunos trabajos que se vieron en el estado del arte tienen en cuenta esta información, consiguiendo resultados bastante interesantes como para tenerlos en cuenta.

  La clasificación basada en Urls es muy interesante, algo tan pequeño y rápidamente accesible como es la url puede aportar gran información. Su principal problema es que por sí sola, en muchos de los casos en los que contiene nombres propios o enmascara un sitio web completo, mediante marcos por ejemplo, no serviría para una clasificación. Pero es una opción interesante para añadir certeza en una clasificación.

  Un par de ejemplos sencillos serían los siguientes:

  - Una página que posiblemente sea una revista, si en su url aparece una subruta o una coletilla que diga revistas la certeza aumenta significativamente.
  - Una página que parezca un Blog, pero no se diferencie de un foro, o un rincón de noticias, con una url que ponga blogspot, o weblog, o algo similar, prácticamente nos da la certeza de lo mismo.

  Es por ello que añadir información de este tipo a la clasificación añadirá certeza en los casos positivos de clasificación.

- Al igual que lo anterior, la utilización de la información contenida en los enlaces es de gran importancia para mostrar la intención del autor a la hora de crear el mismo.

  Así pues, los textos que aparecen en los enlaces, si quieren aumentar la confianza de los usuarios y hacer que se atrevan a navegar hacia delante, deben ser lo suficientemente claros sobre el destino de los mismos, por lo que la información que aportan es significativa, por ejemplo, una revista tendrá textos que nos lleven a los números de la misma, a las secciones, a las noticias, a la publicidad, a tal artículo, al índice, a un sumario, a una entrevista, etcétera. Y una página de festivales nos llevará a los eventos, a la programación, a los patrocinadores, colaboradores y organizadores... o una librería a la cesta de la compra, a realizar el pedido, al detalle de un libro, a los medios de pago o formas de envío.

  Es por ello que estas características también pueden añadir información a la clasificación.

- BoW de características que se consideran únicas o apropiadas para la representación de un documento por las características intrínsecas del mismo. En este caso se puede considerar la estructura propia de los Blogs, en forma de diario y que se describe en el experimento 3.10, dónde a simple vista podemos distinguir partes o estructuras bien diferenciadas, comunes a todas las páginas de este tipo y que difiere del resto de páginas, por lo que pueden resultar en una buena aproximación para la clasificación de los mismos.

  Así mismo, una representación de este tipo podría servir para representar páginas referentes a librerías online, donde una serie de características fijas como números ISBN, cesta de la compra, pago por tarjeta... suele estar presente, o cualquier otra página, siempre que se explore el conjunto de características propias de la misma.

Las vías anteriores son las diferentes alternativas que se han considerado en la investigación.

La basada en BoW a partir del contenido del documento se utilizará como método base para la comparación de los demás métodos, pudiendo de este modo realizar un estudio comparativo y determinando si las mejoras obtenidas son significativas estadísticamente hablando y por lo tanto el trabajo de investigación ha dado sus frutos.

Los diferentes experimentos se orientan a considerar los pros y los contras de los diversos métodos del estado del arte y que en este apartado se han tenido en consideración, de modo que se pueda extraer una idea de cómo utilizarlos o combinarlos para conseguir una representación más robusta.

El método de mejorar el BoW de contenido añadiendo información a partir de los textos del título, enlaces, cabeceras y demás se implementa para servir de segunda base frente al anterior y comprobar la bondad de cada uno de ellos en los diferentes tipos de documentos, viendo la variabilidad de los mismos y comprobando si esta información puede apoertar beneficio a la propuesta.

El método de clasificar mediante las palabras de la Url se implementa para comprobar la bondad del mismo e intentar ver si realmente su utilización puede resultar en una mejora del método propuesto.

El método que se propone pues es la combinación de la información de la cabecera de la página, los textos de los enlaces y las palabras de la Url, la representación que denominamos h&l&u. A diferencia del BoW mejorado propuesto en segundo lugar, dónde el conjunto de características es único y equivalente a las palabras obtenidas del corpus del mismo, y dónde la aparición de una determinada palabra en un determinado tag que la ensalce hace que su ponderación sea mayor, en el método propuesto aquí la combinación se realiza triplicando el conjunto de características. Así pues, una palabra aparecida en la url compondrá una característica diferente a la aparecida en los enlaces, así mismo diferente a la aparecida en la cabecera. De este modo el conjunto de características se dispara al triple, por lo que será necesario realizar cierto tratamiento del corpus utilizado, como se verá a continuación.

Los diferentes experimentos se orientan a comprobar la validez de esta propuesta mediante su comparación con las técnicas actuales.

En último lugar, para los Blogs, se decide realizar un estudio más concreto y representar los mismos bajo una serie de características especiales que sólo ellos comparten. Como se verá en el apartado de evaluación, los métodos anteriores no tienen muy buenos resultados. Los basados en BoW son bastante pobres, dando un elevado número de falsos positivos. Todo ello se debe a la gran variedad de temas que se tratan en los mismos, y por lo tanto habrá que buscar una representación más específica al margen de su contenido.

3.1.2 El corpus en una representación BoW

Las posibilidades respecto al corpus son varias. Por un lado se podría obtener un corpus creado por terceros sobre el ámbito específico del teatro y utilizarlo para la clasificación BoW, o bien crear el corpus a partir del conjunto de documentos que se tiene inicialmente.

Se opta por la segunda opción de crear el corpus a partir de las páginas que componen el repositorio inicial, dentro de lo cual existen varias posibilidades.

Por un lado se podría crear el corpus a partir de todas las palabras que componen el documento, generando un gran corpus con terminología referente a teatro, y hacer luego el tipo de limpieza necesario.

El principal problema de la utilización de un corpus así en el ámbito del teatro es que muchos conceptos son bastante comunes a diferentes categorías, con lo que se generará un espacio de características común a muchas de ellas y que no servirá para discriminar, pudiendo derivar en un mayor número de falsos positivos para las diferentes categorías.

A su vez, la generación de un corpus de tales dimensiones provocará un aumento considerable de la dimensionalidad. Se puede tratar de reducir mediante alguna de las muchas vías de selección de atributos supervisadas como las de [Forman], [Guyon] o [Zheng], pero con ello se corre un riesgo, eliminar características que aporten poca información en general pero sean muy representativas de un tipo concreto en particular. Esto por ejemplo podría suceder con palabras que en conjunto de documentos aparecen muy pocas veces, que incluso aparecen en muy pocos documentos, pero que su aparición va ligada casi seguro a una categoría determinada.

Por ello, una opción como la anterior requerirá de un análisis manual de las características, comparando frecuencias inversas de aparición de la característica en el conjunto total y determinando su idoneidad para representar a los documentos.

Si a lo anterior añadimos que el método de representación propuesto, h&l&u requiere del triple de características de un BoW estándar, el número de dimensiones aumentará considerablemente, con lo que se requerirá una selección de atributos más agresiva y por tanto, dada la diferente distribución entre clases de documentos, una pérdida de representatividad de los mismos.

Por tanto hay que buscar una alternativa que permita mantener la generalidad de la representación pero se adecue a las necesidades y características propias de cada categoría de documentos.

Por ello la idea de crear un corpus independiente para cada tipo de página o categoría parece interesante desde el punto de vista de que se acercará más a la representatividad de la misma, las palabras así extraídas serán más similares desde el punto de vista semántico a la categoría que representan, y desde el punto de vista de la reducción de la dimensionalidad, pues el conjunto de palabras tenderá a ser menor.

Por lo tanto, en la solución propuesta se construye un corpus a partir del conjunto de Webs dadas inicialmente para cada una de las categorías, obteniendo un corpus para cada una de ellas y que será el utilizado para extraer y representar cada uno de los documentos mediante el método H&L&U.

Una cosa más a tener en cuenta es el modo de obtener el vector de palabras a partir de las páginas. Se podría obtener un vector de palabras a partir de todas las palabras que incluyen el documento, previa eliminación de los tags html, o bien se podría obtener sólo a partir del cuerpo del mismo, y así otras muchas alternativas.

Concretamente, para la representación BoW estándar, mejorada y Urls se crea el corpus a partir de las palabras que componen el cuerpo del documento, sin tener en cuenta el encabezado del mismo ni la url.

Para la representación H&L&U se obtiene el corpus a partir de las palabras que forman parte de la cabecera del documento, el texto tanto de los anclajes como de la url de los enlaces, y el de la url del documento actual.

De este modo ambas representaciones se ajustan al conjunto de palabras que mejor definen el contenido que están representando, y así mismo, en el caso de H&L&U disminuye significativamente el tamaño del mismo, y con ello la dimensionalidad final.

En la generación del corpus se tendrán en cuenta ciertos aspectos lingüísticos que se describen a continuación.

3.1.3 Tratamiento de la dimensionalidad y los problemas lingüísticos

El problema de la dimensionalidad aparece en toda actividad de minería de datos encaminada a extraer información de textos, y en el caso de la Web se ve multiplicada por la gran variedad de autores y contenidos de la misma.

En múltiples estudios como [Forman], [Guyon] o [Zheng] se habla de métodos y algoritmos para reducir drásticamente la dimensionalidad y hacer más eficientes los algoritmos de aprendizaje. En algunos de los casos estos algoritmos no son capaces de trabajar con dimensionalidades tan elevadas, cayendo en picado su rendimiento cuando se requiere cierta escalabilidad [Rangel 2007]

En el problema actual no se ha utilizado ningún método de reducción automática, aunque supervisada, de la dimensionalidad. Se hicieron pruebas que no

resultaron en mejoras considerables de la velocidad y sí empeoraban en algunos de los casos los resultados de nuevas clasificaciones. Pero sí se han realizado acciones para reducir en la medida de lo posible la dimensionalidad de manera manual.

En primer lugar con la creación del corpus se ha obtenido el número de apariciones de cada palabra, eliminando todas aquellas que aparecen un número demasiado limitado de palabras, no siempre el mismo, pero normalmente con eliminar todas aquellas palabras que sólo aparecen una vez se ha conseguido reducir el tamaño entre un cuarto y la mitad, tal y como se verá en el experimento 3.6

Para no arriesgar y determinar de manera fortuita un punto de corte, se ha comparado el conjunto de palabras de cada corpus con el conjunto total de las palabras de un corpus generado para todas las categorías, obteniendo una ratio de aparición entre el corpus específico y el general. De este modo, palabras que aparecen muy poco no serán representativas pues no aparecen ni siquiera en la mayoría de páginas de una misma categoría, y palabras cuya ratio es muy baja significará que aunque pueda aparecer muchas veces en una determinada categoría, también así es en otras categorías, lo que hará que tampoco sea demasiado discriminatoria y por tanto aporte mucha información para la clasificación.

Combinando de manera razonable las dos aproximaciones anteriores se consigue una reducción de la dimensionalidad suficiente para un tratamiento eficiente del algoritmo de aprendizaje.

Otro problema que afecta a la dimensionalidad, aunque no sólo a ella, sino que además provoca que las características tengan menos poder discriminatorio son los problemas lingüísticos.

El problema de la aparición de múltiples palabras con una raíz común, y que por tanto aportan un significado similar al documento es muy común en el problema de la categorización de textos. Es por ello que es necesario algún proceso de tipo lingüístico o computacional que reduzca todas estas variaciones a su forma o lema común. Para ello existen analizadores como el de Porter [Porter 1980] y algoritmos de *stemming,* que realizan de manera algo más relajada la reducción a la forma común eliminando prefijos y sufijos, pero sin llegar a realizar un análisis lingüístico exhaustivo.

Se hace uso de un *stemmer* de este tipo para reducir la dimensionalidad y aumentar el rendimiento, tal como muchos estudios ponen de manifiesto [Arregi], y se lleva a cabo un experimento para corroborarlo.

Por último se hace una limpieza de palabras frecuentes del idioma y que no aportan información, las denominadas palabras vacías o *stop words,* eliminando las mismas del conjunto de palabras incluidas en el corpus, reduciendo de este modo la dimensionalidad eliminando características que no aportan ningún tipo de información.

3.1.4 Asignación de valores a la bolsa de palabras

La función para asignar valores o pesos a los diferentes atributos de la BoW es la función local de frecuencia normalizada.

Los motivos que abogan por esta decisión, si nos hubiéramos basado en un corpus global, se basan principalmente en la disparidad de textos (y tamaños de los textos) en las diferentes clasificaciones dadas, con lo cual una función local tenderá a catalogar los textos por su información local y no por la relación con el conjunto de todos los textos, pues de esta última manera se daría mayor importancia a aquellos cuya clasificación fuera mayoritaria.

Puesto que por el mismo motivo nos decidimos por un corpus local para cada categoría, la decisión quedaría restringida a una asignación global al conjunto de páginas de una misma categoría o local a la misma.

Puesto que no existen estudios teóricos ni resultados empíricos que apoyen una mejora significativa en el uso de una función global frente a una función local, la decisión tomada es la de usar una función local, de cómputo menos costoso y resultados satisfactorios.

3.1.5 Modelo de clasificación

Como se vio en la introducción, el proyecto en realidad persigue la categorización de las páginas web teatro en las posibles categorías a las que pertenezcan, pero como se vio, este problema de categorización se puede resolver mediante sucesivos clasificadores binarios para cada una de las categorías.

Por tanto una de las primeras cuestiones es la de decidir entre usar un único clasificador, o utilizar tantos como categorías hay de manera binaria, aunque queda bastante claro de la necesidad de usar un clasificador binario por cada categoría y es esto lo que se va a implementar.

Una cuestión ligada a lo anterior es si utilizar el mismo método de representación para los documentos para todos los clasificadores, o realizar una representación específica para cada uno.

Vistas las alternativas de representación descritas al principio del presente punto, así como las diferentes evaluaciones realizadas y detalladas en el apartado de evaluación, se decide utilizar una misma representación, la basada en h&l&u para todos los clasificadores, y una representación específica de los Blogs basada en las características analizadas a propósito par allos, conformando con ello los diferentes experimentos en clasificación realizados

Por otro lado, como también se introdujo, son muchas las técnicas de aprendizaje, de las cuales, por los diferentes estudios referentes al aprendizaje de datos con alta dimensionalidad, son tres las que mejores resultados presentan, los separadores lineales de las máquinas de vectores soporte (SVM), los árboles de decisión C4.5 y los métodos bayesianos Naïve Bayes.

Tras varias pruebas con la implementación Weka de los tres clasificadores anteriores se decide la utilización de Naïve Bayes por obtener un resultado superior en la mayoría de los casos y por la mayor rapidez en construir y evaluar los modelos.

3.1.6 Respecto a los métodos de evaluación

Los métodos de aprendizaje descritos permiten construir modelos que generalicen el comportamiento de los datos dados como evidencia para predecir nuevos datos. En todos los casos es necesario evaluar la calidad de estos modelos, para conocer su nivel de precisión, y de este modo saber la adecuación al problema. La evaluación de los modelos es por tanto una de las etapas más importantes en la minería de datos pues es la que determina la validez de los mismos y por tanto el éxito en la tarea realizada.

En el caso concreto del trabajo los modelos que se construyen son clasificadores. Existen diversas técnicas para la evaluación de este tipo de modelos, encuadrándose en dos categorías principales, la evaluación de hipótesis basada en precisión, dónde se evalúa el porcentaje de error que se comete entre la hipótesis formulada y el valor real, con lo que el modelo evaluado de este modo deberá aprenderse teniendo prioridad el minimizar el número de errores cometidos, y la evaluación basada en coste, que determina el coste de los errores cometidos, con lo que no importará tanto el número de errores cometidos como el coste de los mismos, con lo que el modelo deberá ser aprendido minimizando dicho coste en lugar de minimizar el número de errores.

En el problema que nos ocupa, a priori, todos los errores tienen un coste similar, con lo que la evaluación que se efectuará será basada en la precisión, y por tanto los modelos se aprenderán intentando minimizar el número de errores.

La evaluación basada en costes se suele utilizar frecuentemente en las clasificaciones para filtros, dónde puede ser más costoso dejar pasar un documento al que no debe permitírsele el paso, a catalogarlo mal y que no pase un documento que no plantea conflicto alguno, como en el caso del filtrado parental o en la determinación de correo spam.

Dentro de la evaluación basada en precisión existen varios modos de obtener el porcentaje de errores cometidos, diferenciándose principalmente entre el error de muestra, que es el calculado a partir de los datos que se poseen, o evidencia, y el error verdadero, que es el calculado a partir de la distribución D, que es la que define la probabilidad de encontrar cada instancia de la evidencia en el espacio de todas las posibles evidencias [Hernández Orallo]

En nuestro caso vamos a utilizar el error de muestra pues utilizaremos la muestra etiquetada para la validación.

Para realizar la evaluación hay que considerar las alternativas existentes, ver las características de los datos que tenemos, y determinar las ventajas e inconvenientes de cada una, decidiendo utilizar la que incline la balanza hacia el lado de las ventajas.

De entre todas las alternativas como utilizar la evidencia completa, partir el conjunto en dos, entrenamiento y validación, o validación cruzada, directamente se elimina la primera posibilidad por realizar en la mayoría de los casos lo que se denomina *overfitting,* que no es más que un sobre ajuste completo de la función de aprendizaje a los datos disponibles, con lo que el resultado de la evaluación no es realista.

Las dos opciones que quedan tienen sus ventajas y sus inconvenientes, por lo que se analizan a continuación:

- Partición entrenamiento/prueba: El primer subconjunto se utiliza para aprender el modelo y el segundo para probarlo. Se evita completamente el problema del sobre ajuste, pero surgen un par de problemas. El primero es que el modelo aprendido es demasiado dependiente del modo en el cuál se ha realizado la partición, y el segundo es si tenemos pocos datos, al partirlos aún se tienen menos, lo que merma las capacidades de aprendizaje.
- Validación cruzada: El método consiste en dividir la evidencia en k subconjuntos, generalmente 10, y repetir k veces el entrenamiento con k-1 conjuntos y la validación con el conjunto resultante, obteniendo el error total como media de los errores medios. Aunque tiene algunos problemas como se describe a continuación

Con lo anterior parece clara la elección por el segundo de los métodos, pero en el problema que nos ocupa parece haber una característica que es necesario tener en cuenta para no cometer un error.

El número de páginas que se tienen de cada tipo de categorías en algunos casos es muy dependiente del dominio o site al que pertenecen. Dicho de otro modo, páginas diferentes puede haber un elevado número, pero muchas de ellas son páginas de un mismo sitio web, obtenidas mediante un crawl, y que por tanto pueden compartir demasiadas características comunes debido al estilo uniforme de un mismo autor, su vocabulario... Es por ello que una validación cruzada, al realizarse las particiones de manera aleatorias, si una determinada categoría tiene muchas páginas de un mismo sitio puede solapar muchas de ellas, creando un aprendizaje muy ajustado a las mismas, y si el conjunto de prueba ha recibido también muchas páginas de este sitio web seguramente obtendrá buenos resultados, por algo similar a lo comentario del overfitting.

Por ejemplo, simplificando mucho, si tenemos 1500 páginas de una categoría, de dónde hay 6 sitios Webs diferentes y uno de los sitios web lo componen 900 páginas, una partición de k=10 obtendrá 1350 páginas de entrenamiento frente a 150 de prueba, y es bastante probable 810 páginas de entrenamiento y 90 de prueba sean del mismo sitio web, lo que seguramente provocará un aumento de la precisión por un sobre ajuste.

Un problema similar lo defienden algunos autores como [Dietterich 1998] quienes proponen una modificación en la técnica que permita utilizar subconjuntos de entrenamiento independientes.

La ventaja de tener k subconjuntos de prueba independientes no sucede con los conjuntos de entrenamiento que comparten un alto porcentaje de los datos. Este solapamiento entre conjuntos de aprendizaje puede provocar que la evaluación sea demasiado ajustada a los datos y por lo tanto más alejada de la realidad.

[Dietterich 1998] propone una variación del método de validación cruzada al 5x2 Cross Validation consistente en realizar cinco repeticiones de la validación cruzada para el valor de k=2 obteniendo cada vez una partición diferente de los datos.

En el trabajo que nos ocupa, para limitar la complejidad de esta solución alternativa, y puesto que el mayor interés es demostrar la idoneidad de la solución propuesta frente a las conseguidas por las investigaciones del estado del arte, así como para separar el conjunto de los datos a partir del sitio web que los genera, se propone el siguiente método de evaluación para comparar resultados

Se realiza una partición entre entrenamiento y test realizada de manera manual, no aleatoria, separando completamente páginas de un mismo dominio, atendiendo a un porcentaje aproximado del 75/25%.

Se realiza un doble entrenamiento/validación, primero entrenando con una partición y validando con la otra y luego alrevés.

Al final se combinan ambas evaluaciones en una aproximación excesivamente simple al método propuesto por [Dietterich 1998] pero que sirve de base para la comparación, y que denominamos 2x2.

Al margen de la técnica elegida para evaluar los modelos hay que definir las medidas utilizadas para ello. La evaluación mediante Weka es una evaluación basada en la precisión. Para ello utiliza una serie de indicativos como TP (*True Positive*), FP (*False Positive*), Precisión, Alcance y estadístico F. Así mismo define una matriz de confusión dónde se indican tanto los TP como los FP de las clases evaluadas. A partir de todos estos datos se puede calcular el intervalo de confianza del error real a partir del error muestral. Veamos todo ello a continuación:

Matriz de confusión: Es la matriz que muestra la distribución de instancias que han ido a parar a cada clasificación. Para un par de categorías, A y B, la matriz bidimensional mostrará las instancias que siendo de una clase han ido a parar a dicha clase, en este caso los TP de ambas, y que determinará la diagonal principal de la matriz, y las instancias que siendo de una clase se han clasificado como la clase contraria, lo que mostrará la diagonal inversa:

Clase A	Clase B	<- Clasificado como
TPA	FPA	Clase A
FPB	TPB	Clase B

FIGURA 3.1: Matriz de confusión

TP (*True Positive*) es el conjunto de instancias que se han clasificado como una determinada clase cuando realmente son de dicha clase. Es decir, están bien clasificadas.

La medida TP para cada clase se definirá como:

$$TP = \frac{TPc}{TPc + FPc}$$

FIGURA 3.2: True Positive

Donde c indica la celda correspondiente a la fila de esa clase.

FP (*False Positive*) es el conjunto de instancias clasificadas como una determinada clase sin realmente serlo. Es decir, estando mal clasificadas, se asigna este valor a la clase que recibe incorrectamente esta clasificación.

La medida FP para cada clase se definirá como:

$$FP = \frac{FPnc}{FPnc + TPnc}$$

FIGURA 3.3: False Positive

Donde nc indica la celda que corresponde a la fila de la otra clase

Precission (precisión): Es la proporción de documentos de una clase que se han clasificado correctamente como de dicha clase. Mide la probabilidad de que si el sistema clasifica un documento en una determinada categoría, el documento realmente pertenezca a dicha categoría

$$p = \frac{TPc}{TPc + FPnc}$$

FIGURA 3.4: Precission

Recall (alcance): Es la proporción de documentos, que siendo de una clase, se han clasificado correctamente. Mide la probabilidad de que si un documento pertenece a una determinada categoría, el sistema lo asigne a dicha categoría. Da una aproximación a la sensibilidad de la función clasificadora

$$r = \frac{TPc}{TPc + FPc}$$

FIGURA 3.5: Recall

F: La medida F combina la precisión y el alcance mediante una ponderación de ambas medidas. En nuestra evaluación se ha tomado que tienen la misma importancia la precisión que el alcance. La medida F permite obtener un único valor de la calidad de los modelos, lo que hace más fácil su comparación

$$F = \frac{2pr}{p + r}$$

FIGURA 3.6: Estadístico F

Intervalos de confianza de la evaluación

Dada una muestra S de n ejemplos tomada a partir de una función objetivo f con una distribución D, es posible establecer unos intervalos de confianza para el error verdadero (errorR(h)) de una hipótesis a partir del error de muestra (errorS(h))

Aunque la distribución que se debería utilizar es la polinomial, para valores de n mayores de 30 se puede utilizar la distribución normal, facilitando de este modo el cálculo.

Con ello, a un nivel de confianza de c% se puede determinar el intervalo del error como:

$$errorR(h) = errorS(h) \pm z_c \sqrt{\frac{errorS(h)(1 - errorS(h))}{n}}$$

FIGURA 3.7: Intervalo de error

Dónde z_c se obtiene a partir de la distribución normal y el valor utilizado en las diferentes evaluaciones es de 1,96 equivalente al 95% de certeza.

Con esto se define el marco teórico y formal sobre el que se realizarán las validaciones de los modelos.

3.1.7 Entorno de desarrollo de la utilidad

Dos son los grandes mundos en lo que se refiere al desarrollo en la actualidad, Java y .Net, tal y como se muestra en estudios como el siguiente:

http://www.cs.berkeley.edu/%7Eflab/languages.html

Es indiscutible que existen partidarios de uno y partidarios del otro, así como también es indiscutible que razones no faltan para elegir uno de ambos.

Atendiendo a criterios objetivos más que puramente filosóficos se puede ver un análisis de rendimiento de varios lenguajes realizando diferentes operaciones en la siguiente web:

http://www.osnews.com/story.php/5602/Nine-Language-Performance-Round-up-Benchmarking-Math-and-File-IO/page3/

y que se puede resumir con el siguiente gráfico:

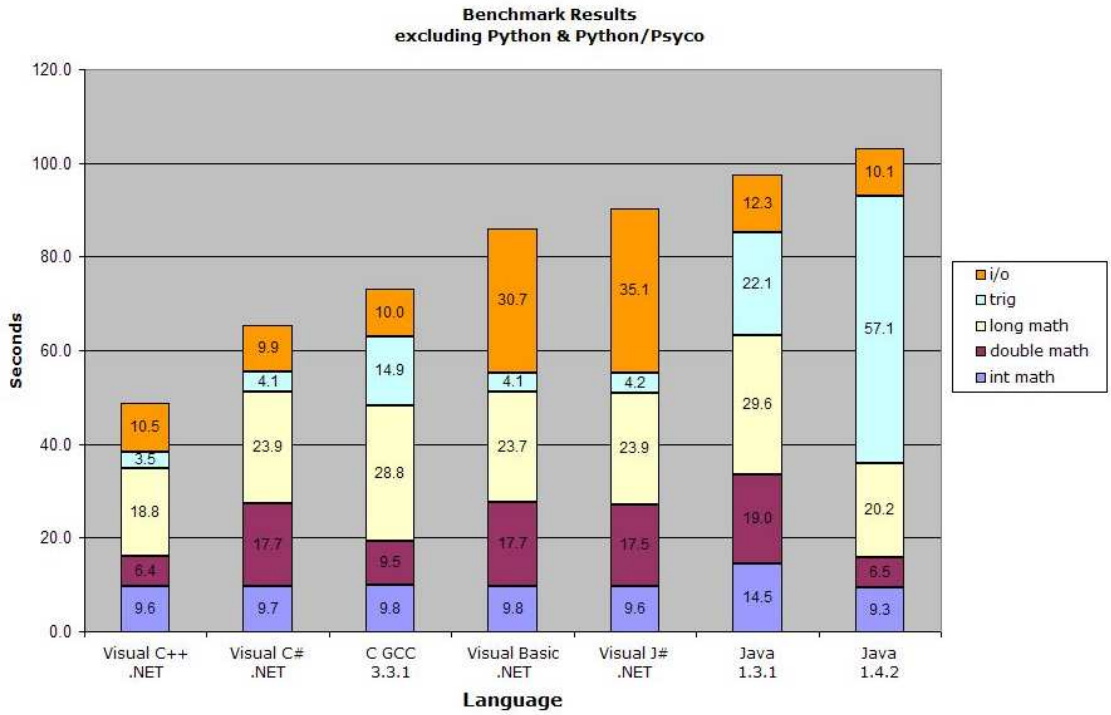


FIGURA 3.8: Rendimiento de los lenguajes de programación

Aunque también es cierto que se está ejecutando sobre un servidor Microsoft, lo que condiciona bastante el resultado.

Por otro lado en la Wikipedia:

http://en.wikipedia.org/wiki/Comparison_of_C_Sharp_and_Java

Se realiza una comparativa bastante amplia entre las dos plataformas, concretamente entre los lenguajes C# y Java.

Pero al margen de todo lo anterior, y atendiendo al criterio que decanta la utilización de C# frente a Java, es la experiencia del autor con el primero siendo mucho mayor que con el segundo, por lo que dado que la importancia de la investigación se centra en obtener un modelo de representación lo suficientemente bien ajustado para obtener unos clasificadores competitivos con el estado del arte, el programa para ello es una mera herramienta y por tanto el lenguaje pasa a un segundo plano de necesidad, pudiéndose como bien se comenta al principio, ínter operar entre ambos e incluso, si en un futuro se requiriese, traducir mediante el análisis y diseño formalizado en los anexos finales.

3.1.8 Herramientas de minería de datos

Una cuestión importante a la hora de implementar un sistema como el propuesto radica en elegir entre realizar una implementación de los algoritmos conocidos (o alguna variante propia resultado de un proceso de investigación), o bien en utilizar algún tipo de paquete o librería externa o de terceros.

Dadas las características del problema se desea tener una herramienta integrada que clasifique las nuevas páginas dadas a partir de modelos aprendidos, por lo que no se tiene como factible la opción de utilizar una herramienta externa.

La opción de implementar un algoritmo resulta fuera del alcance de los objetivos y constituye en sí misma un proyecto completo y muy complejo.

Por tanto la opción que queda es la de utilizar algún tipo de librería, comercial o a ser posible de libre distribución, desde la propia aplicación a desarrollar. De entre todas las existentes se puede considera apropiado la utilización de las librerías de la Universidad de Waikato con su programa Weka, que contiene una de las mayores colecciones de librerías diferentes para todo tipo de tareas de minería de datos, totalmente programadas en lenguaje manejado, en este caso Java, y de licencia GNU, así como por los conocimientos del autor sobre esta herramienta y la amplia bibliografía disponible.

Puesto que el desarrollo de la aplicación se ha pensado realizar en C# bajo la plataforma .Net se necesita algún modo de hacer ínter operar ambos lenguajes de programación. Existe una herramienta denominada [IKVM] que permite la conversión de los paquetes Java a librerías .Net para su utilización desde dicha plataforma, manteniendo la interfaz original descrito en Java, así como manteniendo manejado el código, es decir, consiguiendo una integración total del mismo.

Es por tanto la decisión tomada para la implementación de la solución. Una importante característica que debería cumplir el sistema es la de su fácil distribución. En el caso del .Net sólo es necesario tener el framework del mismo, en su versión 2.0 que es en la que se ha realizado la aplicación, descargable desde cualquier web de Microsoft (descarga directa adjunta en las referencias), y que funciona bajo sistemas Microsoft, y últimamente por lo que el autor ha podido leer, bajo plataformas Linux, aunque no lo he probado personalmente. Así pues, para aumentar la capacidad de distribución se debería componer de librerías con una API definida para realizar las diferentes tareas, separándolas claramente de la presentación y permitiendo de este modo ser utilizadas desde otras aplicaciones .Net, o bien Java mediante acceso vía Web Service o alguna otra herramienta del estilo IKVM, pero en la otra dirección. Esta podría ser una línea de mejora, aunque fuera del alcance del proyecto y fuera del interés de la investigación.

3.2 LA COLECCIÓN DE PRUEBA

La colección de pruebas es uno de los elementos determinantes a la hora de obtener buenos clasificadores porque de ella dependerá tener o no un conjunto suficientemente significativo como para generalizar unos modelos capaces de predecir cualquier nueva entrada a los mismos, y con ello, clasificar nuevos documentos web.

3.2.1 La colección de prueba general DS y DSA

Hemos generado un repositorio de pruebas inicial compuesto con un total de 4801 páginas web correspondientes a 16 categorías predefinidas. Puesto que algunas de estas categorías tienen una representatividad prácticamente nula se han eliminado y han pasado a formar parte de un resto con un 0,35% (17 páginas en total) del total.

La distribución de las páginas por categorías es la siguiente:

Clase	Total	%
Asociaciones	26	0,54%
Blogs	553	11,52%
Compañías	2611	54,38%
Festivales	747	15,56%
Formación	290	5,42%
Revistas	75	1,56%
Salas	300	6,25%
Textos	182	3,79%
Resto categorías	17	0,35%
TOTAL	4801	100%

FIGURA 3.9: Colección de pruebas DS

Como se puede observar, su distribución es muy desigual, desde un 54,38% de la categoría mayoritaria, *Compañías,* hasta un 0,54% de la categoría minoritaria, *Asociaciones*. Esta desigualdad nos servirá para experimentar sobre la necesidad de grandes cantidades de datos, y determinar si esto es necesario o si por el contrario, si los mismos son suficientemente representativos, son suficientes por sí solos para obtener buenas generalizaciones.

Para obtener el repositorio de páginas se ha partido de un conjunto de sitios web previamente etiquetados de la web encuentrateatro.com

El fichero de anotaciones utilizado es un fichero xml estructurado del modo que se indica a continuación y que contiene el conjunto de urls previamente catalogadas, junto con las clases a las que pertenece cada Url.

Así mismo, las Urls pueden tener un símbolo * tras ellas, lo que indica que todas las páginas que comienzan por el mismo prefijo anterior al * pertenecen a esa misma categoría. Esta característica es la que nos permite, mediante un crawl hasta el tercer nivel, expandir el repositorio de páginas para obtener la colección de páginas indicada al principio y que será utilizada para realizar el entrenamiento. Esto es válido por ejemplo para todas las páginas de tipo Blog que partan de la raíz del sitio, como los diferentes archivos, o todas las páginas dinámicas que se creen en tiempo de ejecución variando un parámetro, o casos similares. Todas ellas, por separado, pertenecen a esa misma categoría por lo que son representativas de la misma y sensibles de ser utilizadas para el aprendizaje.

El formato del fichero es el siguiente:

```
<Annotations num="xxx" total="xxx">

<Annotation about="http://www.mi-dominio.xxx/file.ext">
    <Label name="class1" />
    <Label name="class2" />
  </Annotation>

...
```

o bien

```
<Annotation about="http://www.mi-dominio.xxx/path/*">
    <Label name="class5" />
    <Label name="class1" />
  </Annotation>
...

</Annotations>
```

La primera de las anotaciones indica una url única que pertenece a las clases 1 y 2. Podría pertenecer a más clases o a una única clase, con lo que contendrá tantas etiquetas label como necesite.

La segunda de las anotaciones indica una url a partir de la cual todas las Urls que compartan ese fragmento de url, accedidas desde ella o desde cualquier otra url, pertenecerán a las clases indicadas en las etiquetas label.

Con ello se obtienen las páginas web que pertenecen a las diferentes categorías y se construye un repositorio local en formato xml de fácil acceso.

El fichero xml del repositorio incluye la url extraída, junto con la clase a la que pertenece y el html de la web a la que apunta.

El formato del xml del repositorio es el siguiente:

```
<Repos>
  <URLS>
      <URL></URL>
      <CLASS></CLASS>
      <HTML>/<HTML>
  </URLS>
</Repos>
```

Si una url estaba anotada para más de una categoría, aquí habrá tantas etiquetas URLS como categorías hubiera, cambiando el valor de la etiqueta CLASS y manteniendo constante el valor de la etiqueta URL y HTML.

El repositorio completo tiene un tamaño total en disco de aproximadamente 73Mb, conteniendo las 4801 Webs indicadas al principio.

El conjunto de Urls anotadas que da lugar al repositorio descrito consta de 167 Urls anotadas con la distribución que se indica a continuación:

Clase	Total	%
Asociaciones	14	8,97%
Blogs	6	3,85%
Compañías	57	36,54%
Festivales	12	7,69%
Formación	22	14,10%
Revistas	7	4,49%
Salas	29	18,59%

Textos	3	1,92%
Resto categorías	6	3,85%
TOTAL	156	100%

FIGURA 3.10: Colección de pruebas DSA

Como se puede observar algunas de las anotaciones contienen muy pocos sitios Webs que dan lugar a muchas páginas diferentes, como los *Blogs* (de 6 a 553) o los *textos* (de 3 a 182) Habrá que comprobar que realmente esto no afecta a la calidad de la evaluación de los modelos, y para ello se ha realizado una serie de experimentos como se verá en los siguientes apartados.

3.2.2 La división de la colección de pruebas: DS1 y DS2

Además de este repositorio inicial, en algunos de los experimentos, como se podrá comprobar, se optará por utilizar un conjunto de test diferente al utilizado en el entrenamiento de manera que no se mezclen en ambos conjuntos o repositorios páginas de un mismo sitio web, de manera que se eliminen las posibles implicaciones o relaciones que pudieran existir por ello.

Para ello, se obtienen dos repositorios separados a partir del repositorio inicialmente propuesto, de manera que uno de ellos podrá ser utilizado para entrenar y el otro para probar, indistintamente, obteniendo de este modo unos datos más reales.

La separación se realiza de la siguiente manera:

Distribución de Urls

Clase	Sitios / Webs	DS1	Ratio	DS2	Ratio
Asociaciones	14 / 26	8 / 18	57,14% / 69,23%	6 / 8	42,86% / 30,77%
Blogs	6 / 553	3 / 435	50,00% / 78,66%	3 / 116	50,00% / 20,98%
Compañías	57 / 2611	53 / 307	92,98% / 11,76%	4 / 2303	7,02% / 88,20%
Festivales	12 / 747	8 / 741	66,67% / 99,20%	4 / 6	33,33% / 0,80%
Formación	22 / 290	17 /181	77,27% / 62,41%	5 / 109	22,73% / 37,59%
Revistas	7 / 75	4 / 59	57,14% / 78,67%	3 / 16	42,86% / 21,33%
Salas	29 / 300	22 / 255	75,86% / 85,00%	7 / 45	24,14% / 15,00%
Textos	3 / 182	3 / 182	100%	0 / 0	0%
Resto	6 / 17	6 / 17	100%	0 / 0	0%
TOTAL	156 / 4801	118 / 2184	75,64% / 45,49%	32 / 2603	20,51% / 54,22%

FIGURA 3.11: Distribución de páginas en la colección de pruebas DS, DS1, DS2 y DSA

El criterio seguido para la división de los repositorios ha atendido preferentemente a mantener la separación de las páginas de los sitios web

completamente, así como a intentar representar el 75/25% de las páginas en uno y otro, para poder comprobar de este modo el efecto del entrenamiento para las diferentes clasificaciones cuando el conjunto de datos es diferencialmente grande o pequeño.

3.2.3 La colección de pruebas especial para la validación de los Blogs DSE

Por último, se ha creado un repositorio específico para testar el modelo de representación de los Blogs, obtenido a partir de un crawl hasta el nivel 3 de 42 sitios web clasificados como Blogs encontrados con el buscador de Blogs de Google, y que realmente son Blogs (Google incluye aquí todas las páginas con suscripción rss/atom), formando un conjunto total de 3.696 páginas Blog, así como un conjunto de 5.462 páginas que no son Blogs obtenidas de un crawl hasta el 5 nivel del directorio Yahoo España (http://es.yahoo.com)

La distribución de las Urls corresponde a la siguiente tabla:

Clase	Sitios	Total	%
No Blogs	1	5.462	59,64%
Blogs	42	3.696	40,36%
TOTAL	43	9158	100%

FIGURA 3.12: Distribución de páginas en la colección de pruebas DSE

La distribución por idiomas es la siguiente:

Idioma	Sitios	Total	%
Castellano	39	3566	96,48%
Alemán	1	26	0,70%
Inglés	1	103	2,89%
Francés	1	1	0,02%

FIGURA 3.13: Distribución de idiomas en las páginas de la colección de pruebas DSE

Lo que permitirá comprobar en cierta medida el grado de independencia de la representación frente al idioma.

3.2.4 Pretratamiento de las Webs

Para asegurar una correcta adecuación de las páginas a sus respectivas categorías se han realizado una serie de comprobaciones manuales sobre las mismas y se han eliminado aquellas que no fueran representativas de la categoría que se le asigna.

Para ello, básicamente se ha realizado lo siguiente:

- Se ha eliminado del conjunto de anotaciones las páginas que no responden, que dan error de no-existencia o de redirecció, o que realizan una redirección Javascript o similar y su contenido es algún indicativo de error, de manera que su descarga no corresponda a un contenido adecuado a la categoría que se le pretende asignar.
- Se ha eliminado del conjunto de anotaciones las páginas que diciendo ser de una categoría no pertenecen a la misma.

- Se ha expandido el conjunto de Urls anotadas y pretratadas y se ha eliminado del resultado el conjunto de páginas que no responden, dan un error de no existencia o de redirección, o que realizan una redirección Javascript o similar y su contenido es algún indicativo de error, de manera que su descarga no corresponda a un contenido adecuado a la categoría que se le pretende asignar.
- Se ha eliminado aquéllas páginas que claramente no pertenecen a una categoría dada por ser páginas demasiado genéricas como las de contacto, login o similares, sobre todo de la categoría Blogs, pues con una revisión de la url de la misma es suficiente.

El resultado del preprocesado anterior conforman los cinco repositorios utilizados en los experimentos, el repositorio de anotaciones DSA, el repositorio expandido DS, las dos divisiones del anterior DS1 y DS2 y el repositorio específico para testar la representación de los Blogs DSE.

3.2.5 Comentarios al repositorio DS y sus divisiones DS1 y DS2

Debido al elevado número de características y su diferente proporción de páginas es necesario realizar unos comentarios generales para una mejor interpretación de los resultados.

Algunas categorías, como las *Asociaciones* por ejemplo, tienen una representatividad muy baja, menos de un 1% de las páginas son de este tipo. Así mismo, otras categorías como las *Compañías,* tienen una división muy dispar entre el DS1 (11,76%) y el DS2 (88,24%)

Por otra parte la categoría *Textos* no tiene representación en DS2, por lo que únicamente se puede realizar una validación cruzada con todo el conjunto de datos, y no las realizadas para el resto de categorías DS1/DS2, DS2/DS1 y 2x2.

Todas estas situaciones son en cierto modo especiales y pueden condicionar los datos obtenidos en la evaluación de los modelos, y por tanto hay que tenerlas en cuenta.

Un resumen de los riesgos o situaciones especiales que se pueden dar son los siguientes:

- Riesgo por número reducido de Webs frente al total: Las categorías de este tipo pueden obtener unos errores muy dispares entre la clasificación positiva y negativa a la categoría, y pese a ello obtener un error total muy alto o muy bajo, ya que la clase mayoritaría tendrá más peso. Es especialmente importante tener esto en cuenta y no tomar estas clases como una base sólida para la comparación de representaciones, y sí únicamente para complementar las conclusiones extraídas con otras categorías más sólidas. Las clases de este tipo son:
    - Asociaciones, con un 0,54% de representatividad
    - Revistas, con un 1,56% de representatividad
    - Textos, con un 3,79% de representatividad
    - Formación, con un 5,42% de representatividad
    - Salas alternativas, con un 6,25% de representatividad

Aunque como se verá en los resultados, estas dos últimas categorías son suficientemente representativas para obtener valores comparables, aunque no modelos competitivos, y por tanto, junto con textos, se podrán utilizar para comparar la bondad de los clasificadores en las diferentes representaciones.

- Riesgo por distribución demasiado dispar entre las divisiones DS1 y DS2: Una distribución demasiado dispar podrá marcar una diferencia significativa entre la validación realizada por ambos repositorios, aunque la media 2x2 puede ser representativa, habrá que tener en cuenta esta situación. Algunas Webs de este tipo son:
    - Festivales, con un 99,20% frente a un 0,80%
    - Compañías, con un 11,76% frente a un 88,24%
    - Salas, con un 85% frente a un 15%
- Riesgo por distribución irregular de sitios: En este caso, aunque la proporción entre Webs sea adecuada y ajustada al 75/25 pretendido, no así sucede con los sitios web que dan lugar a todas esas Webs, lo que puede provocar que el aprendizaje esté muy ajustado al estilo propio de dichos sitios Webs. La única categoría en esta situación es:
    - Compañías: con un 92,98% frente a un 7,02% de sitios.
- La última anotación es indicar que en el caso de *Compañías*, es el repositorio DS2 el que tiene el mayor número de elementos de la misma, de manera que el número de datos totales utilizados para entrenar y validar fueran similares, de modo que la media 2x2 no dependiera en mayor medida de un repositorio que del otro, dándose por tanto la misma prioridad a un aprendizaje con muchos datos de entrenamiento que a uno más limitado.

3.3 EFECTOS DE LA EXPANSIÓN DE LAS URLS SOBRE LA CALIDAD DE LOS MODELOS

La necesidad de tener un conjunto inicial suficientemente representativo de todas las categorías para aprender modelos generalizadores de las mismas es algo que está en la mente de todos.

El presente experimento consiste en comparar los resultados de la clasificación a partir de las páginas que componen el fichero de anotaciones con los obtenidos por las páginas expandidas mediante un crawl hasta el nivel tres de las mismas, y determinar si es necesaria dicha expansión para la consecución de los modelos.

3.3.1 Definición del experimento

El experimento consiste en realizar una comparativa entre la clasificación web a partir de las páginas dadas en las anotaciones, sin expandir, con la realizada tras la expansión de las mismas, de modo que se compruebe si dicha expansión tiene repercusiones en la calidad de los modelos obtenidos.

- La hipótesis de partida es que un conjunto de entrenamiento demasiado pequeño tiene repercusiones en la calidad de los modelos, que no son capaces de obtener generalizaciones suficientemente buenas, por lo que se hace necesario el uso de técnicas como el crawl de los sitios web para obtener un repositorio significativamente mayor

3.3.2 Realización del experimento y resultados

Para la realización del experimento se ha obtenido una representación BoW estándar a partir de las palabras contenidas en el corpus de todos los documentos de ambos repositorios (el de anotaciones DSA y el expandido DS)

Para obtener la representación BoW se obtiene primero los corpus de todas las páginas que componen ambos repositorios y para ello se realiza una extracción de palabras mediante la expresión regular siguiente:

```
\b(?<word>[a-zñáàéèíóòúëï]+)\b
```

Mediante la anterior expresión, se obtiene el conjunto de caracteres no numéricos agrupados y separados entre sí mediante un separador alfanumérico corriente.

A este conjunto de palabras se le aplica un proceso de stemming de Porter adaptado al castellano, de manera que se reducen a su forma truncada común, reduciéndose de este modo la dimensionalidad y agrupando palabras con un mismo significado, aumentando la representatividad de las mismas.

Tras esto se pasa por un filtro de palabras vacías eliminando aquellas que lo sean, reduciendo aún más la dimensionalidad y eliminando características que no aportan información alguna.

Tras la creación del corpus de la manera anteriormente descrita, se procede a la representación de los ejemplos etiquetados (páginas web anotadas) en ambos repositorios por separado, mediante un modelo de vector de palabras.

Para ello se obtiene una vez más el conjunto de palabras que forman cada uno de los ejemplos, mediante la expresión regular anterior, y se compara con la existencia de la misma en el corpus correspondiente.

Cada aparición se anota con un incremento unitario sobre el valor de la característica correspondiente. Finalmente, se procede a la normalización dividiendo todas las apariciones por el número de características (palabras) del ejemplo en cuestión.

Con lo anterior se ha obtenido un vector de palabras dónde cada dimensión se corresponde con una palabra del corpus previamente creado, y cada valor se corresponde con la frecuencia local de aparición de la palabra correspondiente del corpus en el documento en cuestión.

Con el conjunto de ejemplos obtenido se ha entrenado un clasificador, único para todas las categorías, de tipo bayesiano, concretamente Naïve Bayes, y se ha evaluado, mediante el método descrito en el apartado 3.1.6 basado en la precisión, por validación cruzada con un valor de *fold* igual a 10, lo que significa que se ha repetido 10 veces el entrenamiento/validación utilizando para ello un conjunto cada vez diferente de parejas de 9 frente a 1 elementos.

El resultado utilizado para comparar la evaluación de ambos repositorios es el del valor de F, calculado como se describe en el punto 3.1.6, y que es el más representativo cuando se pretende dar igual importancia a la precisión y al alcance.

Los resultados obtenidos para ambos conjuntos de datos se expresan conjuntamente en la siguiente tabla (los resultados completos se adjuntan en el Anexo IV apartados 1 y 2):

	Sin expandir	**Expandida**
Asociaciones	0,000	0,135
Blogs	0,211	0,776
Compañías	0,136	0,900
Festivales	0,250	0,745
Formación	0,000	0,736
Revistas	0,000	0,296
Salas Alternativas	0,000	0,659
Textos	0,286	0,936

FIGURA 3.14: Estadístico F BoW std para las colecciones de pruebas DSA y DS

Como se puede observar, muchas categorías como Asociaciones, Formación, Revistas o Salas Alternativas tienen en el repositorio sin expandir un valor de F igual a 0, lo que indica que prácticamente nunca consigue una clasificación válida para estos tipos.

Así mismo, en el resto de categorías los valores apreciados son muy bajos, siendo el mayor de ellos algo inferior al 0,30.

En cambio, todos los valores obtenidos para el repositorio expandido muestran valores muy superiores al anterior, llegando en alguno de los casos a obtenerse tasas cercanas al 1.

Se obtiene a partir del número de aciertos en ambos clasificadores (concretamente 9 de 97 en el primero y 3604 de 4663 en el segundo) el error real esperado a partir del error de muestra, con un grado de certeza del 95%,

Respecto al error de clasificación obtenido por ambos modelos, con un intervalo de confianza del 95% calculado como 3.1.6, se tienen los siguientes valores:

Sin expandir: ErrorR(S) = 0,907 +- 0,058
Expandido: ErrorR(S) = 0,227 +- 0,012

Que como se pueden apreciar son muy diferentes, pasando desde un 90,7% del caso de las anotaciones a un 22,7% en el caso de las Urls expandidas.

3.3.3 Conclusiones

El experimento nos ha mostrado la necesidad de trabajar con un conjunto o repositorio inicial de páginas lo suficientemente representativo.

Para ello se ha realizado el crawl hasta el tercer nivel de manera que se obtiene un conjunto superior de datos y que, tal y como se ha visto en la realización del experimento, consigue obtener mejores resultados en la clasificación y por tanto servir de mejor base para una generalización a partir de los mismos.

Atendiendo al error cometido se pasa de un 90,7% a un 22,7%, un descenso significativo, teniendo en cuenta además, que al 95% de confianza, el intervalo se amplía en un 5,8% para primer caso, descendiendo a un 1,2% en el segundo, lo que lo hace más acotado en torno a la media.

Se concluye que se corrobora la hipótesis de que un conjunto demasiado pequeño de datos no es lo suficientemente representativo para el problema de la clasificación, y con ello la necesidad de utilizar un repositorio mayor, nacido en este caso a partir de un crawl del anterior, de manera que los resultados obtenidos alcancen cierto grado de calidad.

3.4 EFECTOS DE LA EXPANSIÓN DE LAS URLS SOBRE EL MÉTODO DE VALIDACIÓN CRUZADA

Ahora bien, obtener un repositorio por el método de obtener todas las páginas de un determinado sitio web puede tener consecuencias no apreciadas a simple vista por cuestiones de relación entre ellas, tales como un mismo estilo o un vocabulario común.

Páginas de un mismo autor seguramente compartirán características comunes, tales como el mismo o similar vocabulario, la misma estructuración de las páginas e incluso el uso similar de etiquetas de enmarcación html.

Es por ello necesario comprobar la validez de los resultados obtenidos mediante una validación cruzada, y que sirva de base empírica para las afirmaciones realizadas en la evaluación del resto de experimentos.

En la validación cruzada, los diferentes subconjuntos de prueba son independientes entre sí, no así los conjuntos de entrenamiento que comparten un alto número de ejemplos comunes. Algunos autores como [Dietterich 1998] defienden que este solapamiento de los conjuntos podría afectar a la calidad de la estimación, sobre ajustándose a los ejemplos y dando una estimación demasiado elevada frente a la real. Para solucionarlo proponen un método de validación cruzada denominado 5x2, dónde se realizan cinco repeticiones de validación cruzada con un valor de fold igual a 2, de manera que los subconjuntos de entrenamiento y test son totalmente disjuntos en cada iteración.

3.4.1 Definición del experimento

El experimento consiste en realizar una comparativa entre los resultados obtenidos por validación cruzada en el repositorio de Webs expandido con los resultados obtenidos a partir del uso de repositorios separados para el entrenamiento y la validación.

- La hipótesis de partida es que existe cierta relación entre las páginas, por lo que no se pueden considerar como independientes para la generación y prueba de los modelos, y por tanto no es conveniente el uso de la validación cruzada pues sus resultados pueden estar sobre ajustados y no ser representativos del aprendizaje

Para ello se hará uso de los tres repositorios creados y explicados en el punto 3.1, el expandido, la primera partición del mismo (DS1) y la segunda partición del mismo (DS2), realizando un entrenamiento con DS1 y validando con DS2 y viceversa, así como obteniendo el valor medio en lo que sería una variación del método propuesto por [Dietteritch] pero con dos iteraciones (2x2) y una separación total en ambos repositorios no por páginas sino por sitios web.

La utilización de todo el repositorio para entrenamiento y validación cruzada, en caso de considerar que la hipótesis es falsa, debería obtener unos valores similares del estadístico F, así como del intervalo de error cometido, respecto a la utilización de parte del repositorio para evaluar otra parte, en este caso el uso de DS1 para evaluar DS2 y viceversa.

Por tanto la prueba a realizar será la de determinar con qué grado de certeza las dos evaluaciones (*cross validation y 2x2*) son equivalentes y rechazar por tanto la hipótesis nula, o si por el contrario no lo son y se acepta, afirmando que existe relación entre los contenidos de las páginas y por tanto el método de validación cruzada no es fiable respecto a la precisión obtenida para los modelos, de manera que será precisa la utilización del método de evaluación por previa partición del repositorio para obtener datos fiables y representativos de la realidad.

3.4.2 Realización del experimento y resultados

Para la realización del experimento se obtiene una representación BoW de los tres repositorios de datos, de igual manera que se explicó en el punto 3.2.2

El entrenamiento y evaluación se realiza sobre un único modelo de clasificación, Naïve Bayes, para todas las categorías.

La comparativa se realiza mediante el valor del estadístico F obtenido para cada una de las características en cada una de las evaluaciones, así como para el intervalo de error obtenido para cada una de ellas.

El estadístico F se obtendrá de la manera en que se definió en el apartado 3.1.6, ahora bien, para el método de validación cruzada se utilizará el repositorio general tanto para entrenamiento como para validación, mediante un valor de *fold* igual a 10, y el cálculo para los repositorios DS1 y DS2 se realizará mediante el entrenamiento con una de las dos particiones y la validación con la contraria. En el método 2x2 se realizará la evaluación media de las dos anteriores. Será con esta última con la que se comparará la validación cruzada, determinando su similitud mediante un test t-student pareado de dos colas.

De este modo los resultados obtenidos son los mostrados en la siguiente tabla:

	Validación cruzada	**DS1/DS2**	**DS2/DS1**	**2x2**
Asociaciones	0,135	0	0	0
Blogs	0,776	0,058	0,389	0,303
Compañías	0,900	0,037	0,123	0,048

Festivales	0,745	0	0	0
Formación	0,736	0,061	0,124	0,107
Revistas	0,296	0,011	0,079	0,031
Salas Alternativas	0,659	0,233	0,227	0,228
Textos	0,936	0	0	0

FIGURA 3.15: Estadístico F validación cruzada vs. 2x2

No hace falta realizar ningún test estadístico para determinar que los valores obtenidos por los diversos métodos son significativamente diferentes. En cualquier caso se compararán aquellos resultados para las categorías no comentadas como de riesgo en el punto 3.1.5:

	Validación cruzada	**DS1/DS2**	**DS2/DS1**	**2x2**
Blogs	0,776	0,058	0,389	0,303
Compañías	0,900	0,037	0,123	0,048
Formación	0,736	0,061	0,124	0,107
Salas Alternativas	0,659	0,233	0,227	0,228

FIGURA 3.16: Estadístico F validación cruzada vs. 2x2

En cualquier caso sigue sin ser necesaria la realización del estadístico t-student pues los datos son significativamente diferentes para todas las categorías a simple vista, tanto en las validaciones DS1/DS2 y DS2/DS1 como en su combinación 2x2.

Los diferentes intervalos de confianza al 95% para el error cometido por los diferentes clasificadores se puede ver a continuación:

Validación cruzada: 0,227 +- 0,012

DS1/DS2: 0,997 +- 0,002
DS2/DS1: 0,818 +- 0,017

2x2: 0,917 +- 0,008

Lo que claramente muestra esta diferencia, siendo significativamente inferior (22,7%) en el caso de la validación cruzada frente a la validación 2x2 (91,7%)

3.4.3 Conclusiones

El experimento anterior ha mostrado que los resultados obtenidos por validación cruzada difieren significativamente de los obtenidos por validación mediante repositorios separados, tanto analizando sus valores para la prueba F como para el intervalo de error cometido.

Métodos como el de [Dietterich 1998] consistente en realizar 5 particiones sucesivas de los datos en dos conjuntos disjuntos y utilizarlos para entrenamiento y validación, o métodos como [Bouckaert] consistente en obtener un conjunto de N particiones de M bloques de datos y realizar sucesivas validaciones cruzadas sobre los

mismos, nos permiten obtener una validación de los modelos más ajustada a la realidad que los métodos anteriores de validación cruzada o el 2x2 propuesto.

Pero dado que el principal objetivo perseguido con la evaluación de los modelos no es tanto determinar la validez real de los mismos, para lo cual se haría precisa una evaluación como las anteriores, como permitir comparar su bondad para las diferentes representaciones, el método de validación 2x2 nos permitirá realizarlo de manera global para el método, así como las dos validaciones que dan lugar a ello nos permitirán comparar la bondad de las representaciones en un aprendizaje normal, y en otro forzado a un conjunto de entrenamiento limitado, y en todos los casos más ajustada a la realidad y evitando el sobre ajuste a los datos que en el método de la validación cruzada.

De este modo definimos el repositorio y validamos su idoneidad para la realización del estudio empírico sobre el mismo, así como definimos el marco de evaluación necesario para la evaluación comparativa de los diferentes modelos.

3.5 EFECTOS DEL TRATAMIENTO LINGÜÍSTICO: EL STEMMING

Respecto al proceso de tratamiento lingüístico ha quedado bastante claro en diversos artículos [Arregi] que es necesario no sólo para reducir la dimensionalidad del problema, sino además para obtener mejores resultados, reuniendo palabras con una misma raíz que de otro modo aparecerían por separado, dividiendo así su importancia.

El tratamiento lingüístico es algo complejo y lejos del alcance de la investigación por lo que se ha acudido al estado del arte para obtener algún tipo de aplicativo que lo contemple.

Una manera de relajar este análisis lingüístico es mediante los denominados algoritmos de stem, que mediante una serie de reglas computacionales reducen las palabras a sus lemas comunes mediante eliminación de prefijos, sufijos y características similares.

En la literatura es común oír hablar del stemmer de Porter, que es un proceso para eliminar las finalizaciones morfológicas e inflexivas más comunes de las palabras en idioma inglés. Su principal uso ha sido para la normalización de términos en las tareas de recuperación de información.

La implementación del algoritmo de [Porter 1980] está realizada en diversos lenguajes de programación. Así mismo aparecen variaciones para contemplar otras lenguas diferentes del inglés, como en nuestro caso el castellano.

En un repositorio sobre recursos lingüísticos relacionados con esta área [Snowball] se encuentra una serie de ficheros, por idiomas, dónde se relacionan par a par las palabras del idioma con su raíz previamente *truncada,* por lo que se decide utilizar este fichero, cargándolo como una tabla hash de acceso rápido y directo, en lugar de implementar ningún algoritmo.

3.5.1 Definición del experimento

El experimento consiste en comprobar la necesidad de la aplicación de un algoritmo de este tipo tanto para reducir la dimensionalidad como para obtener representaciones más adecuadas y por tanto obtener mejores tasas de aprendizaje.

- La primera hipótesis es que la aplicación del algoritmo de stem de Porter reduce significativamente la dimensionalidad, haciendo más manejable el conjunto de datos de entrenamiento y por tanto creando un modelo más ligero
- La segunda hipótesis es que la aplicación del algoritmo de stem, al juntar palabras de un mismo significado, consigue una representatividad mayor del documento y por tanto alcanza una precisión mayor en la evaluación de los modelos

Para corroborar ambas hipótesis se obtiene el corpus correspondiente al conjunto de pruebas con y sin realizar procesamiento de truncado, y se obtiene la representación BoW a partir de dichos corpus.

La primera hipótesis se validará mediante el tamaño en palabras del corpus obtenido, que resultará en la dimensionalidad total del conjunto de entrenamiento.

La segunda hipótesis se validará mediante la evaluación cruzada del modelo aprendido con un clasificador único bayesiano, obteniendo mediante *fold* igual a 10 la media de valores de cada clasificación y con ella el estadístico F que se compararán en ambos casos mediante un test de student para determinar su similitud. Aunque no será el método de evaluación por los motivos descritos en el experimento 3.3, para la evaluación comparativa en este punto es suficiente.

Por lo tanto, la segunda hipótesis se reescribe como una hipótesis nula dónde se dice que:

- Las series de valores de precisión obtenidos para cada representación mediante 10 validaciones tienen igual media y por tanto ambas son iguales.

El test de t-student para una significación del 95% nos indicará si dicha hipótesis nula es correcta, con lo que se rechaza la hipótesis alternativa que es la segunda hipótesis formulada, o si por el contrario ambas series no son de media igual y por tanto se rechaza la hipótesis nula y con ello se afirma la segunda hipótesis del experimento, y el proceso de stem sí que tiene implicación en la calidad del modelo resultante.

3.5.2 Realización del experimento y resultados

Para la realización del experimento se obtiene el corpus total de las páginas aplicando en uno de ellos el proceso de stem y no en el otro.

El tamaño total de los corpus en palabras queda reflejado en la siguiente tabla:

	Sin stem	Con stem
Número de palabras/características	51.292	7.019

Como se puede apreciar a simple vista el corpus obtenido sin proceso de stem es algo más de siete veces mayor que el obtenido mediante la aplicación del proceso de stem.

Lo anterior resulta en una representación BoW, sin realizar pretratado alguno de las características, de un tamaño muy superior al obtenido mediante stem, tal y como se refleja en la siguiente tabla:

	Sin stem	Con stem
Número de valores	246.252.892	33.698.219

FIGURA 3.17: Número de palabras corpus con y sin stem

Como se puede observar, el número de valores obtenidos es muy elevado..

Tras la obtención de los corpus se procede a la creación de la representación BoW que servirá para la construcción de los modelos y su evaluación, y que será la base para la realización de la comparación y por tanto de la aceptación o el rechazo de la segunda hipótesis.

Como se puede apreciar en la tabla anterior el número de características y valores obtenidos para la opción sin truncamiento es demasiado elevada y su aplicación a la construcción del modelo tiene un coste computacional, temporal y espacial, fuera de las prestaciones disponibles, con lo que se debe proceder a realizar una limpieza o preprocesado del mismo.

Para ser imparcial en la selección de atributos y no mejorar la capacidad clasificadora de los mismos por ella, se eliminan aquellos que menos capacidad discriminante tengan, que son los que menos veces aparecen en los documentos.

Concretamente se eliminan todas aquellas palabras que aparecen menos de 30 veces en el corpus, quedando un corpus final de tamaño equivalente al obtenido mediante truncado.

Tras la obtención y validación cruzada de los modelos para las representaciones anteriores se obtienen los resultados de la prueba F tabulados a continuación:

	Sin stem	Con Stem
Asociaciones	0,114	0,135
Blogs	0,827	0,776
Compañías	0,942	0,900
Festivales	0,814	0,745
Formación	0,798	0,736
Revistas	0,397	0,296
Salas Alternativas	0,774	0,659
Textos	0,930	0,936

FIGURA 3.18: Prueba F para corpus con y sin stem

Las series a simple vista son muy similares, por lo que es preciso la aplicación de un test de student pareado, ya que se comparan par a par los resultados obtenidos para cada categoría.

Para un nivel de significación de 0,05 y 7 grados de libertad que tiene el problema (hay 8 datos con media y desviación estándar dada), se tiene que el test de t-student debe ser superior a 2,365 (porque se realiza un test de dos colas para comprobar que las medias son iguales)

Las series a comparar son:

Con stem X = {0,114; 0,827; 0,942; 0,814; 0,798; 0,397; 0,774; 0,930}
Sin stem Y = {0,135; 0,776; 0,900; 0,745; 0,736; 0,296; 0,659; 0,936}

Y la serie diferencia de las anteriores D = |X-Y| es la siguiente y será sobre la que se aplicará el test de t-student:

D = {0,021; 0,051; 0,042; 0,069; 0,062; 0,101; 0,115; 0,006}

Los estadísticos básicos obtenidos para las tres series son los siguientes:

	Media	**Desviación estándar**
X	0.699	0,272
Y	0,648	0,266
D(iferencia)	0,058	0,035

FIGURA 3.19: Estadísticos de posición de las series de pruebas F para corpus con y sin stem

Con los cálculos anteriores realizados se aplica el test de t-student sobre los mismos y se compara con el valor crítico, aceptándose la hipótesis nula en caso de que el resultado del test sea inferior al valor crítico dado para ese nivel de significación, y rechazándose en caso contrario.

Dados:

$$\hat{X}_i = (X_i - \overline{X})$$
$$\hat{Y}_i = (Y_i - \overline{Y})$$

Se define el test de t-student como:

$$t = (\overline{X} - \overline{Y}) \sqrt{\frac{n(n-1)}{\sum_{i=1}^{n} (\hat{X}_i - \hat{Y}_i)^2}},$$

O de manera alternativa, se puede calcular su valor cómo:

$$z = \frac{\overline{x} - \mu}{\hat{\sigma}/\sqrt{n}} = \frac{\overline{x} - \mu}{s\sqrt{n}}$$

donde $\overline{x}$ es la media de la distribución de valores diferencia D (en valor absoluto), μ es igual a cero, $\hat{\sigma}$ es la desviación típica de la distribución D, y n es igual a 8 (grados de libertad + 1)

Se calcula el resultado siendo igual a 3,098 > 2,365, lo que demuestra la diferencia de las medias de ambos métodos a un nivel de significación del 95%.

Calculando por otro lado el intervalo de error a una confianza del 95% se obtiene los siguientes valores:

Sin stem: 0,162 +- 0,011
Con stem: 0,227 +- 0,012

Que aunque similares, muestran también una diferencia de al menos un 5% no cubierta por el margen del intervalo, con lo que demuestran la superioridad del método sin stem.

3.5.3 Conclusiones

Las conclusiones se desprenden directamente de los resultados, la primera hipótesis se corrobora, el tamaño del corpus, y por tanto de la representación BoW, a partir de un proceso sin stem es significativamente superior, llegando a ser intratable en la práctica, por lo que se requiere dicho proceso para reducir la dimensionalidad.

Respecto a la segunda hipótesis, y en contra de estudios como [Arregi] dónde se afirma la importancia de este proceso para una mejor obtención de los modelos, vistos los resultados del experimento dónde el test de t-student nos muestra que ambas medias no son iguales, nos hace rechazar la hipótesis de que el proceso de stem mejora los resultados de la clasificación, pues sin stem los resultados son, al 95% de significación, superiores sin el proceso de stem.

Las posibles causas de lo anterior puede que estén directamente relacionadas con el modelo utilizado, el modelo bayesiano Naïve Bayes, que sea capaz de generalizar con la misma precisión aunque las características estén más dispersas, o bien puede estar relacionado directamente con el corpus de teatro, donde las características más significativas y diferenciadoras entre categorías no sufran mucho el efecto de las variaciones lingüísticas.

En cualquier caso queda demostrada la necesidad de la aplicación del stem para reducir la dimensionalidad, no así pues para conseguir modelos mejores. Su utilización parece que implica una reducción de la calidad de los mismos, pero aunque esto suceda, dado que el objetivo principal de la evaluación en esta investigación es la de comparar diferentes métodos, a los que afectará por igual por tratarse de métodos basados en bolsas de palabras, se utilizará el proceso de stem para la reducción de la dimensionalidad, y aunque en principio empeora ligeramente la calidad de los modelos,

para la evaluación comparativa entre modelos no será un impedimento y permitirá un mejor manejo de los mismos.

3.6 EFECTOS DEL PREPROCESAMIENTO: LA SELECCIÓN DE CORPUS

La clasificación de documentos, y en este caso concreto la clasificación Web, se basa en grandes cantidades de información caracterizadas por múltiples variables o características, en el caso de la BoW palabras, de las cuales no todas ellas son relevantes desde el punto de vista del problema a abordar.

Es por ello que uno de los pasos previos a todo tratamiento de minería de datos es la eliminación de información no relevante, para lo que una de las técnicas más estudiadas es la de selección de variables o características, cuyos principales objetivos son los siguientes [Orallo]:

- Reducir el tamaño de los datos, eliminando características o atributos irrelevantes o redundantes
- Mejorar la calidad del modelo, permitiendo al método de minería centrarse en las características relevantes
- Permitir expresar el modelo resultante en función de menos variables, de manera que sea más fácilmente comprensible

Existen dos tipos generales de métodos para la selección de atributos, los basados en filtro, que calculan mediante técnicas estadísticas medidas como distancia, ganancia de información, dependencia o inconsistencia, y los métodos basados en modelo, que utilizan el desempeño de un clasificador para determinar lo deseable de un subconjunto de características y las obtiene a partir de ello.

En cualquier caso es un proceso previo basado bien en técnicas estadísticas, bien en un entrenamiento previo del clasificador. En el apartado anterior se propuso una técnica para reducir el conjunto de atributos de la representación, mediante el equivalente al tratamiento lingüístico denominado proceso de stem o truncado, y en este apartado se tratará de la selección de un corpus significativo para la clasificación, con aquellas palabras (atributos) más característicos para cada clasificador en cuestión.

3.6.1 Definición del experimento

El experimento consiste en obtener un clasificador binario para cada categoría basado en la representación BoW dónde el corpus utilizado sea el formado únicamente por las palabras de dicha categoría.

El rendimiento de los clasificadores así obtenidos se comparará con el rendimiento de los clasificadores obtenidos a partir de la representación BoW con el conjunto de características común e igual al corpus general obtenido a partir de todas ellas.

En ambos casos los corpus se han obtenido aplicándoles el proceso de stem y de eliminación de palabras vacías descritos en el punto anterior, de manera que se reduce significativamente su tamaño.

Las hipótesis del experimento son dos:

- El conjunto de corpus para cada característica tendrá menos palabras con lo que los modelos serán construidos con un conjunto significativamente menor de características
- Al construirse los modelos de cada clasificador con las palabras específicas de su corpus el resultado será una serie de clasificadores con mejor rendimiento en cuanto a precisión y alcance que los obtenidos con el corpus general.

Para comprobar la primera hipótesis bastará contrastar el tamaño en número de palabras del corpus general con cada uno de los corpus para cada característica, comprobando que realmente el primero es muy superior a los segundos, resultando en una cantidad mucho mayor de datos y por tanto en la construcción de modelos más pesados.

Para comprobar la segunda hipótesis se tabularán los resultados del test F obtenidos para cada categoría por un lado para la pertenencia a la categoría, y por otro lado para la no-pertenencia, es decir, los dos posibles valores de cada clasificador pero por separado.

Así mismo se comparará el intervalo de error para cada uno de ellos, por pertenencia o no y por clasificador, comprobando si la diferencia es tal que corrobora la hipótesis.

En caso de no estar clara se aplicará un test t-student a la serie de resultados obtenidos por los clasificadores, par a par, para el caso de pertenencia y no-pertenencia, por separado, contrastando la hipótesis nula de que ambas series tienen la misma media y por tanto los clasificadores obtienen tasas similares, lo que rechazaría la segunda hipótesis del experimento.

3.6.2 Realización del experimento y resultados

Se obtiene el conjunto de los corpus de las diferentes categorías así como el general mediante la expresión regular de obtención de palabras descrita con anterioridad y aplicada al conjunto total de los documentos pertenecientes a una categoría, o del repositorio total en el caso del corpus común.

Las palabras repetidas se desechan de manera que se anotan únicamente palabras diferentes, obteniendo lo siguiente:

	CORPUS COMÚN	**CORPUS PROPIO**	**%**
Asociaciones	7019	522	7,4%
Blogs		4516	64,3%
Compañías		3091	44,0%
Festivales		5375	76,6%
Formación		3274	46,6%
Revistas		2557	36,4%
Salas Alternativas		2989	42,6%

Textos		2316	33,0%

FIGURA 3.20: Número de palabras corpus común vs. específicos

La categoría con mayor conjunto de atributos es la de *Festivales* que alcanza un conjunto de 5375 palabras, un 76,6% del tamaño del corpus general, de 7019 palabras.

La categoría con menor conjunto de atributos es la de *Asociaciones* con un conjunto de 522 palabras, un 7,4% del tamaño del corpus general.

En cualquier caso la mayoría de categorías están por debajo de la mitad del tamaño del corpus general.

Una vez comparado el tamaño de los diferentes corpus, se obtiene una representación BoW estándar para cada uno de ellos y se generan los diferentes clasificadores binarios de las diferentes categorías, evaluándose mediante validación 2x2 y obteniéndose los siguientes resultados, primero para la evaluación de pertenencia a la categoría:

PERTENENCIA A LA CATEGORÍA	CORPUS COMÚN	CORPUS PROPIO
Asociaciones	**0,013**	0
Blogs	**0,300**	0,299
Compañías	**0,708**	0,660
Festivales	0,084	0,084
Formación	0,155	**0,157**
Revistas	**0,048**	0,036
Salas Alternativas	0,185	0,185
Textos	**0,889**	0,814

FIGURA 3.21: Estadístico F en la clasificación de pertenencia con corpus común vs específico

En ella se puede apreciar que la mayoría de veces el corpus común obtiene tasas ligeramente superiores, aunque se necesitará un test de t-student para determinar si dicha mejora es significativa.

Respecto a la no-pertenencia se obtienen los siguientes datos:

No-PERTENENCIA A LA CATEGORÍA	CORPUS COMÚN	CORPUS PROPIO
Asociaciones	0,910	**0,958**
Blogs	0,706	**0,707**
Compañías	**0,727**	0,706
Festivales	0,760	0,760
Formación	0,756	**0,761**
Revistas	**0,969**	0,959
Salas Alternativas	0,794	**0,795**
Textos	**0,995**	0,991

FIGURA 3.22: Estadístico F en la clasificación de no-pertenencia con corpus común vs específico

En este caso la similitud es mucho mayor, estando más repartidos los resultados y muy igualados en cualquier caso.

En ambos casos los valores son muy similares por lo que es necesaria la aplicación del test t-student para comprobar la hipótesis nula.

Las series a comprobar son:

X1 = {0,013; 0,300; 0,708; 0,084; 0,155; 0,048; 0,185; 0,889} M=0,298; D=0,279
Y1 = {0,000; 0,299; 0,660; 0,084; 0,157; 0,036; 0,185; 0,814} M=0,279; D=0,281
D1 = {0,013; 0,001; 0,048; 0,000; 0,002; 0,012; 0,000; 0,075} M=0,019; D=0,026

Estadístico t = 1,846 < 2,365

X2 = {0,910; 0,706; 0,727; 0,760; 0,756; 0,969; 0,794; 0,995} M=0,827; D=0,106
Y2 = {0,958; 0,707; 0,706; 0,760; 0,761; 0,959; 0,795; 0,991} M=0,830; D=0,112
D2 = {0,048; 0,001; 0,021; 0,000; 0,005; 0,010; 0,001; 0,004} M=0,011; D=0,015

Estadístico t = 0,351 < 2,365

Y por último, los intervalos de error de cada clasificador se muestran a continuación, con los valores para la clasificación de pertenencia a la categoría primero:

PERTENENCIA A LA CATEGORÍA	CORPUS COMÚN	CORPUS PROPIO
Asociaciones	0,688 +- 0,227	1 +- 0
Blogs	0,249 +- 0,036	0,256 +- 0,036
Compañías	0,375 +- 0,019	0,315 +- 0,013
Festivales	0,891 +- 0,022	0,891 +- 0,022
Formación	0,409 +- 0,058	0,409 +- 0,058
Revistas	0,887 +- 0,079	0,887 +- 0,079
Salas Alternativas	0,387 +- 0,057	0,391 +- 0,057
Textos	0,022 +- 0,021	0,044 +- 0,030

FIGURA 3.23: Intervalo de error de pertenencia con corpus común vs específico

Como se puede comprobar en la mayoría de los casos los resultados son muy similares e incluso concordantes.

Respecto a la no-pertenencia a la categoría se tiene:

No-PERTENENCIA A LA CATEGORÍA	CORPUS COMÚN	CORPUS PROPIO
Asociaciones	0,163 +- 0,011	0,078 +- 0,008
Blogs	0,436 +- 0,015	0,434 +- 0,015
Compañías	0,171 +- 0,016	0,162 +- 0,016
Festivales	0,283 +- 0,014	0,284 +- 0,014
Formación	0,377 +- 0,014	0,370 +- 0,014
Revistas	0,049 +- 0,006	0,067 +- 0,007
Salas Alternativas	0,325 +- 0,014	0,323 +- 0,014
Textos	0,009 +- 0,003	0,016 +- 0,004

FIGURA 3.24: Intervalo de error de no-pertenencia con corpus común vs específico

Dónde también se puede apreciar que muchos intervalos son totalmente coincidentes y otros, aunque ligeramente movidos, tienen parte de su recorrido común.

3.6.3 Conclusiones

De los datos obtenidos se desprenden dos conclusiones directas. Por un lado la utilización de un corpus propio para cada categoría reduce significativamente el espacio de características, por lo que reafirma la primera hipótesis.

Así pues, tanto los resultados para la prueba F como para los intervalos muestran claramente la similitud entre ambos métodos, y así lo confirma el test t-student con una significación del 95%, que corrobora la hipótesis nula y por tanto rechaza nuestra segunda hipótesis de que esta reducción conseguiría un aumento en las prestaciones de los clasificadores.

En cualquier caso la utilización de corpus propios tiene mayores beneficios, pues reducen significativamente el tamaño de los datos a procesar, que perjuicios, pues aunque no introducen mejora, tampoco empeoran la calidad de los clasificadores.

Con los resultados de este experimento y el anterior se puede reafirmar el objetivo de comprobar la importancia del pre-procesado, y aunque no se puede afirmar que mejore el resultado de los clasificadores obtenidos con el preprocesado de los mismos, sí que se puede afirmar que reducen significativamente la dimensionalidad, lo que además de cumplir los objetivos enumerados en la introducción del experimento, excepto el segundo de mejorar el rendimiento de los clasificadores, también hacen que el proceso de creación de los mismos sea un proceso más ligero en el tiempo y menos costoso en espacio.

3.7 MEJORA A LA CLASIFICACIÓN BoW: CARACTERÍSTICAS CONTEXTUALES

La representación BoW consistirá en obtener las palabras de las páginas en cuestión que formen parte del corpus dado, obteniendo una frecuencia de aparición y formando ésta el valor asignado a la característica en cuestión, tal y como se vio y se describió en el experimento del punto 3.2.

Ahora bien existen diversos modos de obtener estas palabras así como diversos modos de ponderar el valor de las diferentes características, como se explica a continuación de manera más detallada a la realizada en aquél punto 3.2.

3.7.1 ¿Obtener palabras o contar apariciones?

La obtención de las palabras es una tarea relativamente sencilla. Con la aplicación de una expresión regular, previamente eliminadas las etiquetas html, se puede extraer el conjunto de palabras de un documento.

La expresión regular concreta es la siguiente:

```
\b(?<word>[a-zñáàéèíóòúëï]+)\b
```

En ella se observa que se obtiene la aparición una o más veces de todo aquél conjunto delimitado por separadores de caracteres alfabéticos, con ó sin acentos y con ó

sin diéresis. No se toman palabras que contengan números ni signos de cualquier tipo, pues no se consideran características de este tipo.

Mediante un proceso de stemming como el de Porter y cuyos efectos se estudiaron en el experimento 3.5 se obtiene el conjunto de lemas de la misma y se compara con el conjunto de lemas del corpus, anotando la frecuencia de aparición de cada palabra del corpus dividiendo por el número total de palabras.

El proceso es sencillo, pero con las Urls tiene una pega. Las Urls no suelen estar formadas por palabras completas y separadas por separadores estándar, como puedan ser en un documento textual la coma, el punto, el punto y coma, el espacio, etcétera. Es muy común que una url sea del tipo revistadeteatro.com o festivaldemerida.es/festivalesdeverano, es decir, formada por la concatenación de palabras en una única palabra resultante.

En estos casos el proceso de extracción de palabras resultaría en tres palabras únicamente, *revistadeteatro*, *festivaldemerida* y *festivalesdeverano*, que por otro lado, al no tener correspondencia en el stemmer, se perderían, o si no se aplica proceso de stem, formarían parte de las características eliminadas por tener una baja frecuencia de aparición.

Por ello, en la comparación con las Urls es más interesante el método de comparar que de obtener. Comparar significa obtener cada una de las palabras del corpus y contar el número de apariciones de la misma en el conjunto de palabras de la Url. De este modo, la palabra revista, o su lema revist, aparecería una vez en la primera Url y la palabra festival o su lema festiv aparecería dos.

3.7.2 ¿Ponderar o duplicar?

Cuando las características se toman de diferentes sitios de un documento, como por ejemplo las palabras del texto plano, del título, de la url, los enlaces o los encabezados, se debe tomar una decisión: ponderar los valores, mediante alguna distribución de pesos, para cada uno de los lugares anteriores, o duplicar las características tantas veces como lugares haya el conjunto de las mismas.

En el primero de los casos, el valor de una característica equivaldrá a la suma ponderada de los resultados obtenidos en cada lugar:

$$X(i) = f(links)*w(links) + f(body)*f(body) + ...$$

En el segundo de los casos cada uno de estos lugares tendrá una característica:

$$XLinks(i) = f(links)$$
$$XBody(i) = f(body)$$
$$...$$

En el estado del arte estudiado parece que se utilizan ambas representaciones. La ventaja de la ponderación es que reduce considerablemente la dimensionalidad respecto de la representación duplicada. Su inconveniente es que al mezclar datos disminuye la

capacidad de los modelos inductivos de encontrar patrones que podría encontrar duplicando las características.

3.7.3 Definición del experimento

Se realiza un experimento en el cuál se obtiene una representación BoW dónde se le ha dado mayor importancia a las palabras aparecidas en etiquetas específicas como los títulos (h1, h2 y h3) o en el título del documento, además de las encontradas en su cuerpo.

Para ello se obtiene la representación BoW mediante el método de obtener palabras con la expresión regular, al igual que en la representación de comparación base BoW, y dónde se pondera con un valor igual a uno para todos los casos la frecuencia de las palabras obtenidas de los encabezados, título y cuerpo.

- Se parte de la hipótesis de que la incorporación de información contextual como la anterior mejorará significativamente los resultados de la clasificación

Para compararlo se obtiene la representación BoW y BoW mejorada para los repositorios de prueba DS1 y DS2 y se realiza la validación cruzada de uno con el otro, obteniendo la media de aciertos y errores y construyendo el valor de los estadísticos F combinados ambos métodos, obteniendo el valor de F para la validación 2x2 ya introducida.

A la distribución de valores de la prueba F se le añade el conjunto de los intervalos de error obtenidos para cada clasificador en el caso de pertenencia y no-pertenencia a la clase, de manera que se pueda comparar visualmente la similitud o no de los mismos.

En caso de que a simple vista no se pueda determinar la superioridad del método propuesto sobre la base BoW estándar, se realizará un test de t-student pareado para determinar la igualdad o no de los resultados.

Una vez más la hipótesis nula indica que las distribuciones obtenidas son iguales por lo que aceptar dicha hipótesis significará rechazar nuestra hipótesis de mejora, y por el contrario, rechazar la hipótesis nula significará afirmar nuestra hipótesis de mejora y por tanto que la adición de datos contextuales mejora la clasificación estándar basada en BoW.

3.7.4 Realización del experimento y resultados

Se obtiene el modelo BoW estándar a partir de la obtención de las palabras que forman los documentos mediante la expresión regular para obtener palabras y se ponderan los valores de aparición en una característica por cada palabra cuyo valor será el número de repeticiones de la misma dividida por el número total de palabras del documento, al igual que en experimentos anteriores.

La representación BoW mejorada se obtiene de manera similar, pero en este caso se pondera la aparición de las palabras junto con la aparición de las mismas dentro de

los tags de encabezado H1, H2 y H3, de manera que cada aparición de una palabra sumará 1, así como cada aparición de una palabra en H1, en H2, y en H3 sumará 1 también, de modo que el valor final será el número de repeticiones en el cuerpo y en las etiquetas de encabezado, dividido por el número total de palabras.

Como se observa se ponderan los valores en lugar de duplicar características, y por tanto ambos métodos obtendrán representaciones con un número similar de características.

El resultado de ponderar valores será que las palabras que se encuentren enmarcadas en alguna etiqueta de las definidas tendrán mayor peso, en concreto el doble, puntuando más hacia la categoría a la que mejor definan.

Con ambas representaciones se realiza una validación 2x2 de cada clasificador binario, de modo que a continuación se tabula el resultado de las pruebas F para la pertenencia a la categoría correspondiente:

PERTENENCIA A LA CATEGORÍA	BoW std	BoW improv
Asociaciones	0	**0.016**
Blogs	0.299	**0.429**
Compañías	0.660	**0.848**
Festivales	0.084	**0.267**
Formación	0.157	**0.185**
Revistas	**0.036**	0.007
Salas Alternativas	**0.185**	0.122
Textos	**0.814**	0.454

FIGURA 3.25: Estadístico F en la clasificación de pertenencia BoW std vs. BoW mejorado

Como se puede apreciar, en los primeros casos la representación mejorada obtiene tasas superiores del test F, no así en las últimas categorías, dónde se invierte la situación y es la representación estándar la que mejor las obtiene.

En el caso de no-pertenencia a la categoría se obtienen los siguientes resultados:

No-PERTENENCIA A LA CATEGORÍA	BoW std	BoW improv
Asociaciones	**0.958**	0.755
Blogs	0.707	**0.889**
Compañías	0.706	**0.758**
Festivales	**0.760**	0.701
Formación	**0.761**	0.722
Revistas	**0.959**	0.567
Salas Alternativas	**0.795**	0.560
Textos	**0.991**	0.955

FIGURA 3.26: Estadístico F en la clasificación de no-pertenencia BoW std vs. BoW mejorado

Donde se aprecia una mejora en regla general del método estándar sobre el mejorado, al menos en más categorías que en el primero de los casos, aunque la variabilidad entre categorías puede resultar en un rendimiento general similar.

En ambos casos parece que para ciertas categorías el método es claramente superior aunque para otros casos se rechaza esta evidencia. Parece necesaria la aplicación del test de t-student para determinar el resultado:

Las series a comprobar son:

X1 = {0,000; 0,299; 0,660; 0,084; 0,157; 0,036; 0,185; 0,814}
Y1 = {0,016; 0,429; 0,848; 0,267; 0,185; 0,007; 0,122; 0,454}
D1 = {0,016; 0,130; 0,188; 0,830; 0,028; 0,029; 0,063; 0,360}

Estadístico t = 0,185 < 2,365

X2 = {0,958; 0,707; 0,706; 0,760; 0,761; 0,959; 0,795; 0,991}
Y2 = {0,755; 0,889; 0,758; 0,701; 0,722; 0,567; 0,560; 0,955}
D2 = {0,203; 0,182; 0,052; 0,059; 0,039; 0,392; 0,235; 0,036}

Estadístico t = 1,438 < 2,365

Lo que demuestra en ambos casos lo que se pensaba, y es que ambas representaciones obtienen por regla general una clasificación similar, siendo una mejor en ciertas categorías, y peor en otras.

Analizando los intervalos de error de cada clasificador que se muestran a continuación:

PERTENENCIA A LA CATEGORÍA	BoW std	BoW improv
Asociaciones	1 +- 0	0,375 +- 0,194
Blogs	0,256 +- 0,036	0,399 +- 0,041
Compañías	0,315 +- 0,013	0,046 +- 0,008
Festivales	0,891 +- 0,022	0,510 +- 0,036
Formación	0,409 +- 0,058	0,215 +- 0,048
Revistas	0,887 +- 0,079	0,839 +- 0,092
Salas Alternativas	0,391 +- 0,057	0,349 +- 0,054
Textos	0,044 +- 0,030	0,099 +- 0,008

FIGURA 3.27: Intervalo de error en la clasificación de pertenencia BoW std vs. BoW mejorado

NO PERTENENCIA A LA CATEGORÍA	BoW std	BoW improv
Asociaciones	0,078 +- 0,008	0,392 +- 0,014
Blogs	0,434 +- 0,015	0,158 +- 0,011
Compañías	0,162 +- 0,016	0,355 +- 0,020
Festivales	0,284 +- 0,014	0,409 +- 0,015
Formación	0,370 +- 0,014	0,428 +- 0,015
Revistas	0,067 +- 0,007	0,599 +- 0,014
Salas Alternativas	0,323 +- 0,014	0,602 +- 0,014
Textos	0,016 +- 0,004	0,082 +- 0,008

FIGURA 3.28: Intervalo de error en la clasificación de no-pertenencia BoW std vs. BoW mejorado

Se aprecia la gran disparidad de valores, comportándose mejor el clasificador mejorado en el caso de pertenencia de la clase que en el caso de no-pertenencia, aunque

en regla general dependerá mucho de la categoría, pero en cualquier caso no introduce una diferenciación excesiva entre ellas.

3.7.5 Conclusiones

Los resultados anteriores resultan complejos de explicar. Por un lado los valores de F para los diversos clasificadores muestran valores muy similares, lo que ha corroborado el test t-student.

Sin embargo el análisis de los intervalos de error parece indicar una diferenciación en el resultado de los clasificadores cuando se refieren a la pertenencia a la categoría o a la no-pertenencia.

Como se puede apreciar a simple vista, la mayoría de intervalos cuando se refieren a la no-pertenencia son mayores y más elevados en el caso de la mejora introducida. En cambio, cuando se refieren a la pertenencia a la clase suelen ser menores y más reducidos.

Todo parece apuntar a que el método de añadir información contextual como la de los títulos y encabezados añade certeza al método cuando se refiere a clasificar positivamente una clase, pero en cambio también genera mayor número de errores por falso positivo. De ahí que tanto los intervalos se comporten como indicamos, como que los valores de F sean tan similares, pues lo que ganan con la precisión en uno de los casos lo pierden con el alcance, de manera que los valores son más homogéneos y por tanto similares en ambos métodos.

Con lo anterior se puede concluir que no se cumple la hipótesis y el método de añadir información contextual no mejora la clasificación en líneas generales, ni tampoco parece que introduzca demasiada información en casos particulares como para ser tenida en cuenta a la hora de proponer una representación nueva que haga uso de ella.

3.8 MEJORA A LA CLASIFICACIÓN BoW: LA URL

En el estado del arte se estudió un clasificador basado únicamente en las palabras aparecidas en la URL [Kan 2004]

Su rendimiento era considerablemente superior a la línea base del estándar BoW, así como la velocidad de clasificación muy superior pues no necesitaba obtener la página para realizar la clasificación, sino únicamente la url que siempre se encuentra disponible.

La url, en contra de las recomendaciones de la WWW, generalmente incluye información sobre la página, en forma de subrutas dentro de la ruta general, o mediante algún tipo de prefijo indicando a qué parte de la estructura se accede.

Así pues, muchas páginas de tipo Blog tienen por algún sitio en su url dicha palabra. Es bastante común que si la url es legible por el usuario le aporte cierto grado de información.

3.8.1 Definición del experimento

En este experimento se considera la obtención de características de la url para realizar la clasificación y comprobar si se aporta mayor beneficio que la clásica BoW estándar en el ámbito concreto que nos ocupa.

- La hipótesis del experimento es que la mejora obtenida en los estudios [Kan 2004] se reproduce en el ámbito del teatro y por lo tanto el método basado en la Url es superior al BoW estándar, es decir, la hipótesis alternativa a la hipótesis nula, de modo que si se demuestra aquélla se rechazará esta.

3.8.2 Realización del experimento y resultados

Para ello, a partir de los corpus previamente obtenidos y utilizados en experimentos anteriores, se obtendrá el conjunto de palabras que de los mismos aparecen en las url mediante la técnica de contar apariciones, también descrita con anterioridad.

A partir de la bolsa de palabras resultante se efectúa el entrenamiento de los modelos de clasificador bayesiano y se evalúan mediante el método 2x2.

Se compara el valor de la prueba F así obtenida con el de la línea de base BoW estándar, así como el conjunto de los errores y sus intervalos.

En caso de no estar clara la diferencia se aplicará un test t-student para determinar la igualdad de las medias, hipótesis nula, o por el contrario la diferencia y por tanto superioridad de un método sobre otro.

Tras la obtención y validación de los clasificadores se obtienen los siguientes resultados, separando una vez más entre pertenencia y no-pertenencia a la categoría en cuestión:

PERTENENCIA A LA CATEGORÍA	BoW std	BoW url
Asociaciones	0	**0.009**
Blogs	0.299	**0.558**
Compañías	**0.660**	0.143
Festivales	0.084	**0.664**
Formación	0.157	**0.186**
Revistas	0.036	**0.040**
Salas Alternativas	**0.185**	0.140
Textos	**0.814**	0.569

FIGURA 3.29: Estadístico F en la clasificación de pertenencia BoW std vs. BoW url

En el caso anterior se muestra una clara variabilidad entre las representaciones dependiendo de la categoría para la cuál se realice el test, lo que claramente parece conducirnos a un rendimiento general muy similar.

En el caso de la no-pertenencia a la categoría se tienen los siguientes resultados:

No-PERTENENCIA A LA CATEGORÍA	BoW std	BoW url
Asociaciones	**0.958**	0.090

Blogs	0.707	**0.918**
Compañías	**0.706**	0.595
Festivales	0.760	**0.903**
Formación	**0.761**	0.684
Revistas	**0.959**	0.768
Salas Alternativas	**0.795**	0.491
Textos	**0.991**	0.987

FIGURA 3.30: Estadístico F en la clasificación de no-pertenencia BoW std vs. BoW url

Como se puede apreciar el resultado en alguna de las categorías es bastante diferente, siendo superior en algunas de ellas el método de las Urls y en otros el estándar. Será necesaria la utilización de un test t-student para determinar la superioridad de un método sobre el otro:

Las series de datos a analizar son:

X1 = {0,000; 0,299; 0,660; 0,084; 0,157; 0,036; 0,185; 0,814} M=0,279; D=0,281
Y1 = {0,009; 0,558; 0,143; 0,664; 0,186; 0,040; 0,140; 0,569} M=0,289; D=0,246
D1 = {0,009; 0,259; 0,517; 0,580; 0,029; 0,004; 0,045; 0,245} M=0,211; D=0,217

Estadístico t = 0,081 < 2,365

X2 = {0,958; 0,707; 0,706; 0,760; 0,761; 0,959; 0,795; 0,991} M=0,833; D=0,116
Y2 = {0,090; 0,918; 0,595; 0,903; 0,684; 0,768; 0,491; 0,987} M=0,679; D=0,274
D2 = {0,895; 0,211; 0,111; 0,143; 0,077; 0,191; 0,304; 0,004} M=0,242; D=0,261

Estadístico t = 0,231 < 2,365

Lo que muestra claramente en ambos casos que ambas distribuciones de valores de la prueba F tienen una media similar, lo que implica que ambas representaciones, en general, obtienen tasas similares de precisión y alcance, y por tanto se puede afirmar que ambas representaciones obtienen un rendimiento similar en el ámbito de estudio.

A continuación se analizan los intervalos de error de cada clasificador:

PERTENENCIA A LA CATEGORÍA	BoW std	BoW url
Asociaciones	1 +- 0	0,231 +- 0,162
Blogs	0,256 +- 0,036	0,241 +- 0,036
Compañías	0,315 +- 0,013	0,916 +- 0,011
Festivales	0,891 +- 0,022	0,150 +- 0,010
Formación	0,409 +- 0,058	0,145 +- 0,041
Revistas	0,887 +- 0,079	0,403 +- 0,122
Salas Alternativas	0,391 +- 0,057	0,170 +- 0,043
Textos	0,044 +- 0,030	0,582 +- 0,072

FIGURA 3.31: Intervalo de error en la clasificación de pertenencia BoW std vs. BoW url

En el caso de la pertenencia a la categoría se observa una amplia variabilidad dependiendo de la categoría, así pues en categorías como festivales la representación estándar obtiene una tasa de error del 89,1% frente al 15,0% que obtiene la basada en

Url, pero por contra en la categoría textos la representación estándar obtiene únicamente un 4,4% frente al 58,2% de la representación urls.

A continuación se muestran los resultados para la no-pertenencia a la categoría:

No-PERTENENCIA A LA CATEGORÍA	BoW std	BoW utl
Asociaciones	0,078 +- 0,008	0,231 +- 0,162
Blogs	0,434 +- 0,015	0,241 +- 0,036
Compañías	0,162 +- 0,016	0,916 +- 0,011
Festivales	0,284 +- 0,014	0,050 +- 0,016
Formación	0,370 +- 0,014	0,456 +- 0,014
Revistas	0,067 +- 0,007	0,403 +- 0,122
Salas Alternativas	0,323 +- 0,014	0,671+-0,014
Textos	0,016 +- 0,004	0,002 +- 0,001

FIGURA 3.32: Intervalo de error en la clasificación de no-pertenencia BoW std vs. BoW url

Lo que muestra una variabilidad muy similar al caso anterior, y demuestra que ambas representaciones serán muy dependientes de la categoría a la que se apliquen.

Por ello parece lógico pensar que aunque de manera general el clasificador basado en las características extraídas de la url por sí sólo no es superior a la representación estándar, para ciertas categorías sí que introduce una mejora considerable.

3.8.3 Conclusiones

La variabilidad de los resultados obtenidos para ambas clasificaciones indican que la adecuación de un modelo u otro dependerá de la categoría en cuestión que se esté clasificando, de manera que se puede concluir que una representación así no es interesante para ser utilizada de manera general, ya que no reportará un beneficio global a los clasificadores.

En cualquier caso dicha variabilidad también muestra unos resultados muy elevados en algunos de los casos, lo que parece indicar que la combinación de dichos resultados con un método más estable que obtenga buenos resultados en general podrá ser utilizada para mejorar la calidad aportando información en algunos de los casos.

De este experimento se desprende pues dos conclusiones principales:

- La no-adecuación del método de representación de la bolsa de palabras de las Urls para resolver el problema de clasificación de páginas web de teatro pues no obtiene unos resultados significativamente superiores a la representación de bolsa de palabras estándar
- La intuición de que la utilización de las características obtenidas por este tipo de representación, las palabras de la url, pueden aportar información significativa para mejorar otro tipo de clasificación más homogénea para todas las características, dando un apoyo extra en algunos de los casos y mejorando de este modo la clasificación por aquella obtenida.

3.9 CLASIFICACIÓN BASADA EN LA META-INFORMACIÓN: LA INTENCIÓN DEL AUTOR DE COMUNICAR INFORMACIÓN ACERCA DE LA PÁGINA

Como ya se introdujo en el análisis de alternativas, la investigación se va a llevar a cabo en la línea de intentar obtener el máximo poder discriminatorio de las palabras utilizadas para clasificar acudiendo al lugar dónde los autores de las páginas tienen mayor intención de comunicar el fin de las mismas.

En estudios como [Lewis 1992] se argumenta que las buenas características para la clasificación de texto deben cumplir las siguientes propiedades:

- Ser relativamente pocas en cuanto a número de las mismas
- Tener una frecuencia moderada de asignación
- Tener poca redundancia
- Generar poco ruido
- Estar sujetas al ámbito semántico de las clases a las que se van a asignar
- No ser ambiguas

Persiguiendo los objetivos anteriores se pone de manifiesto el acierto en la elección de la alternativa de seleccionar un corpus para cada tipo o categoría, de modo que se consiga reducir el número de características y que sean semánticamente más ajustadas al ámbito de la misma, aunque como se vio en el experimento 3.6 sí se consigue lo primero pero no así lo segundo.

Pero como ya se comentó, el problema desde el punto de vista cualitativo de clasificar las páginas utilizando para ello la información de su contenido está en que al tener todas una estrecha relación con una categoría superior, en este caso el teatro, el vocabulario utilizado compartirá muchas de estas características, formando un conjunto bastante común de las mismas, lo que generará redundancia, ambigüedad y en algunas ocasiones ruido, así como ciertas palabras tendrán una frecuencia muy elevada de aparición, como por ejemplo la palabra teatro, con lo que la diferenciación entre categorías se hará más difusa.

Si partimos del supuesto, bastante razonable, de que toda página está hecha por un autor, quién tiene como objetivo principal comunicar algo, podemos lanzar el supuesto de que ese algo que quiere comunicar quedará enmarcado en sitios concretos de la estructura del documento, en este caso escrito en html.

Al igual que un documento escrito puede tener las secciones del título, índice o tabla de contenidos, resumen o sumario, el propio texto, con sus encabezados de sección y demás, y una posible conclusión, un documento html, partiendo de las propias especificaciones técnicas del lenguaje, tendrá una estructura diferenciada entre encabezado, dónde se da diferente información técnica y de contenido del propio documento, y un cuerpo dónde se escribe el contenido del mismo, con etiquetas de marcado separando y remarcando cierta información.

En html se proporcionan una serie de etiquetas de marcado para especificar información concreta sobre el documento, lo que se denomina meta-información o meta-datos, puesto que aportan datos o información sobre los propios datos.

Los meta-tags, que es como se denominan en este lenguaje, permiten al creador de una página web comunicar cierta información sobre su sitio a los buscadores, servicios automáticos u otros creadores de páginas web.

Los meta-tags nacen orientados a la ayuda a los motores de búsqueda. Así pues, los meta-tags informan a las arañas del buscador del nombre del sitio, la descripción de del contenido del sitio que al creador le gustaría que se usase y las palabras clave por las cuales desearía que fuera encontrado.

Es por ello que estos meta-datos contienen la intención del autor de comunicar cierta información para que sea tratada por parte de los motores y de algún modo catalogada en sus índices, y por tanto, contendrán información muy específica sobre la página, al margen de su contenido, y que será susceptible de ser una importante fuente de información para la clasificación.

El principal problema, como también se comentó en el análisis inicial, es que no siempre esta meta-información se encuentra informada. Muchos creadores de páginas no se detienen a definir correctamente la cabecera de sus documentos, con lo que queda incompleta. Otras veces, aunque la información se encuentre no es relevante, es ambigua o no se corresponde con el contenido de la página, bien por técnicas maliciosas de spamdexing, o simplemente por descuido o desconocimiento del creador de la misma.

[Pierre 2000] realizan un estudio a partir de 29.998 Webs de la cantidad de palabras utilizadas en los diferentes meta-datos, así como las utilizadas en el cuerpo completo del documento, resultando en la siguiente tabla, que se extrae del propio estudio:

TAG	0 palabras	1-10 palabras	11-50 palabras	+51 palabras
Title	4%	89%	6%	1%
Meta-Description	68%	8%	21%	3%
Meta-Keywords	66%	5%	19%	10%
Body	17%	5%	21%	57%

FIGURA 3.33: Número de palabras en secciones HTML

Como se puede observar existe un alto porcentaje de páginas que no tienen información en sus meta-datos descripción y keywords, aunque se compensa con la información del título, aunque esta información no siempre es válida para una clasificación por contener generalmente una descripción más en el ámbito del marketing o la descripción de marca.

El porcentaje de páginas que no tienen información en su cuerpo también es elevado, un 17%, y aunque no tanto como la cabecera, suficiente para mermar la clasificación de los mismos. Esto se debe a la utilización de marcos, a la incorporación de diseños basados en mapas de imágenes, al uso de la redirección, y a la incorporación cada vez más frecuente de elementos multimedia como.

Tal y como se puede observar, la gran cantidad de páginas que tienen elevado número de palabras en el cuerpo hace que varias de las condiciones descritas al principio para una buena representación no se puedan cumplir, esto es, por regla general existirá un número elevado de características, seguramente bastante redundantes, con elevado ruido y no siempre ajustadas al ámbito semántico del documento.

En cambio la utilización de los meta-tags de la cabecera del documento, cuando existen, siendo este su principal problema, cumplirá gran parte de los requisitos, pues se ceñirá más a la información precisa sobre la página, resultando además en un conjunto de características más limitado, más preciso en el ámbito semántico de la categoría sobre la que clasificarlo, redundante en su objetivo por comunicar la intención del autor, y generalmente con poco ruido por apuntarse en ella sólo palabras concretas con un objetivo fijo, el de describir a la web.

Por otro lado se tiene la información de los enlaces. Los enlaces son la vía facilitada por el hipertexto bien para relacionar la información actual con otra información que la complete, complemente o amplíe, bien con otra sección que la desarrollo y estructure.

Siguiendo las pautas de la usabilidad cualquier creador de páginas web intentará que sus enlaces muestren una descripción detallada del destino de los mismos, de manera que se invite al usuario a seguirlos, sabiendo hacia dónde se dirige, proporcionándole la información que necesita.

Esto no siempre es así; los mapas de imágenes, los enlaces con imágenes en lugar de texto, los enlaces del tipo “pinche aquí”..., son prácticas poco recomendables desde la usabilidad y la accesibilidad, pero que en muchos casos se siguen utilizando. En estos casos la información aportada es nula, pero aún así en muchos de los casos esto no es así, y por tanto son una fuente interesante de información que complementa y aumenta la calidad de lo anterior.

Por otro lado se encuentra el uso de las Urls para localizar las páginas o los documentos dentro de ellas. Según las especificaciones de la W3C las Urls deberían ser inocuas y no proporcionar información sobre su destino, siendo simplemente un modo de nombrar al mismo. En realidad el uso de palabras que aporten significado sobre el destino de las mismas es una práctica común, principalmente para dar orden a la propia estructura del sitio en el momento de la creación, y para su posterior mantenimiento, más que por la propia intención del creador de comunicar dicha información.

Pero aunque está falta de intención no así de interés por parte del autor por estructurar el contenido, lo que puede proporcionarnos una fuente de información interesante en la mayoría de los casos. Estudios como [Kan 2004] lo ponen de manifiesto en sus resultados, en una clasificación basada exclusivamente en la url, obtienen una precisión bastante elevada y en cualquier caso superior a la línea base del BoW.

Toda esta información nos viene dada por dos vías. Por un lado por la propia url de la página en cuestión, que en la mayoría de los casos incluirá algún tipo de información en la ruta que pueda indicar el tipo de página que estamos visitando, y por

otro en las diferentes direcciones url de los enlaces a otras páginas, que complementará la información obtenida del texto de los mismos. Así pues, una imagen o un enlace tipo "pinche aquí" que no aporta información, si su url es del tipo "...altacliente.php" o "formalizamatricula.asp" nos aporta una información extra que de otro modo perderíamos.

La combinación de los tres tipos de información anteriores, obtenidos de la cabecera del documento (head), los enlaces (links) y la url, determinan el método seguido en la investigación actual H&L&U.

Con todo lo anterior sólo queda analizar la repercusión de la utilización de toda esta meta-información y compararla con la línea base de extracción del corpus a partir del cuerpo del documento, comprobando la significativa reducción de palabras que lo forman y comprobando la adecuación a los postulados iniciales mediante el estudio de la ratio de aparición de ciertas palabras ligadas semánticamente al contenido del documento de la categoría dada.

Para ello se va a analizar cada una de las categorías de documentos, intentando extraer información de los mismos como posibles palabras que los definan mejor, y comprobando, mediante un estudio comparativo de su corpus específico con el corpus general, extraído de ambas maneras descritas (body vs h&l&u) fundamentar la intuición perseguida.

Una intuición base que se comprobará con la experimentación con datos reales es que en un conjunto de páginas que pueden pertenecer a un conjunto no disjunto de categorías, cuánto más disjuntas aparezcan las categorías frente a los atributos que las definen, mayor será la probabilidad de que las páginas se asignen a las categorías correctas pues los valores para dichos atributos serán más identificativos. Dicho de otro modo, cuanto más se parezcan los elementos de un conjunto entre sí y menos a los de otro conjunto, más sencilla será la tarea de discriminar entre ellos y de algún modo realizar una separación que permita la correcta categorización de los mismos.

3.9.1. Análisis de categorías

El análisis de categorías consiste en un estudio comparativo de los corpus obtenidos a partir del cuerpo de los documentos mediante un BoW estándar y el obtenido por el método propuesto H&L&U mediante un BoW a partir de las palabras obtenidas de sus tres puntos clave, la cabecera, los enlaces y las Urls (url de la página + url de los enlaces)

Analizando el corpus obtenido y ordenando por número de repeticiones así como por la ratio de este corpus específico frente al corpus general, se obtiene una serie de palabras significativas para el contexto de la categoría analizada. Comparando finalmente el incremento o decremento del uso de las mismas según el método utilizado, cuerpo o h&l&u, se formularán las primeras hipótesis basadas en la intuición descrita con anterioridad y base de esta propuesta.

El número de palabras diferentes es el conjunto de características que compone el corpus, primero las del específico y tras estas las del general. El número de palabras totales es el número total de palabras de todos los documentos que componen uno y otro

corpus, eliminado de estas los tags html, las palabras vacías y aplicando un proceso de stem del mismo.

La primera columna contiene aquellas palabras más ligadas semánticamente a la categoría analizada, y que además se hayan repetido un número considerable de veces en el corpus específico y no en el general, cumpliendo de este modo que sean representativas de la categoría y disjuntas del resto de categorías. Su selección ha sido un proceso no exhaustivo, pues no se trata de realizar una selección de atributos, sino de comparar las características de representatividad y discriminatoriedad de las palabras seleccionadas para poder formular hipótesis a priori a partir de la capacidad de clasificación por uno y otro método.

La segunda columna es el número de repeticiones de las palabras en el corpus específico y la tercera la ratio frente al número de palabras total. Cuanto mayor sea el valor de la segunda y más cerca del 100% esté el de la tercera, dicha palabra tendrá mayor frecuencia en el corpus específico y por tanto mejor definirá a la categoría en cuestión, con lo que mayor será su representatividad.

La cuarta columna es el número de repeticiones de las palabras en el corpus general. y la quinta columna es el ratio de repeticiones de la palabra en el corpus específico frente al corpus general. Esta ratio, cuanto más cercana sea al 100%, más indicará que dicha característica pertenece exclusivamente a la categoría en cuestión, y por tanto tendrá gran poder discriminador.

Una palabra, atributo o característica será tanto mejor para la clasificación cuanto mayor sea el valor de la tercera columna y además también lo sea el de la quinta, es decir, mejor represente a la categoría a la que pertenece y mejor discrimine o se diferencie del resto de categorías.

Todas las categorías se analizarán de igual manera, por lo que la anterior explicación se aplica para todas ellas.

3.9.1.1 Festivales

Analizando el conjunto de páginas catalogadas inicialmente como festivales rápidamente aparecen una serie de características o palabras que salen con frecuencia y que tienen alta relación semántica con la categoría.

Algunas de ellas son la propia palabra *festival*, *muestra* como algo equivalente a *festival*, *obra* y *espectáculo* como algo que se realiza o se muestra en el festival, *actores* y *autores* de las mismas, y así una serie de palabras que se enumeran en las siguientes tablas.

CORPUS: Festivales **MÉTODO:** Body **PALABRAS DIFERENTES:** 5.375 / 7.019 **PALABRAS TOTALES:** 99.019 / 676.682				
PALABRA	**Nº APARICIONES**	**RATIO**	**Nº APARICIONES TOTALES**	**RATIO**
actor	724	0,73%	3388	21,37%
autor	1305	1,32%	2617	49,87%
colabor	256	0,26%	1093	23,42%

compañ	695	0,70%	4245	16,37%
direccion	696	0,70%	2323	29,96%
director	674	0,68%	1373	49,09%
encuentr	156	0,16%	1255	12,43%
escen	435	0,44%	1961	22,18%
espectacul	637	0,64%	8762	7,27%
festival	284	0,29%	2604	10,91%
muestr	1199	1,21%	2478	48,39%
obra	1389	1,40%	4239	32,77%
premi	753	0,76%	2470	30,48%
present	157	0,16%	2290	6,86%
program	96	0,10%	1322	7,26%
tecnic	643	0,65%	1373	49,09%

FIGURA 3.34: Corpus Festivales obtenido por método Body

CORPUS: Festivales **MÉTODO:** l&h&u **PALABRAS DIFERENTES:** 651 / 2.954 **PALABRAS TOTALES:** 26.322 / 1.076.679				
PALABRA	**Nº APARICIONES**	**RATIO**	**Nº APARICIONES TOTALES**	**RATIO**
actor	7	0,03%	270	2,59%
autor	2951	11,21%	3678	80,23%
colabor	-	0,00%		
compañ	3	0,01%	610	0,05%
direccion	1	0,00%	218	0,05%
director	-	0,00%		
encuentr	93	0,35%	1029	9,04%
escen	25	0,09%	172	14,53%
espectacul	2	0,01%	3614	0,00%
festival	53	0,20%	2029	0,026%
muestr	1549	5,88%	1832	84,55%
obra	2248	8,54%	3448	65,20%
premi	7	0,03%	1322	0,05%
present	38	0,14%	831	4,57%
program	588	2,23%	826	71,19%
tecnic	-	0,00%		

FIGURA 3.35: Corpus Festivales obtenido por método l&h&u

Como se puede comparar, la propiedad de ser pocas características queda demostrada por el número de palabras totales del corpus generado con el cuerpo, 2.954 características, frente al generado con h&l&u, 651.

Así mismo, en un conjunto más reducido de características, donde la repetición de las mismas se valore en cuatro lugares diferentes en lugar de en uno (h&h&l se evalúa en la cabecera, el texto de los enlaces, el enlace y la url), dará lugar a mayor repetición de las palabras, y si se cumplen las espectativas, mayor agrupación en torno a algunas de ellas. Según los datos, el número total de repeticiones en la obtención del cuerpo es de 99.019 frente a las 26.322 del método h&l&u, que por la cantidad de características es muy superior al anterior en proporción.

Como se puede observar en los cuadros anteriores, palabras como *muestra* (1,21%/5,88%), *obra* (1,40%/8,54%) y *autor* (1,32%/11,21%) tienen un alto número de apariciones en ambos métodos, aumentando considerablemente en el segundo. Así mismo, otras palabras como *dirección* (0,79%/0%), *director* (0,68%/0%) y *premio* (0,76%/0,03%) muestran un descenso en el número de apariciones. Si la ratio comparativa entre el corpus específico y el general sufre para las mismas palabras un incremento se estará mostrando una separación entre categorías, algo similar a cómo un zoom sobre un grupo de elementos los separa cuando te acercas.

Así pues, muestra (48,39% / 84,55%), obra (32,77% / 65,20%) y autor (49,87% / 80,23%) muestran un incremento significativo, tal y como dirección (29,96% / 0,05%), director(49,09% / 0%) y premio(30,48%/0,05%) muestran un descenso significativo, apoyando con números lo descrito anteriormente.

En la siguiente tabla se muestra comparativamente el número de apariciones y las ratios de los corpus obtenidos mediante ambos métodos, resaltando en negrita los aumentos más significativos.

COMPARATIVAMENTE				
PALABRA	**Nº APARICIONES BODY**	**Nº APARICIONES H&L&U**	**RATIO BODY**	**RATIO H&L&U**
actor	724	7	21,37%	2,59%
autor	**1305**	**2951**	**49,87%**	**80,23%**
colabor	256	-	23,42%	
compañ	695	3	16,37%	0,05%
direccion	696	1	29,96%	0,05%
director	674	-	49,09%	
encuentr	156	93	12,43%	9,04%
escen	435	25	22,18%	14,53%
espectacul	637	2	7,27%	0,00%
festival	284	53	10,91%	0,026%
muestr	**1199**	**1549**	**48,39%**	**84,55%**
obra	**1389**	**2248**	**32,77%**	**65,20%**
premi	753	7	30,48%	0,05%
present	157	38	6,86%	4,57%
program	96	588	7,26%	71,19%
tecnic	643	-	49,09%	

FIGURA 3.36: Comparativa corpus Festivales

Los números anteriores ponen de manifiesto las propiedades de h&l&u para conseguir de las palabras las características que deben cumplir para ser buenas a la hora de usarlas para clasificar documentos.

3.9.1.2 Formación

Una rápida visita a las Webs de formación muestra que las palabras más relacionadas semánticamente con dicha categoría son las referentes a la *solicitud*, *admisión*, *matrícula*, *cursos*, *formación*, *estudios*, *carreras*, *alumnos*, *profesores*...

Un estudio del corpus obtenido por ambos métodos arroja los datos resumidos en la siguientes tablas:

CORPUS: Formación **MÉTODO:** Body **PALABRAS DIFERENTES:** 3.274 / 7.019 **PALABRAS TOTALES:** 59.196 / 676.682				
PALABRA	**Nº APARICIONES**	**RATIO**	**Nº APARICIONES TOTALES**	**RATIO**
admisión	62	0,10%	84	73,81%
alumn	206	0,35%	341	60,41%
carrer	221	0,37%	322	68,63%
curs	601	1,02%	1825	32,93%
ejercici	189	0,32%	230	82,17%
escuel	403	0,68%	829	48,61%
estudi	342	0,58%	1074	31,85%
formacion	200	0,34%	831	24,07%
matricul	75	0,13%	109	68,81%
practic	250	0,42%	419	59,67%
profesor	237	0,40%	453	52,32%
solicitud	122	0,21%	134	91,04%

FIGURA 3.37: Corpus Formación obtenido por método Body

CORPUS: Formación **MÉTODO:** h&l&u **PALABRAS DIFERENTES:** 697 / 2.954 **PALABRAS TOTALES:** 33.404 / 1.076.679				
PALABRA	**Nº APARICIONES**	**RATIO**	**Nº APARICIONES TOTALES**	**RATIO**
admision	209	0,63%	209	100%
alumn	27	0,08%	44	61,36%
carrer	11	0,03%	14	78,57%
curs	3563	10,67%	4489	79,37%
ejercici	-	0,00%		
escuel	2287	6,85%	2388	95,77%
estudi	180	0,54%	626	28,75%
formacion	129	0,39%	955	13,51%
matricul	-	0,00%		
practic	4	0,01%	7	57,14%
profesor	52	0,16%	73	71,23%
solicitud	80	0,24%	89	89,89%

FIGURA 3.38: Corpus Formación obtenido por método l&h&u

En este caso, como se puede observar, palabras como *admisión* (0,10% / 0,63%), curso (1,02% / 10,67%) o escuela (0,68% / 6,85%) aumentan considerablemente su número de apariciones, así como su ratio frente al conjunto total *admisión* (73,81% / 100%), *curso* (32,93% / 79,37%) y *escuela* (48,61% / 95,77%)

Otras palabras como *alumno* (0,35%/0,08%), *carrera* (0,37%/0,03%) o *práctica* (0,42%/0,01%) tienen un notable descenso en la frecuencia de aparición, pero sus ratios *alumno* (60,41%/61,36%), *carrera* (68,63% / 78,57%) y *práctica* (59,67% / 57,14%) se mantienen o incluso en algunos casos aumentan, lo que parece indicar que es un descenso de la frecuencia generalizado motivado quizás por una menor aparición en enlaces y Urls, pero su poder discriminatorio se mantiene.

Sí que es importante remarcar palabras como *formación* con (0,34% / 0,39%) apariciones, que aunque muestre un ligero aumento, muestran un descenso brusco de la ratio (24,07% / 13,51%) lo que muestra que no es una palabra muy discriminante, pues aparece en muchas más páginas diferentes, pese a que sea la misma palabra que define la categoría.

COMPARATIVAMENTE				
PALABRA	**Nº APARICIONES BODY**	**Nº APARICIONES H&L&U**	**RATIO BODY**	**RATIO H&L&U**
admision	**62**	**209**	**73,81%**	**100%**
alumn	206	27	60,41%	61,36%
carrer	221	11	68,63%	78,57%
curs	**601**	**3563**	**32,93%**	**79,37%**
ejercici	189	-	82,17%	
escuel	**403**	**2287**	**48,61%**	**95,77%**
estudi	342	180	31,85%	28,75%
formacion	200	129	24,07%	13,51%
matricul	75	-	68,81%	
practic	250	4	59,67%	57,14%
profesor	237	52	52,32%	71,23%
solicitud	122	80	91,04%	89,89%

FIGURA 3.39: Comparativa corpus Formación

3.9.1.3 Asociaciones

La categoría asociaciones, como se comprueba en la distribución de páginas por categoría, tiene un conjunto muy reducido de documentos, lo que provoca que se genere un corpus extremadamente pequeño de palabras comparándolas con categorías como las anteriores.

Es por ello que es más complejo obtener palabras que liguen semánticamente con la categoría y a su vez cumplan las propiedades de representar a la misma, y sobre todo de discriminar a otras categorías.

Pese a ello parece ser que las palabras que más contenido semántico tienen que ligue con la categoría asociaciones son precisamente *agrupación*, *grupo* y *asociación*.

CORPUS: Asociaciones **MÉTODO:** body **PALABRAS DIFERENTES:** 522 / 7.019 **PALABRAS TOTALES:** 1537 / 676.682				
PALABRA	**Nº APARICIONES**	**RATIO**	**Nº APARICIONES TOTALES**	**RATIO**

agrup	5	0,33%	73	6,85%
asoci	32	2,08%	784	4,08%
grup	34	2,21%	1136	2,99%

FIGURA 3.40: Corpus Asociaciones obtenido por método Body

CORPUS: Asociaciones **MÉTODO:** h&l&u **PALABRAS DIFERENTES:** 219 / 2.954 **PALABRAS TOTALES:** 1009/ 1.076.679				
PALABRA	**Nº APARICIONES**	**RATIO**	**Nº APARICIONES TOTALES**	**RATIO**
agrup	1	0,10%	13	7,69%
asoci	23	2,28%	326	7,05%
grup	12	1,19%	2284	0,52%

FIGURA 3.41: Corpus Asociaciones obtenido por método l&h&u

Como se puede observar en el cuadro comparativo la representatividad de las mismas no es muy elevada, a excepción de asociación que consigue algo más del 2% de las repeticiones totales, pero aún peor es la parte de discriminación, no superando el umbral del 8% en ninguno de los casos.

Lo anterior se debe principalmente a la enorme diferencia entre la cantidad de páginas del tipo asociación con el resto, siendo ésta una minúscula parte del total, tal y como se comentó en el punto 3.2.5

Todo lo visto tendrá repercusiones en la clasificación de las páginas de esta categoría, lo que no parece es que se pueda predecir el comportamiento del H&L&U frente al estándar, pues los datos abajo resumidos no arrojan demasiada luz al asunto.

COMPARATIVAMENTE				
PALABRA	**RATIO APARICIONES BODY**	**RATIO APARICIONES H&L&U**	**RATIO BODY**	**RATIO H&L&U**
agrup	0,33%	0,10%	6,85%	7,69%
asoci	2,08%	2,28%	4,08%	7,05%
grup	2,21%	1,19%	2,99%	0,52%

FIGURA 3.42: Comparativa Corpus Asociaciones

3.9.1.4 Compañías

El análisis de las compañías de teatro muestra los datos que se enumeran en las siguientes tablas.

A simple vista se observa un gran número de características que teniendo un grado considerable de aparición consiguen una ratio bastante elevada, con lo que parece ser que definirán bastante bien a la categoría compañías.

CORPUS: Compañías **MÉTODO:** body **PALABRAS DIFERENTES:** 3091 / 7.019 **PALABRAS TOTALES:** 265.960 / 676.682				
PALABRA	**Nº APARICIONES**	**RATIO**	**Nº APARICIONES TOTALES**	**RATIO**
carnaval	637	0,24%	653	97,55%

compañ	865	0,33%	4245	20,38%
consul	2504	0,94%	2506	99,92%
cultural	578	0,22%	1361	42,47%
espectacul	6883	2,59%	8762	78,55%
estil	8	0,00%	149	5,37%
itiner	1138	0,43%	1177	96,69%
moment	10373	3,90%	10727	96,70%
personaj	1788	0,67%	2419	73,91%
transeunt	741	0,28%	743	99,73%
vestuari	1743	0,66%	2238	77,88%

FIGURA 3.43: Corpus Compañías obtenido por método Body

CORPUS: Compañías **MÉTODO:** l&h&u **PALABRAS DIFERENTES:** 707 / 2.954 **PALABRAS TOTALES:** 143.659 / 1.076.679				
PALABRA	**Nº APARICIONES**	**RATIO**	**Nº APARICIONES TOTALES**	**RATIO**
carnaval	6	0,00%	9	66,66%
compañ	157	0,11%	610	25,74%
consul	-	0,00%		
cultural	2027	1,41%	2542	79,74%
espectacul	2965	2,06%	3614	82,04%
estil	2372	1,65%	2732	86,82%
itiner	1	0,00%	26	3,85%
moment	9235	6,43%	9235	100%
personaj	6	0,00%	93	6,45%
transeunt	-	0,00%		
vestuari	-	0,00%		

FIGURA 3.44: Corpus Compañías obtenido por método l&h&u

Comparativamente se puede apreciar un incremento elevado de algunas palabras como *cultural*, de un 0,22% a un 1,41% en frecuencia de aparición y de un 42,47% a un 79,74% en ratio, *estilo* de una aparición de poco más del 0% a un 1,65% y una ratio de un 5,37% a un 86,82%, o *momento* de un 3,90% a un 6,43% en frecuencia y de un 96,70% a un 100% en ratio.

La palabra *compañía* no tiene unos valores demasiado interesantes, y las palabras que sí lo tienen no parece que tengan demasiada correspondencia semántica con la categoría, por lo que la precisión de la clasificación puede estar más condicionada a los ejemplos y no ser tan generalista, aunque los resultados mostrarán más información al respecto.

COMPARATIVAMENTE				
PALABRA	**RATIO APARICIONES BODY**	**RATIO APARICIONES H&L&U**	**RATIO BODY**	**RATIO H&L&U**
carnaval	0,24%	0,00%	97,55%	66,66%
compañ	0,33%	0,11%	20,38%	25,74%
consul	0,94%	0,00%	99,92%	0,00%
cultural	**0,22%**	**1,41%**	**42,47%**	**79,74%**

espectacul	2,59%	2,06%	78,55%	82,04%
estil	**0,00%**	**1,65%**	**5,37%**	**86,82%**
itiner	0,43%	0,00%	96,69%	3,85%
moment	**3,90%**	**6,43%**	**96,70%**	**100%**
personaj	0,67%	0,00%	73,91%	6,45%
transeunt	0,28%	0,00%	99,73%	0,00%
vestuari	0,66%	0,00%	77,88%	0,00%

FIGURA 3.45: Comparativa Corpus Compañías

3.9.1.5 Revistas

Observando las páginas de revistas se extrae rápidamente una serie de características que parece ser común a todas ellas, como por ejemplo la palabra *artículo*, *comentario*, *edición* y *número*, *editor* o *entrevista*. Así mismo *fotografía* y derivados, *noticias*, *libros* y la propia palabra *revista* son algunas de las características que se extraen del corpus, como se ve a continuación:

CORPUS: Revistas **MÉTODO:** body **PALABRAS DIFERENTES:** 2557 / 7.019 **PALABRAS TOTALES:** 35.262 / 676.682				
PALABRA	**Nº APARICIONES**	**RATIO**	**Nº APARICIONES TOTALES**	**RATIO**
articul	25	0,07%	253	9,88%
column	4	0,01%	38	10,53%
coment	218	0,62%	2925	7,45%
edicion	119	0,34%	1007	11,82%
edit	68	0,19%	751	9,05%
entrev	15	0,04%	107	14,02%
escritor	144	0,41%	485	29,69%
fotograf	255	0,72%	1977	12,90%
indic	6	0,02%	63	9,52%
libr	194	0,55%	1499	12,94%
notici	14	0,04%	691	2,03%
numer	25	0,07%	721	3,47%
premi	289	0,82%	2470	11,70%
present	189	0,54%	2290	8,25%
revist	155	0,44%	712	21,77%
seccion	15	0,04%	630	2,38%
sumario	2	0,01%	43	4,65%

FIGURA 3.46: Corpus Revistas obtenido por método Body

CORPUS: Revista **MÉTODO:** h&l&u **PALABRAS DIFERENTES:** 472 / 2.954 **PALABRAS TOTALES:** 7.855 / 1.076.679				
PALABRA	**Nº APARICIONES**	**RATIO**	**Nº APARICIONES TOTALES**	**RATIO**
articul	-	0,00%		
column	-	0,00%		
coment	163	2,08%	3878	4,20%
edicion	11	0,14%	283	3,89%
edit	36	0,46%	394	

entrev	2	0,03%	60	3,33%
escritor	58	0,74%	186	31,18%
fotograf	126	1,60%	891	14,14%
indic	1	0,01%	8	12,5%
libr	18	0,23%	962	1,87%
notici	92	1,17%	1318	6,98%
numer	10	0,13%	29	34,48%
premi	153	1,95%	1322	11,57%
present	63	0,80%	831	7,58%
revist	61	0,78%	557	10,95%
seccion	-	0,00%		
sumario	1	0,01%	42	2,38%

FIGURA 3.47: Corpus Revistas obtenido por método l&h&u

Lo primero que llama la atención es que pese a tener un corpus suficientemente amplio no se alcanza con ninguna palabra un grado de diferenciación mayor de un 35%, por lo que no se puede hablar de características representativas de la categoría revistas. Para ello están los modelos inductivos, para encontrar lo que a simple vista no somos capaces de ver, y en los resultados se observará si esta afirmación se ha correspondido con la realidad o no.

Respecto al análisis que nos ocupa, se puede observar el incremento en ambos valores en algunos de los casos, siendo el más significativo el de la palabra *número*, que pasa de una ratio del 3,47% al 34,48%, lo que es un aumento considerable. Esto seguramente es debido a que las revistas tienen diferentes números, y esto en un sitio web se plasma en enlace hacia los diferentes números de la misma, lo que aumenta considerablemente la repetición de los mismos, a diferencia de otras categorías.

COMPARATIVAMENTE				
PALABRA	**RATIO APARICIONES BODY**	**RATIO APARICIONES H&L&U**	**RATIO BODY**	**RATIO H&L&U**
articul	0,07%	0,00%	9,88%	
column	0,01%	0,00%	10,53%	
coment	**0,62%**	**2,08%**	**7,45%**	**4,20%**
edicion	0,34%	0,14%	11,82%	3,89%
edit	0,19%	0,46%	9,05%	
entrev	0,04%	0,03%	14,02%	3,33%
escritor	**0,41%**	**0,74%**	**29,69%**	**31,18%**
fotograf	0,72%	1,60%	12,90%	14,14%
indic	0,02%	0,01%	9,52%	12,5%
libr	0,55%	0,23%	12,94%	1,87%
notici	**0,04%**	**1,17%**	**2,03%**	**6,98%**
numer	**0,07%**	**0,13%**	**3,47%**	**34,48%**
premi	0,82%	1,95%	11,70%	11,57%
present	0,54%	0,80%	8,25%	7,58%
revist	0,44%	0,78%	21,77%	10,95%
seccion	0,04%	0,00%	2,38%	
sumario	0,01%	0,01%	4,65%	2,38%

FIGURA 3.48: Comparativa Corpus Librerías

3.9.1.6 Salas Alternativas

La visualización de las páginas de salas alternativas nos demuestra una clara tendencia a autodefinirse como tales, por lo que las palabras *sala* y *alternativa* deben tener bastante frecuencia.

Por otro lado, la *taquilla*, el *programa* y las *escenas* son partes fundamentales de las salas, así como el *espectáculo*, la *danza* y los *artistas*.

Por último, está claro que en toda sala actúan *compañías*, así como se pueden dar *cursos*.

Por supuesto, una sala alternativa estará muy ligada a *teatro contemporáneo*, por lo que es una palabra que debe aparecer.

Así pues, tras un análisis del corpus también es interesante hacer notar la aparición de la palabra *obrador* por alcanzar una ratio, en ambos métodos, del 100% teniendo a su vez una frecuencia de aparición bastante elevada.

CORPUS: Salas alternativas **MÉTODO:** body **PALABRAS DIFERENTES:** 2.989 / 7.019 **PALABRAS TOTALES:** 35.262 / 676.682				
PALABRA	**Nº APARICIONES**	**RATIO**	**Nº APARICIONES TOTALES**	**RATIO**
artist	173	0,49%	1540	11,23%
compañ	208	0,59%	4245	4,90%
contemporan	116	0,33%	610	19,02%
curs	173	0,49%	1825	9,48%
danz	297	0,84%	1778	16,70%
escen	199	0,56%	1961	10,15%
espectacul	166	0,47%	8762	1,89%
obrador	197	0,56%	197	100%
part	245	0,69%	1704	14,38%
program	337	0,96%	1322	25,49%
sal	601	1,70%	2917	20,60%
taquill	75	0,21%	88	85,23%

FIGURA 3.49: Corpus Salas Alternativas obtenido por método Body

CORPUS: Salas alternativas **MÉTODO:** h&l&u **PALABRAS DIFERENTES:** 467 / 2.954 **PALABRAS TOTALES:** 7.855 / 1.076.679				
PALABRA	**Nº APARICIONES**	**RATIO**	**Nº APARICIONES TOTALES**	**RATIO**
artist	22	0,28%	1122	1,96%
compañ	10	0,13%	610	1,64%
contemporan	201	2,56%	241	83,40%
curs	121	1,54%	4489	2,69%
danz	21	0,27%	214	9,81%
escen	65	0,83%	172	37,79%
espectacul	3	0,04%	3614	0,00%

obrador	190	2,42%	190	100%
part	87	1,11%	154	56,49%
program	132	1,68%	826	15,98%
sal	415	5,28%	1325	31,32%
taquill	2	0,03%	2	100%

FIGURA 3.50: Corpus Salas Alternativas obtenido por método l&h&u

Tal y como se aprecia con las ratios, las palabras *contemporáneo* (0,33% / 2,56%), *curso* (0,49% / 1,54%) y *obrador* (0,56% / 2,42%) tienen un incremento en su aportación total al corpus bastante significativo, aunque el incremento más interesante lo genera la palabra *sala* (1,70% / 5,28%), palabra ligada a la propia categoría y que supone un buen porcentaje del texto del documento.

Comparando a su vez la ratio de aparición entre el corpus específico y el general, se observa que *contemporáneo* (19,02% / 83,40%) tiene un incremento sustancial, así como *parte* (14,38% / 56,49%)

La palabra *sala* que consiguió un incremento sustancial en frecuencia de aparición también lo consigue, aunque algo menos, en ratio frente al corpus general, de un 20,60% a un 31,32%, y ambos valores en conjunto harán de dicha palabra una característica de mayor poder discriminatorio.

Palabras como *taquilla* tienen una ratio de aparición muy bajo, disminuyendo incluso con h&l&u, y aunque tiene una ratio específico/general muy elevado, del 100% incluso en el segundo de los casos, no será una característica demasiado relevante, provocando overfitting en el caso concreto de aparición, por lo que sería buena opción su eliminación del corpus.

A continuación se muestra la tabla comparativa con los resultados más interesantes marcados en negrita:

COMPARATIVAMENTE				
PALABRA	**RATIO APARICIONES BODY**	**RATIO APARICIONES H&L&U**	**RATIO BODY**	**RATIO H&L&U**
artist	0,49%	0,28%	11,23%	1,96%
compañ	0,59%	0,13%	4,90%	1,64%
contemporan	**0,33%**	**2,56%**	**19,02%**	**83,40%**
curs	**0,49%**	**1,54%**	**9,48%**	**2,69%**
danz	0,84%	0,27%	16,70%	9,81%
escen	**0,56%**	**0,83%**	**10,15%**	**37,79%**
espectacul	0,47%	0,04%	1,89%	0,00%
obrador	**0,56%**	**2,42%**	**100%**	**100%**
part	0,69%	1,11%	**14,38%**	**56,49%**
program	0,96%	1,68%	25,49%	15,98%
sal	**1,70%**	**5,28%**	**20,60%**	**31,32%**
taquill	**0,21%**	**0,03%**	**85,23%**	**100%**

FIGURA 3.51: Comparativa Corpus Salas Alternativas

3.9.1.7 Textos

La categoría textos es quizás más general a la hora de utilizar palabras para describir su propósito. Aún así, aventurándonos, se puede citar palabras como *biblioteca*, *libro*, *artículo*, *bibliografía* y similares, que tengan relación semántica con los textos, así como no la palabra *textos*.

Además de estas palabras se obtendrán aquellas que presenten características interesantes según los rasgos de representatividad y discriminatoriedad descritas al principio.

Los resultados se muestran en las siguientes tablas:

CORPUS: MÉTODO: body **PALABRAS DIFERENTES:** 2316 / 7.019 **PALABRAS TOTALES:** 29.182 / 676.682				
PALABRA	**Nº APARICIONES**	**RATIO**	**Nº APARICIONES TOTALES**	**RATIO**
american	85	0,29%	123	69,11%
artist	55	0,19%	1540	3,57%
autor	274	0,94%	2617	10,47%
bibliograg	61	0,21%	129	47,29%
bibliotec	1262	4,32%	1558	81,01%
busqued	222	0,76%	367	60,49%
catalog	505	1,73%	518	97,49%
cienci	163	0,56%	238	68,49%
conten	365	1,25%	1127	32,39%
histori	390	1,34%	2706	14,41%
infantil	299	1,02%	914	32,71%
investig	167	0,57%	699	23,89%
juvenil	254	0,87%	321	79,13%
lengu	249	0,85%	317	78,55%
literatur	485	1,66%	1448	33,49%
obra	753	2,58%	4239	17,39%
sign	217	0,74%	243	89,30%
tesis	171	0,59%	184	92,93%
text	102	0,35%	1504	6,78%

FIGURA 3.52: Corpus Textos obtenido por método Body

CORPUS: MÉTODO: **PALABRAS DIFERENTES:** 1131 / 2.954 **PALABRAS TOTALES:** 31.405 / 1.076.679				
PALABRA	**Nº APARICIONES**	**RATIO**	**Nº APARICIONES TOTALES**	**RATIO**
american	152	0,48%	152	100%
artist	983	3,13%	1122	87,61%
autor	577	1,84%	3678	15,69%
bibliograg	27	0,09%	28	96,43%
bibliotec	1565	4,98%	1926	81,26%
busqued	167	0,53%	183	91,26%
catalog	308	0,98%	308	100%
cienci	198	0,63%	205	96,58%

conten	327	1,04%	366	89,34%
histori	874	2,78%	1714	50,99%
infantil	391	1,25%	1358	28,79%
investig	192	0,61%	199	96,48%
juvenil	295	0,94%	298	98,99%
lengu	738	2,35%	738	100%
literatur	1298	4,13%	2626	49,43%
obra	462	1,47%	3448	13,40%
sign	1031	3,28%	1034	99,71%
tesis	580	1,85%	580	100%
text	67	0,21%	704	9,51%

FIGURA 3.53: Corpus Textos obtenido por método l&h&u

Como se ve en la tabla comparativa siguiente son muchas las características que han conseguido un incremento en ambas ratios, de las que cabe destacar principalmente *artista*, que ha conseguido pasar de 0,19% a 3,13% en el número de repeticiones y de un 3,57% a un 87,61% en el ratio específico/general, *signo* con un incremento de 0,74% a 3,28% en el número de repeticiones y de 89,30% al 99,71% en la ratio comparativa, y *lengua* con un pase de 0,85% al 2,35% y del 78,55% al 100% en apariciones / ratio comparativo respectivamente.

En cualquier caso se observa que son muchas las características que poseen gran capacidad representativa, por formar parte importante del corpus, y gran capacidad discriminatoria, por aparecer casi en exclusiva en el corpus específico de los textos. El primer problema que se puede considerar es precisamente el de ser muchas características, lo que va un poco en contra de las propiedades que debía cumplir una buena característica, pero en todo caso en la evaluación de los resultados se podrá contrastar lo aquí comentado con la observación empírica.

COMPARATIVAMENTE				
PALABRA	**RATIO APARICIONES BODY**	**RATIO APARICIONES H&L&U**	**RATIO BODY**	**RATIO H&L&U**
american	**0,29%**	**0,48%**	**69,11%**	**100%**
artist	**0,19%**	**3,13%**	**3,57%**	**87,61%**
autor	0,94%	1,84%	10,47%	15,69%
bibliograg	**0,21%**	**0,09%**	**47,29%**	**96,43%**
bibliotec	**4,32%**	**4,98%**	**81,01%**	**81,26%**
busqued	**0,76%**	**0,53%**	**60,49%**	**91,26%**
catalog	1,73%	0,98%	97,49%	100%
cienci	**0,56%**	**0,63%**	**68,49%**	**96,58%**
conten	**1,25%**	**1,04%**	**32,39%**	**89,34%**
histori	**1,34%**	**2,78%**	**14,41%**	**50,99%**
infantil	1,02%	1,25%	32,71%	28,79%
investig	0,57%	0,61%	23,89%	96,48%
juvenil	0,87%	0,94%	79,13%	98,99%
lengu	**0,85%**	**2,35%**	**78,55%**	**100%**
literatur	1,66%	4,13%	33,49%	49,43%

obra	2,58%	1,47%	17,39%	13,40%
sign	**0,74%**	**3,28%**	**89,30%**	**99,71%**
tesis	**0,59%**	**1,85%**	**92,93%**	**100%**
text	0,35%	0,21%	6,78%	9,51%

FIGURA 3.54: Comparativa Corpus Textos

3.9.1.9 Blogs, Teatros, Productoras, Actores, Autores, Bibliografía, Cartelera y Montajes

El análisis de la categoría Blogs se ha separado en un nuevo apartado pues se realizará una clasificación específica basada en unas características propias de los mismos que se describirán en dicho apartado. Pese a ello, se realizará una clasificación de los Blogs mediante todos los métodos o alternativas enumeradas en su momento para tener una línea comparativa, y en este caso se comprobará la bondad del método H&L&U frente al BoW estándar, como en los casos anteriormente analizados. En el análisis concreto de los Blogs se incluye un análisis de sus características textuales más representativas del mismo modo que se ha realizado en este apartado para el resto de categorías.

El resto de categorías enumeradas en el título forman parte de un subconjunto muy pequeño de páginas del repositorio utilizado, como se vio en la descripción del repositorio DS en el apartado 3.2.1, por lo que no se puede realizar un análisis suficiente del mismo y por tanto que pueda generalizar algunas propiedades. Así pues, se ha tomado una de estas categorías con tan pocos elementos, *Asociaciones*, para intentar realizar un análisis de la misma y comprobar los resultados alcanzados, de manera que se pueda contrastar la bondad de los diferentes métodos cuando la cantidad de datos de entrenamiento es extremadamente reducida.

3.9.2 Definición del experimento

El experimento consistirá en obtener la representación &H&L&U de las páginas del repositorio DS1 y DS2 y realizar la validación 2x2 anteriormente usada para comparar los resultados con los obtenidos por el método BoW estándar y comparar los resultados.

- La hipótesis es que la adición de características obtenidas de la meta-información de la página obtiene un incremento significativo del rendimiento de los clasificadores.

Una vez más, si a simple vista no se puede inferir dicho incremento se recurrirá al test t-student de dos colas para un nivel de significación del 95%, dónde la hipótesis nula es precisamente la contraria a la hipótesis formulada, y es que ambas series de datos tienen media similar.

Por último se adjuntarán los errores cometidos enmarcados en su intervalo de confianza del 95% para comparar la bondad de ambos métodos en cuanto a la precisión de los mismos.

En ambos casos las medidas se separarán entre los resultados para la clasificación de pertenencia a la clase y la clasificación de no-pertenencia, y se comentarán sus resultados.

3.9.3 Realización del experimento y resultados

La representación H&L&U es una representación BoW a partir del corpus obtenido desde la cabecera, los enlaces y la url del documento, previo proceso de stem y filtrado por stop word list, al igual que los anteriores.

El conjunto de características es mucho más reducido, como se vio anteriormente, se puede pasar de un corpus de unas 3000 palabras a uno de apenas 700, aunque el caso medio es reducirlo algo más de la mitad. De este modo la duplicación, en este caso triplicación, de características no supone un incremento considerable de la dimensionalidad, manteniéndose en aproximadamente la misma cantidad de atributos que las representaciones anteriores.

Las características H&L&U se obtendrán de la siguiente manera:

En primer lugar se obtiene el conjunto de html de la cabecera del documento, mediante la expresión regular siguiente:

```
<head[^>]*>((.|\r|\n|\t)*?)(</head>|<body)
```

Puesto que muchos diseñadores olvidan, quizás intencionadamente, cerrar la etiqueta head, y para evitar un timeout del parser regex, se permite el cierre de dicha etiqueta con la apertura del cuerpo del documento.

Una vez obtenidas el conjunto de palabras se procede a su separación mediante la expresión regular definida anteriormente para obtener palabras, y se crea el conjunto de características como el nombre de la misma seguido por la palabra HEAD.

Las palabras de la url se obtienen comprobando la ocurrencia de cada palabra del corpus en la misma, rellenándose con ellas las características con su mismo nombre seguidas de la palabra URL.

Para los enlaces se realiza la obtención desde dos propiedades de los documentos, concretamente de los enlaces que contiene, su texto y su url. Para ello se recorre el conjunto de enlaces de la página web y se obtiene por el primer método, el de la expresión regular, el conjunto de palabras de los mismos, y por el segundo método, el de comparar la aparición, el número de apariciones de las palabras en la URL. Ambos valores (frecuencias, al dividirse entre el número total de palabras) se suman, con una ponderación igual a uno, de modo que se toman con la misma importancia ambos, y se constituyen como una nueva característica denominada el nombre de la palabra seguida por LINK.

Con ello queda definida la estructura de la representación y el modo de obtenerla.

En las siguientes tablas se muestra el valor de las pruebas F obtenidas tanto para la pertenencia como la no-pertenencia a la clase para el clasificador basado en BoW estándar y el propuesto en el experimento basado en la meta-información de la página:

PERTENENCIA A LA CATEGORÍA	BoW std	BoW h&l&u
Asociaciones	0	**0.016**
Blogs	0.299	**0.663**
Compañías	0.660	**0.825**
Festivales	0.084	**0.740**
Formación	0.157	**0.356**
Revistas	**0.036**	0.010
Salas Alternativas	0.185	**0.406**
Textos	0.814	**0.868**

FIGURA 3.55: Prueba F en la clasificación de pertenencia BoW std vs. BoW h&l&u

No-PERTENENCIA A LA CATEGORÍA	BoW std	BoW h&l&u
Asociaciones	**0.958**	0.788
Blogs	0.707	**0.947**
Compañías	**0.706**	0.699
Festivales	0.760	**0.939**
Formación	0.761	**0.909**
Revistas	**0.959**	0.506
Salas Alternativas	0.795	**0.941**
Textos	0.991	**0.994**

FIGURA 3.56: Prueba F en la clasificación de no-pertenencia BoW std vs. BoW h&l&u

En negrita se ha mostrado una vez más los valores mayores resultantes de las clasificaciones aparejadas.

Como se puede apreciar el clasificador H&L&U es superior en la mayoría de ellas por una diferencia considerable, con lo que no es necesaria la realización del test t-student para afirmar su superioridad, pero pese a ello se realiza para comprobar hasta qué punto su resultado está alejado del esperado para un nivel de significación del 95%

Las series a comprobar son:

X1 = {0,000; 0,299; 0,660; 0,084; 0,157; 0,036; 0,185; 0,814} M=0,279; D=0,281
Y1 = {0,016; 0,663; 0,825; 0,740; 0,356; 0,010; 0,406; 0,868} M=0,486; D=0,322
D1 = {0,016; 0,364; 0,165; 0,656; 0,199; 0,026; 0,221; 0,054} M=0,213; D=0,200

Estadístico t = 2,636 > 2,365

Lo que demuestra que el clasificador H&L&U, en el caso de la pertenencia a la categoría, obtiene tasas significativamente superiores a la base BoW estándar.

X2 = {0,958; 0,707; 0,706; 0,760; 0,761; 0,959; 0,795; 0,991} M=0,833; D=0,116
Y2 = {0,788; 0,947; 0,699; 0,939; 0,909; 0,506; 0,941; 0,994} M=0,840; D=0,156
D2 = {0,197; 0,240; 0,007; 0,179; 0,148; 0,453; 0,146; 0,003} M=0,172; D=0,133

Estadístico t = 0,090 < 2,365

En cambio no sucede lo mismo cuando no pertenece a la categoría, donde el test muestra que ambos resultados son similares.

Como se puede apreciar en los resultados anteriores es en todas las categorías excepto asociaciones y revistas dónde la representación H&L&U obtiene tasas superiores que su comparativa BoW estándar.

Como se recordará, en el punto 3.1.5 se indicaba los posibles riesgos que podrían aparecer con estas dos categorías debido a la pequeña representatividad de dichas páginas frente al total, menores del 0,6% y 1,6% respectivamente.

Si se repite el test t-student excluyendo dichas categorías se aprecia lo siguiente:

Las series a comprobar serían entonces, para la pertenencia a la clase:

X1 = {0,299; 0,660; 0,084; 0,157; 0,185; 0,814} M=0,366; D=0,273
Y1 = {0,663; 0,825; 0,740; 0,356; 0,406; 0,868} M=0,643; D=0,197
D1 = {0,364; 0,165; 0,656; 0,199; 0,221; 0,054} M=0,276; D=0,193

Estadístico t =3,31 > 2,365

Lo que demuestra que el clasificador H&L&U, en el caso de la pertenencia a la categoría, obtiene tasas significativamente superiores a la base BoW estándar, y un estadístico superior al obtenido teniendo en cuenta las categorías eliminadas (3,31 > 2,636)

Y para la no-pertenencia a la clase:

X2 = {0,707; 0,706; 0,760; 0,761; 0,795; 0,991} M=0,787; D=0,097
Y2 = {0,947; 0,699; 0,939; 0,909; 0,941; 0,994} M=0,905; D=0,095
D2 = {0,240; 0,007; 0,179; 0,148; 0,146; 0,003} M=0,120; D=0,087

Estadístico t = 2,920 > 2,365

Cuyo estadístico también es superior al esperado, lo que indica que la clasificación H&L&U es superior en ambos casos a la estándar basada en BoW.

El análisis de los intervalos de error de cada clasificador se muestran a continuación:

PERTENENCIA A LA CATEGORÍA	BoW std	BoW H&L&U
Asociaciones	1 +- 0	0,409+-0,205
Blogs	0,256 +- 0,036	0,261+-0,037
Compañías	0,315 +- 0,013	0,221+-0,012
Festivales	0,891 +- 0,022	0,098+-0,009
Formación	0,409 +- 0,058	0,213+-0,050
Revistas	0,887 +- 0,079	0,754+-0,014
Salas Alternativas	0,391 +- 0,057	0,612+-0,057

Textos	0,044 +- 0,030	0,033+-0,026

FIGURA 3.57: Intervalo de error en la clasificación de pertenencia BoW std vs. BoW h&l&u

No-PERTENENCIA A LA CATEGORÍA	BoW std	BoW H&L&U
Asociaciones	0,078 +- 0,008	0,348+-0,014
Blogs	0,434 +- 0,015	0,068+-0,008
Compañías	0,162 +- 0,016	0,427+-0,021
Festivales	0,284 +- 0,014	0,104+-0,010
Formación	0,370 +- 0,014	0,156+-0,011
Revistas	0,067 +- 0,007	0,658+-0,014
Salas Alternativas	0,323 +- 0,014	0,090+-0,009
Textos	0,016 +- 0,004	0,011+-0,003

FIGURA 3.58: Intervalo de error en la clasificación de no-pertenencia BoW std vs. BoW h&l&u

Como se puede apreciar los intervalos obtenidos para la representación H&L&U son significativamente inferiores a los obtenidos por el método estándar, exceptuando los casos antes nombrados de Asociaciones y Revistas, y en menor caso las Compañías.

3.9.4 Conclusiones

La primera conclusión que se desprende y que cumple uno de los objetivos principales de la investigación es que la representación propuesta basada en la meta-información de las páginas, bien desde la intención del autor de comunicar información sobre el sitio, como la que se encuentra en las cabeceras, bien desde la intención del autor de estructurar el propio sitio, como la que se encuentra en la url y los enlaces, obtiene una tasa significativamente superior de rendimiento en la mayoría de los clasificadores que los existentes en la investigación actual.

Una segunda conclusión que se desprende de los datos y del análisis realizados en el punto anterior es la sensibilidad del método a la distribución y proporción de páginas de entrenamiento. Dicha sensibilidad hace que decaiga más rápidamente su rendimiento cuando la proporción de páginas es pequeña o su representatividad es baja. Ello seguramente esté debido al menor grado de información que aportan los meta-datos a partir de los cuales se construye la representación y se requiera de mayor cantidad de los mismos y sobre todo mayor calidad, para obtener un modelo entrenado con una precisión aceptable, en cuyo caso, la misma es superior a otros métodos.

3.10 CLASIFICACIÓN DE LOS BLOGS. UNA APROXIMACIÓN ESPECÍFICA

Los Blogs quizás sean una de las categorías más heterogéneas en cuanto a contenido que puedan aparecer en la red, pues pueden hablar de temas muy variados, incluso dentro de un mismo Blog.

Por ello a simple vista parece claro que una clasificación basada en contenido únicamente, mediante una representación BoW, aunque ésta esté mejorada mediante información contextual o cualquiera de los medios vistos con anterioridad, será una solución demasiado general, y los resultados mostrados en el apartado de evaluación demuestran este supuesto.

Pero también es innegable que cualquier persona a simple vista sabría determinar si una página web es o no un Blog, y eso es porque independientemente del contenido, autor o estilo del mismo, se comparten una serie de características diferenciadoras que se podrían utilizar para su clasificación.

3.10.1 Análisis de características específicas de los Blogs

Así pues, todos los Blogs, heredado quizás de sus comienzos basados en la idea de diario, dónde el autor va escribiendo reflexiones, contienen unas estructuras comunes entre sí y muy características de los mismos, y que difícilmente se dan en páginas que no sean Blogs, por lo que formalizando dichas características se tendría una representación base bastante sólida para realizar el aprendizaje.

Una de las primeras apreciaciones que podemos hacer es la de la estructura en bloques de información, denominados post, generalmente sin conexión entre ellos en cuanto a contenido, pero que ya nos dan una idea visual del concepto de diario.

Generalmente todos estos post comienzan con un encabezado, dónde se describe a modo de titular el contenido de dicho post, el tema del mismo. Este tema nos es indiferente, da igual que el post vaya sobre un festival de teatro en Mérida que sobre una nueva librería abierta en Albacete. Lo que nos da idea de comienzo de post es su resaltado frente al resto del texto y no su contenido, así como la aparición de varios de estos a lo largo del mismo.

Junto a este comienzo de post suele aparecer la fecha de publicación del mismo. La plantilla utilizada para generar los post puede ser muy diferente, así pues la fecha puede ir antes del encabezado, después, o incluso al final del post. El formato de las fechas también puede estar en formatos muy dispares, puede estar en formato corto o largo, y dentro de cada cual con el mes escrito en letra o en número. Así mismo, pese a ser de habla hispana, el formato puede ser el inglés. Pero lo que suele ser cierto es que todas las fechas que acompañan al post están en un mismo formato, lo que permite diferenciarlas de fechas que aparecen por dentro del mismo, aunque tampoco tiene por qué ser así, todas podrían llevar el mismo formato. En cualquier caso prácticamente siempre un post lleva asociada una fecha.

Una de las grandes características de Internet es la facilidad con la que se puede reunir a la gente, formar comunidades, actualmente conformando lo que se denomina la Web 2.0, dónde grandes comunidades comparten información, en diferentes formatos, pero al fin y al cabo información. Los Blogs no son distintos y forman un excelente punto de encuentro e intercambio de información. A diferencia de los foros, dónde cualquier persona abre un hilo sobre lo que quiera y otras personas contestan, hablando de lo que les place, un Blog es un sitio controlado por un usuario, su autor, que es quién determina el contenido del mismo, creando post y escribiendo sus reflexiones, sobre el tema que desee. Pero esto, sin realimentación, sin la posibilidad de comunicación, sería algo triste y pobre, como las primeras páginas personales que se creaban al surgir Internet, parte de las causas principales que provocaron el boom de las dotcom, y dónde miles de autores escribían sus páginas acerca de los temas que se les ocurriera, y la realimentación quedaba limitada, y no en todos los casos, a poner un enlace hacia un eMail. Los Blogs en cambio deben permitir comunicación y por ello, todos los post,

suelen ir acompañados de un enlace hacia comentarios del mismo, así como un contador de los comentarios que hay hechos por el momento.

Sólo las tres características anteriores son suficientemente representativas de los Blogs, por ser compartidas por todos ellos, así como suficientemente diferenciadoras del resto de tipos de páginas como para suponer una base interesante sobre la que trabajar para definir la caracterización formal de los mismos.

Pero aún existen más características, que aunque puedan estar compartidas por otros tipos de páginas, son bastante representativas, junto con las anteriores, de las páginas de tipo Blog y que pueden añadir certeza en la predicción automática de los mismos. Son las que denominamos islas de enlaces, grandes conjuntos de enlaces agrupados mediante listas hacia contenidos de otros archivos o cronologías del propio Blog, incluyendo además una serie de enlaces hacia Blogs "amigos" o considerados de interés.

Generalmente estas islas aparecen en un lateral de la página y agrupadas bajo bloques de tipo lista. Esto suele ser sensiblemente diferente a otro tipo de páginas, como las personales o las empresariales, que suelen tener maquetación mediante tablas o scripts, determinando un diseño menos secuencial y más estilo menú. En los Blogs dicha secuenciación en los enlaces es un objetivo perseguido, de manera que el usuario pueda explorar el contenido del mismo en base a una cronología o en base a un conjunto diferenciado de temas, sin necesidad de filigranas de estilo o diseño.

Es interesante además añadir una serie de características textuales tras el estudio del corpus obtenido a partir de los Blogs. Las palabras que más aparecen en la categoría Blog y que menos aparecen en el resto de categorías, es decir, tienen una ratio bastante cercano al 100%, son las palabras *blog*, *post* y *rss / atom*. Así mismo la palabra *comentario* (y *comment* en inglés) también tiene una frecuencia de aparición elevada, como se puede intuir de la tercera característica descrita para los Blogs, el feedback.

Todo esto se puede apreciar en las siguientes tablas que muestran el número de palabras del corpus específico utilizado, en este caso el de Blogs, la técnica con la que fue creado, mediante el cuerpo del documento o mediante h&l&u, la cabecera, los enlaces y la url, el número de veces que aparece repetida la palabra en el cuerpo actual y la ratio frente al número total de repeticiones de las palabras del corpus Blogs, el número de veces que aparece dicha palabra repetida en el corpus general obtenido del mismo modo y el ratio frente al número total de repeticiones de la palabra en el corpus Blog frente al corpus general:

CORPUS: Blogs **MÉTODO:** Body **PALABRAS DIFERENTES:** 4516 / 7019 **PALABRAS TOTALES:** 152806 / 676682				
PALABRA	**Nº APARICIONES**	**RATIO**	**Nº APARICIONES TOTALES**	**RATIO**
coment	2492	1,63%	2925	85,20%
blog	204	0,13%	214	95,33%
post	732	0,48%	749	97,77%
rss/atom	793	0,52%	801	99,00%

FIGURA 3.59: Corpus Blogs obtenido por método Body

CORPUS: Blogs **MÉTODO:** h&l&u **PALABRAS DIFERENTES:** 1389 / 2954 palabras **PALABRAS TOTALES:** 230343 / 1076679				
PALABRA	**Nº APARICIONES**	**RATIO**	**Nº APARICIONES TOTALES**	**RATIO**
coment	3669	1,59%	3878	94,61%
blog	8814	3,83%	8890	99,15%
post	2775	1,20%	2778	99,89%
rss/atom	2048	0,89%	8963	22,85%

FIGURA 3.60: Corpus Blogs obtenido por método l&h&u

Las palabras *blog* aparecen frecuentemente en la url, así como en los enlaces al propio dominio (el propio Blog), así como a Blogs "amigo". Es frecuente que los Blogs estén bajo Urls del tipo http://midominio/loquesea/blog, o bajo dominios proveedores de blogs como weblog, blogger o blogspot.

Así pues, una página bajo un dominio de este tipo tendrá dicha palabra en la url, pero además la tendrá muchas veces en cada uno de los enlaces internos. Por otro lado, páginas que no pertenezcan a dominios de este tipo, mediante los enlaces a dominios externos, generalmente también tendrán un elevado número de apariciones de la palabra Blog en los enlaces, por la misma razón anterior.

Es por tanto una característica adicional a considerar para maximizar la tasa de aciertos. Por otro lado, por la ratio de aparición que se muestra en los datos anteriores, un 95,33% y 99,15%, no es una palabra que aparezca demasiado en otros tipos de páginas, lo que le da la propiedad de ser bastante discriminante, ya que casi prácticamente siempre que aparece lo hace en una página de tipo Blog.

La palabra *post* es muy común. Los autores suelen denominar a sus propias entradas de este modo, haciéndolo explícito en muchos comentarios. Así mismo, se suele incluir automáticamente en muchos generadores de Blogs un comentario del tipo *posted by,* lo que mediante un correcto tratamiento lingüístico aporta la información pretendida. Exceptuando en las páginas de tipo foro la palabra post tiene un bajo porcentaje de aparición, como se ve en los datos de las tablas anteriores, un 97,77% y 99,89%, lo que al igual que la anterior, dota de gran capacidad discriminatoria, y por tanto aportará gran información a la clasificación.

Por último hablar de *rss/atom* [Microsiervos], que aunque no es una característica especial de los Blogs, sí que es algo lo suficientemente utilizado en ellos como para aportar cierta información.

Rss, así como atom y otros formatos, no son más que ficheros xml que contienen un resumen de lo publicado en el sitio, de modo que leyendo este único fichero se podría conocer el contenido del sitio, así como si fue modificado y cuándo, sin necesidad de recorrer sus páginas y leer sus publicaciones.

Pero por sus propiedades rss/atom no son características propias de los Blogs. Toda aquella página que sirva noticias, así como los foros, puede permitir una suscripción a las mismas mediante este protocolo, de manera que podría darse el caso de páginas que no sean Blogs que tengan este tipo de suscripciones, como por ejemplo la del Ayuntamiento de Fuengirola (http://www.fuengirola.es)

Por otro lado, no todos los Blogs tienen por qué permitir la suscripción de sus contenidos mediante este formato, por lo que no es una característica determinante de los mismos. Por ejemplo el Blog teaterías no lo contiene (http://www.teatrerias.blogspot.com/).

Pese a ello, y por la frecuencia de aparición de estas palabras en estos documentos, tal y como se muestra en la tabla, 99% y 22,5% (en el segundo caso es porque dicha palabra no suele aparecer bajo enlaces a no ser que aparezca en el alt de su logotipo, característica que no se ha explorado), podría ser una aportación interesante a la clasificación.

Otras características susceptibles de explotar podrían ser las siguientes:

- Una secuenciación cronológica en las fechas, mostrando su carácter de diario
- Enlaces hacia el desarrollo de los contenidos, a modo de titulares de noticias que se despliegan en su descripción completa
- Existencia de marcos publicitarios, dependientes del proveedor del Blog.

3.10.2 Definición del experimento

El experimento consiste en explotar todas las características analizadas en el punto anterior y comprobar su adecuación al problema de la clasificación de los Blogs realizando una comparación con la línea base de la BoW estándar, el BoW mejorado, el BoW de las Urls y el BoW de la meta-información H&L&U.

Para ello se extraerá las características que formarán parte de su representación y se entrenará a los modelos, validándolos mediante el método 2x2 descrito anteriormente y comparándolo con los resultados obtenidos por los otros métodos, mediante comparación de sus pruebas F.

Así mismo se compararán los intervalos de error de ambos clasificadores para comparar la bondad de los mismos desde el punto de vista de la precisión únicamente.

- La hipótesis de partida es que la representación propuesta alcanza unos valores muy superiores a las representaciones existentes hasta el momento, incluida la propuesta en el experimento anterior basada en la meta-información H&L&U

3.10.3 Realización del experimento y resultados

Para demostrar esta hipótesis lo primero a realizar es definir el modo de extraer las características para el entrenamiento de los modelos.

El diseño de las características que formarán parte de la representación específica de los Blogs viene directamente determinado por su análisis. Quizás más interesante será cómo obtener ciertas características.

En un principio el análisis realizado para la representación de los Blogs parecería indicar la comparación de plantillas, determinando si se adecua o no a la estructura dada.

En realidad determinar esta comprobación sería una tarea compleja ya que, aunque todos los Blogs comparten más o menos estas características, las formas de combinarlas pueden ser muchas y muy variadas.

Por ello se obtiene una representación basándose en la frecuencia de aparición de determinados elementos así como en ratios de aparición de unos frente a los otros.

En primer lugar es interesante enumerar las características a obtener, 15 en total, para describirlas y ver qué se necesita para obtenerlas. En segundo lugar se diseñará el modo de obtenerlas y en tercer lugar la manera en que algunas de ellas se componen mediante la combinación con otras.

Las características son:

- NBLOGINURL: Número de veces, o mejor dicho frecuencia de aparición de la palabra Blog en la URL. Para obtenerlo se comprueba el número de apariciones de la palabra a lo largo de la URL y se divide entre el número total de palabras de que consta la misma.

- NBLOG: Frecuencia de aparición de la palabra Blog en el documento. Para ello se obtiene, mediante la expresión regular de las palabras, las palabras que componen el documento y se obtiene la frecuencia de la palabra blog en las mismas.

- NPOST: Lo mismo que lo anterior pero con la palabra POST

- NRSS: Lo mismo pero con las palabras RSS y ATOM. La aparición de ambas se suma; suelen ser disjuntas por lo que si aparece una no aparecerá la otra, no alterando de este modo la frecuencia.

- CommentsVsDates: Ratio entre los comentarios aparecidos en la página y las fechas aparecidas. Para ello se hace uso de las siguientes expresiones regulares, para obtener los comentarios, en diferentes formatos y lenguas, y las fechas, también en diferentes formatos, cortos y largos, así como de localización:

  Para los comentarios:

```
(?<comment>((\[?[0-9]*\\]?
*(comments|comentarios))|((comment|comentario) *\[?[0-9]*\]?)))
```

  Para las fechas:

```
// week day, month day, year
(?<date>(monday|tuesday|wednesday|thursday|friday|saturday|sunday)?,?
*(january|february|march|april|may|june|july|august|september|october|
november|december) *(0?[1-9]|[12][0-9]|3[01]),? *(19|20)?\\d\\d)

?<date>(monday|tuesday|wednesday|thursday|friday|saturday|sunday)?,?
*(jan|feb|mar|apr|may|jun|jul|aug|sep|oct|nov|dec) *(0?[1-9]|[12][0-
9]|3[01]),? *(19|20)?\\d\\d)

// día de mes de año
```

```
(?<date>(0?[1-9]|[12][0-9]|3[01]) *(-|de)?
*(enero|febrero|marzo|abril|mayo|junio|julio|agosto|septiembre|octubre
|noviembre|diciembre) *(-|de)? *(19|20)?\\d\\d)

(?<date>(0?[1-9]|[12][0-9]|3[01]) *(-|de)?
*(ene|feb|mar|abr|may|jun|jul|ago|sep|oct|nov|dic) *(-|de)?
*(19|20)?\\d\\d)

// DD/MM/AA
(?<date>(0?[1-9]|[12][0-9]|3[01])(/|-)(0[1-9]|1[1|2])(/|-
)(19|20)?\\d\\d)

// MM/DD/YY
(?<date>(0[1-9]|1[1|2])(/|-)(0?[1-9]|[12][0-9]|3[01])(/|-
)(19|20)?\\d\\d)
```

- LinkCommentsVsDates: Ratio entre los comentarios, obtenidos mediante la expresión regular anterior, que forman parte de un enlace y las fechas, obtenidas también mediante las expresiones regulares anteriores.

- LinkCommentsVsComments: Ratio entre los comentarios que forman parte de un enlace y el total de comentarios, obtenidos ambos mediante las expresiones regulares anteriores.

- CommentsRatio: Ratio de aparición de comentarios frente al número de encabezados de posts.

  Esta característica, al igual que las dos siguientes, se obtiene por comparación con los tres tipos de encabezados que suelen aparecer en los blogs, H1, H2 y H3, realizando la ratio con aquél de ellos cuya cantidad de apariciones sea más cercana al de la característica a emparejar, es decir, si se obtienen las tres ratios posibles, nos quedamos con el más cercano a uno.

  Esto es así porque en cualquier Blog suelen haber tres tipos de encabezados, el del Blog en general, el de cada post y el de otros menús como las islas de enlaces, u otros resaltados. Pero esto puede variar, con lo que la aplicación de la ratio puede servir de más ayuda a un modelo probabilístico de aprendizaje como el Naïve Bayes.

- LinkCommentsRatio: Ratio de aparición de comentarios en enlaces frente al número de encabezados de posts.

- DatesRatio: Ratio de aparición de fechas frente al número de encabezados de posts.

- HasIslands: Indica la aparición de islas de enlaces. Para ello hace uso de la siguiente expresión regular, que si devuelve alguna coincidencia indicará un valor positivo, 1, y en caso contrario negativo, 0.

  ```
  <ul[^>]*>((.|\r|\n|\t)*?)</ul>
  ```

  A partir de los enlaces aquí obtenidos se calculan las siguientes características

- NILinksRatio: Ratio entre el número de enlaces hacia el propio dominio aparecidos en las islas de enlaces frente al número total de enlaces de la página

- NOLinksRatio: Ratio entre el número de enlaces hacia fuera del dominio aparecidos en las islas de enlaces frente al número total de enlaces de la página

- NIOLinksRatio: Ratio entre el número de enlaces dentro y fuera del dominio aparecidos en islas de enlaces frente al número total de enlaces de la página

Una vez definidas las características que formarán parte de la representación, así como el modo de obtenerlas, se procede al entrenamiento y validación de los clasificadores para obtener la comparativa con el resto de métodos de representación.

Los resultados obtenidos para el test 2x2 se muestran comparativamente en las siguientes tablas, separando la pertenencia y la no-pertenencia a la categoría Blogs:

	BoW std	**BoW improv**	**BoW url**	**BoW h&l&u**	**Blogs specifics**
Blogs	0.299	0.429	0.558	0.663	**0.920**
No Blogs	0.707	0.889	0.918	0.947	**0,989**

FIGURA 3.61: Estadístico F en la clasificación BoW specifics vs. el resto

Que como se puede apreciar, para ambos casos, el resultado de la prueba F es superior para la clasificación propuesta que para las comparativas, incluso superior a la propuesta en el experimento anterior.

Los intervalos de error cometidos en esta evaluación 2x2 se muestran a continuación:

	BoW std	**BoW improv**	**BoW url**	**BoW h&l&u**	**Blogs specifics**
Blogs	0,256 +- 0,036	0,399 +- 0,041	0,241 +- 0,036	0,261 +- 0,037	**0,022 +- 0,012**
No Blogs	0,434 +- 0,015	0,158 +- 0,011	0,125 +- 0,010	0,068 +- 0,008	**0,020 +- 0,004**
TOTAL	0,413 +- 0,014	0,186 +- 0,011	0,138 +- 0,010	0,091 +- 0,008	**0,020 +- 0,004**

FIGURA 3.62: Intervalo de error en la clasificación BoW specifics vs. el resto

Donde se aprecia claramente un descenso significativo de la tasa de errores desde los clasificadores comparados.

En las siguientes tablas se muestran los resultados para la prueba F y para los intervalos de error cometidos en las validaciones de los repositorios divididos que dan lugar a la validación 2x2 anterior, de manera que se pueda comprobar la influencia en los resultados de la baja y alta proporción de páginas de tipo Blog en el entrenamiento.

El repositorio DS1 consta de 2184 páginas de las cuales el 19,9% pertenece a páginas de tipo Blog, concretamente 435 páginas. El repositorio DS2 consta de 2603 páginas, de las cuales el 4,5% son de tipo Blog, 116 páginas en total.

Analizando estos dos casos extremos, dónde se realiza la validación de una partición con la otra, los resultados obtenidos muestran lo siguiente, para el entrenamiento con DS1 y la prueba con DS2:

DS1/DS2	BoW std	BoW improv	BoW url	BoW h&l&u	Blogs specifics
Blogs	0,037	0,035	0,707	0,143	**0,997**
No Blogs	0,808	0,892	**0,981**	0,978	0,937

FIGURA 3.63: Estadístico F en la clasificación BoW specifics vs. el resto validación DS1/DS2

Lo que muestra que el método específico es superior en la detección cuando pertenece a la clase de manera significativa respecto al resto de métodos, siendo el basado en url el que más se acerca, no así en el caso de no-pertenencia a la clase, dónde el método basado en url es ligeramente superior, así como el H&L&U, aunque en todos los casos superiores al 0,900

A continuación se muestran los intervalos de error dónde claramente se aprecia que el resultado general es superior en el caso de la representación específica, aunque en el caso de pertenencia a la clase el método url es superior, casi por la mitad en la tasa de errores, aunque se ve compensado por la no-pertenencia, dónde el clasificador específico es muy superior al otro.

Blogs	0,862 +- 0,063	0,922 +- 0,049	**0,052 +- 0,040**	0,922 +- 0,049	0,103 +- 0,055
No Blogs	0,295 +- 0,018	0,161 +- 0,014	0,034 +- 0,007	0 +- 0,001	**0,000 +- 0,001**
TODO	0,320 +- 0,018	0,195 +- 0,015	0,035 +- 0,007	0,043 +- 0,008	**0,005 +- 0,003**

FIGURA 3.64: Intervalo de error en la clasificación BoW specifics vs. el resto validación DS1/DS2

Y en el caso inverso de entrenamiento con DS2 y la prueba con DS1:

DS2/DS1	BoW std	BoW improv	BoW url	BoW h&l&u	Blogs specifics
Blogs	0,417	0,63	0,519	0,722	**0,916**
No Blogs	0,519	0,885	0,821	0,892	**0,976**

FIGURA 3.65: Estadístico F en la clasificación BoW specifics vs. el resto validación DS2/DS1

Se aprecia que en el caso de la pertenencia y no-pertenencia a la clase el método específico es muy superior a los demás, añadiendo en el peor de los casos valores de F superiores a 20 puntos cuando pertenece a la clase y de cerca de 10 puntos cuando no.

Blogs	0,094 +- 0,027	0,260 +- 0,041	0,292 +- 0,043	0,085 +- 0,026	**0 +0**
No Blogs	0,641 +- 0,023	0,154 +- 0,017	0,253 +- 0,020	0,173 +- 0,019	**0,041 +- 0,009**
TODO	0,527 +- 0,021	0,176 +- 0,016	0,261 +- 0,018	0,154 +- 0,016	**0,037 +- 0,008**

FIGURA 3.66: Intervalo de error en la clasificación BoW specifics vs. el resto validación DS1/DS2

Los resultados para las tasas de errores y su intervalo muestran unos valores muy superiores al triple en la mayoría de los casos, estando en el caso de la clasificación específica por debajo del 5% y con un intervalo muy reducido de menos de un 1%

3.10.4 Conclusiones

Los primeros resultados arrojan unos valores muy favorecedores para la clasificación específica de los Blogs propuesta, siendo en ambos casos, pertenencia y no-pertenencia, superior el valor tanto de la prueba F como el error cometido y su intervalo de confianza.

Como se puede apreciar, el clasificador específico consigue mejores valores en la clasificación correcta cuando pertenece a la clase Blog que cuando no, en proporción al resto de clasificaciones. Valores de F de 920 puntos frente a 663, 558, 429 o 299 obtenidos por el resto de representaciones muestran un incremento de más de 300 puntos de dicho estadístico, y encima un valor muy cercano al máximo de 1000 puntos.

Así mismo, la reducción del error cometido hasta aproximadamente un 2% y con un balance bastante equilibrado entre ambas clasificaciones (pertenencia y no-pertenencia) mejoran sensiblemente los alcanzados por el método propuesto basado en la meta-información que obtenían una tasa de error de un 9,1% aproximadamente, y que en cualquier caso mejoran considerablemente los resultados obtenidos por los métodos del estado del arte, que en el mejor de los casos obtienen una tasa de error del 13,8%

Analizando los casos más extremos se comprueba un resultado inesperado quizás a priori, y es que todos los clasificadores, a excepción del específico y el basado en Urls, obtienen mejores tasas tanto en la prueba F como en el margen de error cuando el entrenamiento se realiza con el repositorio que menor proporción de páginas de tipo Blog contienen, el DS2, y sobre todo para la clasificación afirmativa de pertenencia a la categoría.

Esto puede estar debido a dos motivos, por un lado a la proporción de palabras perteneciente al corpus de unas y otras páginas. Es posible que el repositorio con menos representatividad de Blogs en cuanto a número, sea más representativo en cuanto a calidad para el aprendizaje. También puede estar debido a que un repositorio con mayor número de páginas del tipo Blog como el DS1 contenga un conjunto mayor de palabras que lo definan, y puesto que los Blogs tratan de temas muy dispares, las características obtenidas no sean suficientemente representativas y se combinen más cerca del resto de categorías.

En cualquier caso la evaluación del experimento es que la representación específica de los Blogs alcanza mejores resultados que las demás, independientemente del conjunto de entrenamiento y validación utilizado, con lo que consigue el objetivo de obtener una representación específica capaz de superar la clasificación de este tipo de páginas respecto a las habidas en el estado del arte.

3.11 EVALUACIÓN DE LOS BLOGS EN UN DOMINIO DISTINTO AL TEATRO

Los resultados del experimento anterior muestran un incremento significativo en el rendimiento del clasificador específico de los Blogs frente a la clasificación basada en los métodos de la investigación actual.

Pero es necesario ir más allá y comprobar realmente la bondad del método realizando un test exhaustivo con un repositorio externo al facilitado, obtenido de diversas fuentes de manera que sea lo más disperso posible y no limitado o ajustado al ámbito del teatro.

3.11.1 Definición del experimento

Para realizar esta comprobación se hace uso del repositorio extendido descrito en el apartado 3.1.3 que consta de 9158 páginas de las cuales 3696 pertenecen a la

categoría Blog, obtenidas a partir del crawl hasta el tercer nivel de 42 páginas Blogs en diferentes idiomas, y de 5462 páginas de tipo no Blog obtenidas a partir de un crawl hasta el nivel 5 del directorio de Yahoo!

El experimento consiste en obtener un clasificador específico para los Blogs a partir del repositorio completo DS y evaluarlo mediante este repositorio extendido DSE, comprobando que la bondad del mismo se mantiene respecto a los datos obtenidos en el apartado anterior, para lo cual se compara el valor de los estadísticos F y los intervalos de confianza del error.

- La hipótesis de partida es que el clasificador específico obtiene tasas similares tanto para las pruebas F como para los intervalos de error que las obtenidas en los experimentos anteriores, validando de este modo su representatividad y adecuación a la clasificación en esta categoría independientemente del dominio al que se aplique, siendo por tanto generalizable y extensible a la clasificación general de los Blogs en la Web.

3.11.2 Realización del experimento y resultados

Para la realización del experimento se realiza el entrenamiento con la representación específica obtenida de todo el conjunto del repositorio DS, dónde existen tanto ejemplos positivos como negativos, y se valida con la representación específica obtenida del repositorio extendido DSE, obteniendo los siguientes resultados para el estadístico F, y que se muestran de manera completa en el Anexo IV, punto 10:

	Blogs Specific	Blogs Specific Exhaustivo
Blogs	0.920	0,913
No Blogs	0,989	0,930

FIGURA 3.67: Prueba F en la clasificación BoW specifics con DSE

Como se puede apreciar los valores no son muy inferiores a los obtenidos en el anterior experimento controlado y en ningún caso son inferiores a los 90 puntos del estadístico F.

Los intervalos de error cometidos en esta evaluación 2x2 se muestran a continuación:

	Blogs specifics	Blogs specifics exhaustivo
Blogs	0,022 +- 0,012	0,009 +- 0,003
No Blogs	0,020 +- 0,004	0,126 +- 0,009
TOTAL	0,020 +- 0,004	0,078 +- 0,003

FIGURA 3.68: Intervalo de error en la clasificación BoW specifics con DSE

En ellos se aprecia que hay un ligero aumento en el caso de la no-pertenencia a la categoría, para lo que es interesante mostrar la matriz de contingencia resultante para poder analizar los resultados anteriores:

4726	679
33	3720

FIGURA 3.69: Matriz de contingencia en la clasificación BoW specifics con DSE

Donde claramente se observa el aumento del número de falsos positivos cuando se clasifica la página como Blog sin que lo sea.

3.11.3 Conclusiones

El objetivo principal de este último experimento era el de comprobar la adecuación del modelo representado para generalizar sus propiedades y sus resultados a un dominio diferente del caso de estudio del teatro.

Para ello se ha diseñado el experimento sobre páginas de muy diferentes dominios como la informática, la cocina, las personales o la música, tal y como se comprueba en el diseño de la colección de pruebas del punto 3.SS

De los resultados anteriores se desprende que la clasificación específica no pierde apenas rendimiento cuando se aplica a un conjunto de pruebas más general. Se puede comprobar cómo el valor del estadístico F se mantiene superior al 0,900 y en ambos casos no desciende más que 0,007 y 0,059 puntos.

A partir de los errores y su intervalo de confianza se aprecia que la clasificación cuando pertenece a la categoría es incluso más ajustada que la obtenida en la clasificación 2x2 del experimento anterior, no así la obtenida para la clasificación como no-pertenencia a la clase, que aumenta desde un 2% a un 12,6%.

Comprobando la matriz de contingencia se aprecia que de las 3753 páginas referentes a Blogs que forman parte de la evaluación, se han clasificado mal únicamente 33, lo que da esa tasa de error tan reducida. En cambio, de las 5405 páginas que no son Blogs, se ha cometido error en 679, un porcentaje del 12,6% como muestra el intervalo de error, y es el valor que afecta disminuyendo ambos estadísticos F, ya que perjudica a la precisión de uno y al alcance del otro.

Lo anterior se debe principalmente a la aparición de páginas y grupos de noticias que permiten a su vez cierto feedback, de manera que el método de clasificación se confunde clasificándolas como Blogs.

Realmente, atendiendo a las características propuestas en el método, las páginas serían correctamente clasificadas pues cumplen las mismas en una proporción similar a los Blogs, aunque una persona que las visitase sabría que no se refieren a Blogs sino a noticias.

Es por tanto que aunque queda demostrada la validez y superioridad del método para la clasificación de páginas de tipo Blog, es necesario realizar ciertos trabajos futuros para asegurar la correcta separación entre estos dos tipos de páginas de manera que no se llegue a confusión y la clasificación alcance cotas si bien no superiores, sí más ajustadas en ambos casos.

Con este experimento quedan sentadas las bases para un perfeccionamiento de la representación capaz de generalizar sus resultados a cualquier ámbito o dominio de la Web en lo que se refiere a la clasificación de los Blogs, cumpliendo la segunda parte del último objetivo que así se fijó.

3.12 RECAPITULACIÓN DE LOS EXPERIMENTOS

A lo largo del capítulo se han realizado una serie de experimentos orientados a conseguir de manera progresiva cumplir los objetivos planteados, de manera que la realización de varios experimentos supone la consecución de uno o varios objetivos.

Recordando un poco aquellos se comenzaba con la necesidad de crear y determinar la validez de un repositorio de datos inicial sobre el cual realizar los experimentos de clasificación, para lo cuál era necesario definir el repositorio y experimentar sobre el mismo para comprobar su adecuación al problema, obteniendo clasificadores de una calidad mínima que permitiera su comparación, así como su relación con los métodos de evaluación, de manera que la comparación se realizase sobre una base sólida de evaluación de los modelos, lo que liga con el segundo objetivo, que es la determinación de dicho marco de evaluación.

Para la consecución de ambos objetivos se propone un conjunto de sitios Webs previamente clasificados, mediante ciertas anotaciones que enlazan cada página o páginas de un sitio con una o más categorías, y se expande dicho repositorio mediante un crawl del mismo.

El primer experimento consiste en comprobar los efectos de dicha expansión mediante el crawl sobre la calidad de los modelos, y determinar si es necesaria a partir de la comparación con el repositorio de anotaciones, y cuyas conclusiones muestran que sí es necesaria dicha expansión para obtener clasificadores con cierto grado de calidad, ya que un repositorio muy limitado como el de anotaciones resulta en modelos que en la mayoría de casos no consigue un rendimiento superior al nulo.

Pero con esta expansión surge la duda de la posible implicación en la evaluación de los modelos, para lo que se diseña el segundo experimento consistente en comparar los efectos de la expansión de las Urls en los métodos clásicos de evaluación cruzada, demostrándose que se obtienen tasas muy superiores a una evaluación más ajustada a la realidad como la evaluación propuesta 2x2 y que parte de la idea propuesta por [Dietterich] en su evaluación 5x2.

Este segundo experimento nos valida tanto el repositorio de datos para las tareas de clasificación, cumpliendo junto con el anterior experimento el primer objetivo, como nos determina un marco de evaluación sostenible sobre el que realizar las comparativas entre métodos, cumpliendo con ello el segundo objetivo.

El tercer objetivo consistía en comprobar la importancia del preprocesado de los datos y su implicación en la evaluación de los modelos, para lo cual se han realizados sendos experimentos de tratamiento lingüístico mediante stemming y selección de atributos mediante la elección de corpus.

En ambos experimentos se concluye que no son determinantes para la calidad de los modelos al menos en el sentido de mejorarlos, y aunque en uno de ellos, el de la selección de corpus, se determina que empeora la clasificación general, no así influyen en la comparación entre los mismos, pues todos partirán de dicha representación, y en cualquier caso la disminución del rendimiento responde a márgenes poco apreciables.

Pero por otro lado sí que se determina la necesidad de ambos métodos para reducir considerablemente la dimensionalidad, de manera que los modelos resultantes sean mucho más manejables y el modo de obtenerlos menos costosos en el ámbito temporal y espacial.

Lo anterior marca la consecución del tercer objetivo, que junto con los anteriores, da un marco básico sobre el que realizar la comparación entre métodos.

El cuarto objetivo pretende enmarcar la investigación actual en clasificación web realizando una comparativa de los métodos explorados en la misma de manera que se intenta mejorar la clásica basada en la representación de bolsa de palabras de su contenido.

Para ello se definen dos experimentos dónde se compara la representación base estándar con una representación mejorada a base de información contextual extraída de las etiquetas en un primer experimento, y con una representación basada en las palabras contenidas en la url de los documentos a clasificar, en un segundo experimento.

En ambos experimentos se concluye, mediante métodos estadísticos, que la mejora de manera general no es significativa, lo que indica que las clasificaciones propuestas no son significativamente mejores en el caso general de la clasificación web que nos ocupa que su línea base BoW estándar.

Así mismo el segundo experimento concluye además que la variabilidad de sus resultados es tal que en algunos casos, categorías concretas, obtiene unos resultados sorprendentemente superiores a la línea comparativa base, lo que aporta la idea de que podría servir de mejora a una clasificación más adecuada y estable, mejorando los resultados en ciertos casos concretos, pero manteniéndose en una línea general no tan variable.

Con ambos experimentos se concluye el cuarto objetivo y junto con los anteriores da paso a la búsqueda de una representación capaz de superar las habidas en la investigación en la mayoría de los casos por un tanto significativamente apreciable, lo que conforma el quinto objetivo.

Para la consecución del quinto objetivo se investiga en las características que aparecen en la meta-información que aportan las páginas y se define un experimento en el que se exploran las mismas y se comparan sus resultados con la investigación base.

La conclusión de este experimento es que es en líneas generales significativamente superior a los existentes en la investigación actual, siempre y cuando se eliminen del aprendizaje las clases que contengan una baja representatividad de páginas, pues es más sensible a estas situaciones que aquéllos.

El penúltimo objetivo de la investigación era determinar una representación específica para un tipo concreto de páginas, los Blogs, que fuera fuertemente unida a los mismos y que consiguiese unas tasas significativamente superiores a todos los métodos anteriores.

Para ello se han definido los dos últimos experimentos, en el primero del cual se describe qué características definen a los Blogs, cómo se obtienen dichas características y construye un modelo a partir de las mismas, determinando no sólo que su clasificación general es significativamente superior sino que lo es por una diferencia bastante elevada tanto en estadístico F como en la tasa de errores cometidas y el intervalo en el que se enmarcan.

En este penúltimo experimento se deja entrever un problema, y necesidad de trabajo futuro, que con el último experimento se confirmará. En este último experimento se realiza la evaluación de un conjunto superior de datos, de un dominio diferente al específico del teatro, extraídos de manera controlada desde Internet y se comprueba que la bondad de la representación se mantiene similar a la obtenida en el anterior experimento, concluyendo de este modo con la consecución del sexto objetivo de la investigación.

Pero en este experimento se aprecia lo que en el anterior se dejaba entrever, y es el número de errores cometidos con ciertas páginas que aún no siendo Blogs, tienen características muy similares a estos, tales son los grupos de noticias. Para evitar este error será necesaria una investigación futura que determine lazos de unión y sobre todo de separación entre ambas categorías para evitar clasificaciones erróneas, que por otro lado, están dentro de unos márgenes bastante aceptables en comparación con el resto de las técnicas estudiadas.

Este último experimento a su vez cumple el último objetivo dónde se marca la necesidad de hacer extensibles los resultados de manera que sean generalizables fuera del dominio concreto de estudio, lo que con los resultados obtenidos queda demostrado y abre una línea de investigación futura para su perfeccionamiento.

CAPITULO 4 CONCLUSIONES

4.1 DETERMINACIÓN DE UN MARCO IDÓNEO DE PRUEBAS Y EVALUACIÓN

La investigación se ha centrado en la clasificación de páginas web en un dominio específico mediante un caso de estudio en el ámbito del teatro. Para ello se ha creado una colección de pruebas y se ha determinado, mediante diversos análisis estadísticos, la idoneidad de la misma.

Con los primeros experimentos ha quedado demostrada la idoneidad del repositorio creado, así como se ha fijado el marco de evaluación a utilizar en la comparativa entre modelos.

Todo proyecto de minería de datos debe tener una etapa previa de preprocesado de los datos para determinar la validez de los mismos y la adecuación al problema. En el caso de la investigación que nos ocupa, la clasificación de páginas web, esta etapa previa se orienta hacia la obtención de las características que puedan resultar de mayor interés para el aprendizaje, en este caso concreto las palabras que formarán el conjunto de atributos del modelo.

Para ello se ha experimentado sobre el corpus utilizado para la creación de los modelos y sobre técnicas lingüísticas como la aplicación del stemer de Porter, determinando que no son necesarias para aumentar la calidad de los modelos, pero sí para reducir significativamente la dimensionalidad, con lo que se ha fijado unas bases para la extracción de las mismas basadas en estas dos conclusiones.

Todas estas tareas son previas a la realización de experimentos sobre el marco de la investigación actual, de manera que se facilite la base necesaria para ello: el repositorio de pruebas, el marco de evaluación y el preprocesado de los datos.

Con los experimentos y validaciones realizadas se ha determinado la idoneidad de la colección de pruebas para las tareas comparativas y se ha creado un marco de creación y evaluación de los modelos sobre la base de la relevancia estadística.

4.2 DETERMINACIÓN DE UNA LÍNEA BASE DE COMPARACIÓN DE LAS PROPUESTAS

Una vez definido todo lo anterior se ha procedido a obtener algunas de las aproximaciones existentes en el estado del arte que mejores resultados presentan, de manera que se genera con ello una línea de métodos base sobre los que comparar las propuestas del trabajo.

Para realizarlo se han obtenido y evaluado las diferentes representaciones en el marco de prueba creado, y se han extraído conclusiones acerca de su rendimiento, variabilidad, tasas de error, etcétera, reforzando y guiando la investigación en la propuesta hacia su consecución final.

Se ha podido comprobar que la aplicación de métodos basados en bolsa de palabras no obtienen tasas demasiado significativas de rendimiento, así como que las mejoras introducidas a la misma no producen efectos significativamente mejores sobre la línea base, a excepción de la Url, cuyos resultados en algunas ocasiones puntuales son muy superiores a aquella.

4.3 CREACIÓN DE UN MODELO BASADO EN LA META-INFORMACIÓN DEL SITIO

El presente trabajo se ha enmarcado en las líneas de investigación actuales para proponer una nueva manera de caracterizar o representar los documentos que permita una mejora significativa en la clasificación Web frente a aquéllas.

Esta nueva manera que se propone se basa en obtener la meta-información acerca de la página que se encuentra en los meta-tags de la misma, lugar dónde el autor define la información que desea sobre su página, es decir, existe en esos datos una intención del autor por comunicar información acerca de la página.

Esta información equivaldría al resumen que el autor de la página daría sobre su propia página, de modo que cualquier proceso automático, como los motores de búsqueda, supieran a qué se refiere sin necesidad de analizarla completamente con métodos más complejos.

El principal problema como ya se vio es que no siempre esta información está disponible, muchos autores no rellenan esta información, por desconocimiento, pereza o intención, por lo que las páginas que así sean escritas no aportarían información para una correcta clasificación.

Por ello se añadieron ciertas características extraídas a partir de la misma idea de meta-información o información acerca de la página, y apolladas en los resultados obtenidos en los experimentos sobre las técnicas de la investigación actual, pero esta vez no sólo a partir de la intención del autor de comunicar información, sino de su intención por dotar de estructura a su sitio web, información que aparece en los enlaces de la misma, así como en su Url.

Las posibilidades que brinda el hipertexto son muchas, se puede llegar a construir un entramado tan complejo como se desee de páginas enlazadas unas con otras, pero esto tiene un problema en tiempo de diseño para el autor de las mismas, si no identifica de manera visible los enlaces entre ellas, le será difícil modificarlos en el caso que sea necesario.

Es por lo anterior por lo que pese a que las recomendaciones de la W3C indican que las Urls deben ser lo más inocuas posible y no aportar información, los escritores de páginas Web tienden a estructurar las mismas con nombres concretos que den idea de en qué orden de la jerarquía nos encontramos y hacia dónde vamos.

La adición de esta meta-información, porque al final actúa de información sobre la información proporcionada por el contenido de la página, permite compensar la falta de información de aquellas páginas cuyo autor no definiera el conjunto de meta-tags

descriptivos de la misma, y en caso que así sea, incrementar las posibilidades de un aprendizaje correcto.

Con esta representación se consigue, tal y como se ha mostrado en los resultados, tasas de rendimiento superiores a las de otras técnicas del estado del arte, aportando como principal novedad la manera en que se combina la diferente información para obtener la representación, así como la sencillez del método a la hora de obtener información equivalente a la generada por un resumen de la página, tarea compleja y motivo de investigación en otras áreas.

4.4 CREACIÓN DE UN MODELO DE REPRESENTACIÓN ESPECÍFICA DE LOS BLOGS

El trabajo va más allá proponiendo un objetivo más: conseguir una representación específica del tipo de páginas Blogs que consiga ajustarse de tal manera que su clasificación obtenga unas tasas muy elevadas de rendimiento.

Mediante la investigación realizada se ha pretendido encontrar un conjunto de características ligadas exclusivamente y de manera intrínseca a los Blogs, de modo que el aprendizaje de un modelo sobre las mismas permita la generalización con un alto grado de certeza.

Para ello se ha experimentado sobre una representación de características propias de los Blogs e independiente de su contenido, dominio e idioma, se ha obtenido un modelo con un rendimiento muy superior a todos los demás tanto dentro como fuera del dominio del teatro.

La representación específica de los blogs no se basa en ninguna característica estudiada en otros métodos de clasificación sino que propone una aproximación a los aspectos visuales del mismo desde técnicas computacionales, intentando formalizar mediante ratios estadísticos los efectos visuales que un ser humano distingue de estas páginas.

Además de novedad en su representación el método propuesto añade un alto grado de certeza en la clasificación, siendo poco dependiente del idioma de la página que se clasifica, e independiente del dominio en el que se aplique.

Un punto en contra de la representación es la obtención de un porcentaje algo más elevado de falsos positivos cuando aparecen grupos de noticias en la clasificación, y que aunque conceptualmente sean diferentes, comparten hasta tal punto las características seleccionadas que para el clasificador es difícil distinguirla de los Blogs reales, con lo que con esto se abre una línea de mejora futura imprescindible para un desarrollo correcto del modelo.

Se concluye que la investigación se ha llevado en aras a conseguir una representación más ajustada a la clasificación Web que las existentes, para lo cuál se ha definido un marco sólido sobre el que trabajar y evaluar comparativamente los modelos, y se ha realizado las comparaciones necesarias entre los mismos de manera que se ha determinado, mediante un grado de certeza estadística, la consecución de los objetivos y abriendo nuevas líneas de investigación futuras.

4.5 LÍNEAS FUTURAS DE TRABAJO

Las líneas de trabajo futuras son muchas y muy variadas y se desprenden en su mayoría directamente de las conclusiones anteriores, aunque también muchas de los resultados de los experimentos realizados:

- El objetivo principal de una investigación de este tipo es conseguir demostrar la adecuación de los resultados específicos a la extensión a otros dominios diferentes.

  La base de la clasificación propuesta estriba en la intención del autor de comunicar información sobre su sitio Web, mediante la meta-información que aparece en la cabecera de las páginas, y de generar cierta estructura lógica en el mismo, lo que se infiere a partir de la estructura de enlaces que crea.

  A lo largo de los experimentos se ha demostrado la superioridad del método en el caso concreto del dominio específico del teatro.

  La base para ello es similar a la utilizada por otras aproximaciones del estado del arte como la que se apoya en resúmenes o la que lo hace en el análisis contextual de los enlaces, pero combinando ambas ideas en una representación obtenida de manera sencilla a partir de las características hipertextuales de las páginas Web.

  Es por tanto que una línea de investigación futura sería determinar la validez de estas conclusiones cuando se refieren a otros dominios diferentes de la web, y por lo tanto la generalización del método.

  Para llevarlo a cabo se podrían definir una serie de experimentos sobre colecciones de texto previamente clasificadas. Algunas de estas colecciones, muy utilizadas en la investigación y que por tanto servirían para comparar los resultados obtenidos con otras aproximaciones a las tareas de clasificación, contienen clasificaciones de primer y segundo nivel de la Web, como el repositorio Cade (http://www.cade.com.br)

  Se consideran de primer nivel en Cade categorías como Educación, Ciencia, Computadores y similares, y de segundo nivel Biología, Química, Música, etcétera. Por lo tanto, según este modo de estructurar las colecciones, nuestra colección de pruebas pertenecería a un tercer nivel, mucho más específico que los niveles Cade.

  Cuando más profundidad existe en la tipología de las páginas, mayor es el grado de similitud entre las páginas que lo conforman, por lo que si el método propuesto ha obtenido tasas más elevadas de rendimiento que los del estado del arte, es muy probable que así sea en niveles superiores donde la separación entre las páginas es mayor, por lo que sería un buen punto de partida.

  Otros repositorios de pruebas muy utilizados como WebKB (http://www.cs.cmu.edu/~webkb/) contienen páginas recopiladas de diferentes departamentos de computación de diferentes universidades, categorizadas en 7 categorías diferentes, de manera que también podría ser un punto de validación interesante.

- La clasificación propuesta ha obtenido unos resultados significativamente superiores desde el punto de vista estadístico, pero también ha mostrado comportamientos especiales en determinados casos, cuando el conjunto de páginas de entrenamiento de una determinada categoría es muy limitado, siendo esta representación más sensible a ello.

  Para mejorarlo se podría optar por añadir algún conjunto de características que complemente las utilizadas de la cabecera y los enlaces, para lo cuál habría que investigar y experimentar de manera que se consiga el objetivo de obtener una representación más sólida a casos extremos.

- La evaluación del preprocesamiento de los datos, en concreto el experimento con el stemer, arroja unos resultados nada esperados y enfrentados a los resultados obtenidos por otros trabajos como [Arregi]

  La posibilidad de que esto haya sucedido a consecuencia de los datos utilizados para su validación es elevada. Al definirse el trabajo sobre el ámbito concreto del teatro, la relación lingüística entre sus palabras es quizás más elevada que en un dominio superior donde las categorías difieran más, por lo que no hay que abandonar la idea de mejorar el tratamiento lingüístico de los documentos.

  La necesidad de tratamiento lingüístico ha quedado latente en múltiples trabajos, y la incorporación del stemer en el trabajo actual no es más que una aproximación simple a ello.

  Sería por tanto una línea futura de trabajo que se debería explorar la de mejorar el tratamiento lingüístico mediante la incorporación de analizadores morfológicos, léxicos y sintácticos, reduciendo de este modo la dimensionalidad y ajustando semánticamente el contenido de las características a las diferentes categorías.

- En ampliación a lo anterior se hace necesaria la inclusión de algún tipo de tratamiento multilingüe.

  Una de las principales características de Internet es su multilingüismo, y lo que hace que sea extremadamente difícil la extracción de información y la clasificación.

  Muchas vías de investigación han optado por la obtención de características meramente estructurales, de manera que no sea precisa la manipulación de términos en diferentes idiomas.

  En el caso concreto de la clasificación de páginas de teatro, así como sucederá en todos los casos que sean clasificaciones dentro de una categoría superior que defina un dominio del conocimiento, como grupos de noticias sobre un tema concreto como economía (microeconomía, macroeconomía, bolsa, finanzas...), informática (robótica, programación, visión por computador, clasificación de textos...) o cualquier otra, la clasificación basada únicamente en datos estructurales no es tan precisa como la obtenida mediante su contenido, debido a que en su mayoría comparten este tipo de características.

Por lo que si se desea una clasificación multilingüe se deberá invertir tiempo en tratar de algún modo el contenido de las páginas para obtener bien una representación común traducida a algún lenguaje final o intermedio, bien generando corpus mayores que contengan varios idiomas, bien entrenando clasificadores para cada idioma.

En cualquier caso esta línea de trabajo entrará en estrecha colaboración con otras áreas de investigación como la extracción de información o la traducción automática.

- Una línea de trabajo directa para reforzar el método de clasificación propuesto sería el análisis del texto de las imágenes, sobre todo cuando se encuentran en los enlaces, aunque posiblemente la información de las mismas en cualquier caso pueda aportar información a la representación.

  El problema está una vez más en que no siempre se siguen las prácticas recomendadas de accesibilidad y estos textos alternativos no siempre están disponibles, por lo que sería necesario realizar una serie de experimentos previos para determinar si vale la pena invertir tiempo o no en la mejora por esta vía.

  Además el tamaño de las imágenes también podría aportar información, ya que las imágenes publicitarias suelen estar en unos rangos determinados de tamaño, así como muchas imágenes que suelen ser comunes como las de suscripción rss, las de cesta de la compra, las de selección de productos,...

- Respecto a la clasificación de los Blogs es necesaria la continuación de los trabajos de investigación por un lado para reforzar la representación obtenida, de modo que sea más robusta, y por otro lado para conseguir diferenciar entre los tipos blog y los tipos noticia.

  Para ello habría que investigar en alguna característica diferenciadora, seguramente de la categoría de las noticias, de manera que con un entrenamiento con ambos tipos se consiga reducir el número de falsos positivos para esta categoría.

  Así mismo habría que investigar en más idiomas y ajustar las características, o más bien el modo de obtenerlas, a nuevos formatos de fecha, comentarios y demás, de manera que la representación sea totalmente independiente del idioma.

- Por último sería necesario mejorar los aspectos técnicos de extracción de características y formulación de modelos para su utilización en aplicaciones en tiempo real, de manera que no se incorpore una latencia innecesaria y molesta.

  En cualquier caso dependerá de la orientación práctica a la que se destine, así pues para la clasificación de un directorio siempre se puede realizar un proceso previo de clasificación, como el proceso de indexación de los motores de búsqueda, y generar con ello un índice de páginas clasificadas.

  Pero en el caso de que se desee una aplicación en tiempo real, como por ejemplo un clasificador tras un buscador, o un filtro de páginas, será necesario mejorar las

tareas de clasificación, sobre todo la de obtención de la página web a clasificar que es la tarea que mayor latencia incorpora.

En cualquier caso este punto sería una línea de investigación a desarrollar en caso de llevar a producción alguno de los métodos propuestos, y no en sí una línea de investigación para mejorar los métodos.

La idea subyacente a todas estas líneas de investigación es la misma, conseguir métodos de representación de las páginas capaces de conseguir modelos de clasificación automática cada vez más robustos y precisos, de manera que se consiga, sin demasiado coste económico y de recursos humanos, dar el mayor orden posible al gran repositorio de datos que es la Web, facilitando de este modo las tareas de búsqueda y visualización de la información al destinatario e interesado final, el usuario.

BIBLIOGRAFÍA Y REFERENCIAS

Arregi, O; Fernández, I. Clasificación de Documentos Escritos en Euskara: Impacto de la Lematización. (http://ixa.si.ehu.es/Ixa/Argitalpenak/Artikuluak/1023985947/publikoak/cfd_3.pdf)

Attardi, Giuseppe; Gulli, Antonio; Sebastiani, Fabrizio. Automatic Web Page Categorization by Link and Context Analysis (http://citeseer.ist.psu.edu/rd/90024581,354923,1,0.25,Download/http://citeseer.ist.psu.edu/cache/papers/cs/7952/http:zSzzSzfaure.iei.pi.cnr.itzSz~fabriziozSzPublicationszSzTHAI99zSzTHAI99.pdf/attardi99automatic.pdf)

Barzilay, R., Elhadad N., McKeonwn K. R. Inferring Strategies for Sentence Ordering in MultiDocument News Summarization. Journal of Artificial Intelligence Research, 17:35-55, 2002

Billsus D., Pazzani, M.J. A Hybrid User Model for News Story Classification. In Proceedings of the Seventh Intl. Conference on Usr Modeling, pages 99-108. Springer-Verlag New Work, Inc., 1999

Bouckaert, R. Estimating Replicability of Classifier Learning Experiments (2004) (http://www.aicml.cs.ualberta.ca/_banff04/icml/pages/papers/61.pdf)

Calado, Pável; Cristo, Marco; Moura, Edleno; Ziviani, Nivio; Ribeiro-Neto, Berthier; Adré Goncalves, Marcos. Combining Link-based and Content-based Methods for Web Document Classification (http://portal.acm.org/citation.cfm?id=956938)

Casas Sánchez, José M.; Santos Peñas, Julián. Introducción a la Estadística para Administración y Dirección de Empresas

Chakrabarti S, Dom B, Indyk P. Enhaced Hypertext Categorization Using Hyperlinks. In Proceedings of the ACM SIGMOD International Conference on Management of Data, pages 307-318, Seattle, Washington, June 1998

Cristo, Marco; Calado, Pável; Silva de Moura, Edleno; Ziviani, Nivio; Berthier, Ribeiro-Neto. Link Information as a Similarity Measure in Web Classification (http://www.springerlink.com/content/aryan7c17m1b94ta/)

Dietterich, T. G. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms, 1998

Forman, George. An Extensive Empirical Study of Feature Selection Metrics for Text Classification, 2002 (http://www.hpl.hp.com/techreports/2002/HPL-2002-147R1.html)

Furnkranz J. Exploiting Structural Information for Text Classification on the WWW. In Intelligent Data Analysis, pages 487-498, 1999

García Pérez, Alfonso. Problemas Resueltos de Estadística Básica

Gómez Hidalgo, José María; Puertas Sanz, Enrique; Carrero García, Francisco; Buenaga Rodríguez, Manuel de. Categorización de Texto Sensible al Coste para el Filtrado de Contenidos Inapropiados en Internet (http://www.sepln.org/revistaSEPLN/revista/31/31-Pag13.pdf)

Google Blog Search
http://www.google.com/intl/es/help/about_blogsearch.html

Gabrilovich, Evgeniy; Markovith, Shaul. Text Categorization With Many Redundant Features: Using Aggressive Feature Selection to Make SVMs Competitive with C4.5. ACM International Conference Proceeding Series, 2004 (http://portal.acm.org/citation.cfm?id=1015330.1015388)

Glover E.J, Tsioutsiouliklis K., Lawrence S, Pennock, D.M., Flake G.W.. Using Web Structure for Classifying and Describing Web Pages. In Proceedings of WWW-02. International Conference on the World Wide Web, 2002

Guyon, Isabel; Eliseef, Andree. The Journal of Machine Learning Research, 2003 (http://portal.acm.org/citation.cfm?coll=GUIDE&dl=GUIDE&id=944968)

Hernández Orallo, José; Ramírez Quintana, Mª José, Ferri Ramírez, César. Introducción a la Minería de Datos

IKVM .Net
http://en.wikipedia.org/wiki/IKVM.NET
http://weka.sourceforge.net/wiki/index.php/Use_Weka_with_the_Microsoft_.NET_Framework
http://weka.sourceforge.net/wiki/index.php/IKVM_with_Weka_tutorial
http://www.onjava.com/pub/a/onjava/2004/08/18/ikvm.html
http://www.ikvm.net/userguide/tutorial.html

Joachinms T., Cristianini N., Shawe-Taylor J. Composite kernels for hypertext categorisation. In C. Broodley and A.Daniluk, editors, Proceedings of ICML-01, 18th International Conference on Machine Learning, pages 250-257, Williams College, US, 2001. Morgan Kaufmann Publishers, San Francisco, US.

Joachims. Learning to Classify Text Using Support Vector Machines. Methods, Theory and Algorithms. 2002

Kan, Min-Yen. Web Page Classification Without the Web Page (http://portal.acm.org/citation.cfm?id=1013367.1013426)

Lewis, D. Text Representation for Intelligent Text Retrieval: A Classification-Oriented View. In P. Jacobs, editor, Text-Based Intelligent Systems, Chapter 9. Lawrence Erlbaum, 1992 (http://compstat.chonbuk.ac.kr/Sisyphus/CurrentStudy/TDM/Paper/lewis92.ps)

Lindemann, Christoph; Littig, Lars. Classifying Web Sites. (http://www.cs.bell-labs.com/cm/cs/who/pfps/temp/web/www2007.org/posters/poster876.pdf)

Marckini, Fedrick. El Posicionamiento en Buscadores

MathWorld
Degree of freedom (http://mathworld.wolfram.com/DegreeofFreedom.html)
Paired t-Test: http://mathworld.wolfram.com/Pairedt-Test.html

Microsiervos RSS, Atom...
http://www.microsiervos.com/archivo/internet/que-es-rss-y-xml-rdf-atom.html

Oh H.J, Myaeng S.H., Lee M.H. A practical Hypertext Categorization Method Using Links and Incrementally Available Class Information. In Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval, pages 264-271. ACM Press, 2000

O'Neill T., Lavoie Brian F., Bennet Rick. Trends in the Evolution of the Public Web, 1998-2002. (http://www.dlib.org/dlib/april03/lavoie/04lavoie.html)

PcMagazine. Web Content Filtering
http://www.pcmag.com/article2/0,4149,1538777,00.asp

Pierre, John M**.** Practical Issues for Automated Categorization of Web Sites. Metacode Technologies, Inc. 2000
(http://www.ics.forth.gr/isl/SemWeb/proceedings/session3-3/html_version/semanticweb.html)

Porter, Martin. Stemmer de Porter, 1980 (http://tartarus.org/~martin/PorterStemmer/)
(http://tartarus.org/~martin/PorterStemmer/def.txt)

Rangel Pardo, Francisco Manuel. Minería de Datos Distribuida. 2007
(http://www.frandzi.corex.es/PWP/frandzi/dm4distribuida.html)

Rangel Pardo, Francisco Manuel. Minería de Datos Escalable. 2007.
(http://www.frandzi.corex.es/PWP/frandzi/dm4escalable.html)

Regular Expressions
http://www.regular-expressions.info/
http://regexadvice.com/Blogs/dneimke/archive/2003/12/06/179.aspx
http://www.geekzilla.co.uk/ViewD8902276-CCD3-4B93-8D27-A7A1908D25D3.htm

SBC2. Sistemas Basados en el Conocimiento II: Introducción a la Neurocomputación. Equipo Docente SBC2 UNED
(http://www.ia.uned.es/asignaturas/sbc2/sbc2/libro/book.pdf)

Shen, Dou; Chen, Zheng; Yang, Qiang; Zeng, Hua-Jun; Zhang, Benyu; Lu, Yuchang; Ma, Wei-Ting. Web Page Classification Through Summarization
(http://portal.acm.org/citation.cfm?id=1008992.1009035)

Shih, L.K; Karger, D.R. Using Urls and Table Layout for Web Classification Tasks (http://portal.acm.org/citation.cfm?id=988699)

Slattery S., Craven M. Discovering Test Set Regularities in Relational Domains. In P. Langley, editor, Proceedings of ICML-00, 17th International Conference on Machine Learning, pages 895-902, Stanford, US, 2000. Morgan Kaufmann Publishers, San Francisco, US.

SnowBall
http://snowball.tartarus.org/
http://snowball.tartarus.org/algorithms/spanish/stemmer.html

Sun, Aixin; Lim Ee-Peng; Ng, Wee-Keong. Web Classification Using Support Vector Machine.
(http://citeseer.ist.psu.edu/rd/90024581,668893,1,0.25,Download/http://citeseer.ist.psu.edu/cache/papers/cs/30278/http:zSzzSzwww.cais.ntu.edu.sgzSz~sunaixinzSzpaperzSzsun_widm02.pdf/sun02web.pdf)

Tabalka, Marek; Bieliková, Mária. Using Salient Words to Perform Categorization of Web Sites. Text, Speech and Dialogue: 5th International Conference, TSD 2002, Brno, Czech Republic, September 9-12, 2002. Proceedings. Páginas 130-154. (http://www.springerlink.com/content/65plpup3mwv7cnde/)

WhatIs.com Content Filtering
http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci863125,00.html

Wikipedia
Null Hipotesys : http://en.wikipedia.org/wiki/Null_hypothesis
Student's t-distribution: http://en.wikipedia.org/wiki/Student%27s_t-distribution
Student's t-Test: http://en.wikipedia.org/wiki/Student's_t-test

Zheng, Z; Wu, X, Srihari, R. Feature Selection for Text Categorization on Imbalanced Data. ACM SIGKDD Explorations Newsletter, 2004
(http://portal.acm.org/citation.cfm?id=1007741)

Programas comerciales

CYBERsitter
http://www.cybersitter.com/

GFi Web Monitor
http://www.gfisoftware.de/webmon/
http://pcwin.com/Security___Privacy/Access_Control/GFI_WebMonitor_4_for_ISA_Server/index.htm

KDnuggets Web Mining
http://www.kdnuggets.com/software/web-mining.html

ISYS Search Software
http://www.isys-search.com/

Megaputer Text Mining Technology

http://www.megaputer.com/text_mining.php

SAS Text Miner

http://www.sas.com/technologies/analytics/datamining/textminer/
http://www.sas.com/technologies/analytics/datamining/textminer/factsheet.pdf

Spector Soft

http://www.spectorsoft.com/

ANEXOS

ANEXO I: ANÁLISIS DEL SISTEMA

Para realizar la investigación, y siguiendo las pautas definidas en el análisis de alternativas, ha hecho falta implementar una serie de herramientas auxiliares para obtener los datos necesarios para ello, como las páginas sobre las que trabajar, el corpus para obtener la representación, herramientas visuales para visualizar, manipular y pretratar los datos, etcétera.

Siguiendo la metodología XP expuesta con anterioridad, esto ha ido definiendo las diferentes *releases,* o pequeños desarrollos, ajustados a las necesidades específicas del momento, y que han guiado el desarrollo global del proyecto y la consecución del sistema final.

I.1 Revisión de los requisitos

En una metodología más clásica es lo que se definiría como requisitos de la aplicación, y en este punto se definen los mismos, lo que equivaldría a cada una de las *releases* desarrolladas (aunque alguno de estos requisitos en realidad ha compuesto más de una *release*, por contener funcionalidad añadida a la base para la que fue pensado):

- Obtención de las páginas Web desde Internet a un repositorio local para trabajar de manera desconectada y más eficiente. De este modo se estructura la información y se dota de mayor facilidad de acceso, a diferencia de su estado distribuido, de modo que el tratamiento y manipulación del repositorio sea eficiente, sencillo y completo, no dependiente de la disponibilidad ni la latencia en las comunicaciones.
- Visualización, manipulación, combinación y eliminación de páginas previamente extraídas al repositorio local. Se debe poder visualizar el repositorio de páginas extraídas, obtener estadísticas de la distribución por clases de las mismas, poder eliminar páginas no válidas, dividir el repositorio en diferentes grupos, combinar páginas nuevas con páginas previamente extraídas en otros repositorios. En definitiva, una interfaz que permita el tratamiento del repositorio local.
- Generación del corpus a partir de un repositorio local de páginas Web. Debe recorrer el repositorio, extraer las palabras que lo componen, aplicar las técnicas lingüísticas necesarias para su manipulación y generar el corpus a partir de ellas.
- Manipulación y comparación de corpus para un preprocesado y ajuste del mismo a las necesidades del proyecto. Debe permitir visualizar y editar un corpus previamente extraído, obtener estadísticas de aparición de palabras, para lo cual se debe modificar la anterior *release* pues es en la creación cuando se puede determinar esto, permitir eliminar las palabras aparecidas, modificar su contenido (aunque no se recomienda, para ello mejor modificar el *stemmer*), y comparar con otro corpus mayor para obtener ratios comparativos.
- Extracción de características y generación de los ficheros externos. Debe permitir seleccionar el tipo de representación a obtener y a partir de ella obtener un fichero utilizable por Weka para el entrenamiento y la validación

- Navegación y clasificación. No es objetivo de la investigación pero servirá de uso práctico de la misma. Mediante la lectura de los modelos previamente generados con Weka, el navegador integrado permitirá la navegación simultánea con la categorización de las páginas sobre las que navegar.
- Herramientas estadísticas, de visualización y de manipulación, que permitan ver los repositorios, modificarlos y combinarlos, obtener estadísticas de distribución de Webs por clases y demás herramientas que faciliten las tareas a realizar.

I.2 Análisis del sistema mediante los casos de uso o *use cases*

Los casos de uso permitirán completar los requisitos y especificaciones del sistema así como modelar de manera formal el problema planteado, guiando de manera lógica el proceso de construcción de la solución. A su vez, los casos de uso nos delimitarán el sistema y mejorarán la comprensión de su funcionamiento.

En primer lugar habrá que determinar los actores principales del sistema, posteriormente se identificarían los casos de uso más comunes y por último se representaría de manera gráfica estos casos.

I.2.1 Determinación de los actores

El principal actor del sistema será la persona que lo utiliza en ese momento. El sistema no es multiusuario por lo que las tareas serán realizadas independientemente, de manera secuencial y sin interferencia de agentes externos.

Ahora bien, se pueden identificar dos roles principales en el uso del sistema, el rol TÉCNICO, que sería aquél mediante el cuál la persona que utiliza el sistema realiza las tareas técnicas de extracción de características para el posterior entrenamiento, y el rol USUARIO, que sería aquél usuario que utiliza el sistema, previamente entrenado, para categorizar nuevas páginas web por las que navegue.

Estos dos roles nacen de la naturaleza dual de la propia aplicación, por un lado el conjunto de herramientas que responden al objetivo perseguido en la investigación, que no es más que obtener una representación de las páginas de modo que mediante aprendizaje inductivo se pueda aprender un modelo que sirva para una futura predicción, y por otro lado la aplicación de dicho proceso en una etapa posterior de clasificación o predicción de nuevas páginas dentro de sus categorías.

En cualquier caso, por su naturaleza secuencial y monousuario, el actor USUARIO será el identificado como el que utiliza el sistema bien sea para generar los ficheros de entrenamiento, bien sea para utilizar los modelos aprendidos.

Un segundo actor sería la herramienta de aprendizaje inductivo, en nuestro caso WEKA, que se alimentaría de resultados obtenidos por nuestro sistema, los ficheros arff, y generaría resultados que alimentarían a nuestro sistema, los modelos inductivos aprendidos.

Es por tanto que este segundo actor, WEKA, junto con el primer actor, USUARIO, conforman el conjunto de actores que interactuarán con el sistema.

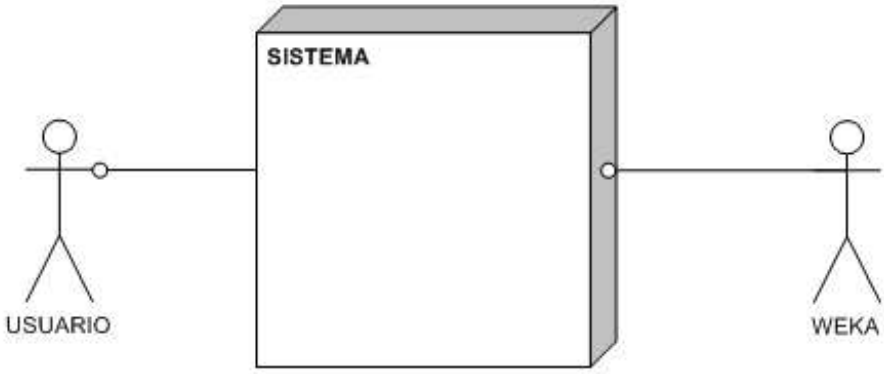


FIGURA I.1: Grafico de actores del sistema

I.2.2 Identificación de los casos de uso (*use cases).* Diagrama de casos de uso

Los casos de uso vienen determinados por las tareas realizadas por los diferentes actores en el sistema, por lo que los mismos vendrán identificados por las diferentes tareas que el sistema tenga encomendado hacer.

Por las especificaciones del sistema, el mismo se puede considerar como un sistema específico que realiza unas determinadas funciones, de manera secuencial, por lo que cada caso de uso se corresponderá directamente con cada una de las funciones que se permitan en el mismo.

El diagrama de los casos de uso resume de manera gráfica los casos de uso que han aparecido en el sistema, lo que clarificará gráficamente la interrelación de los elementos:

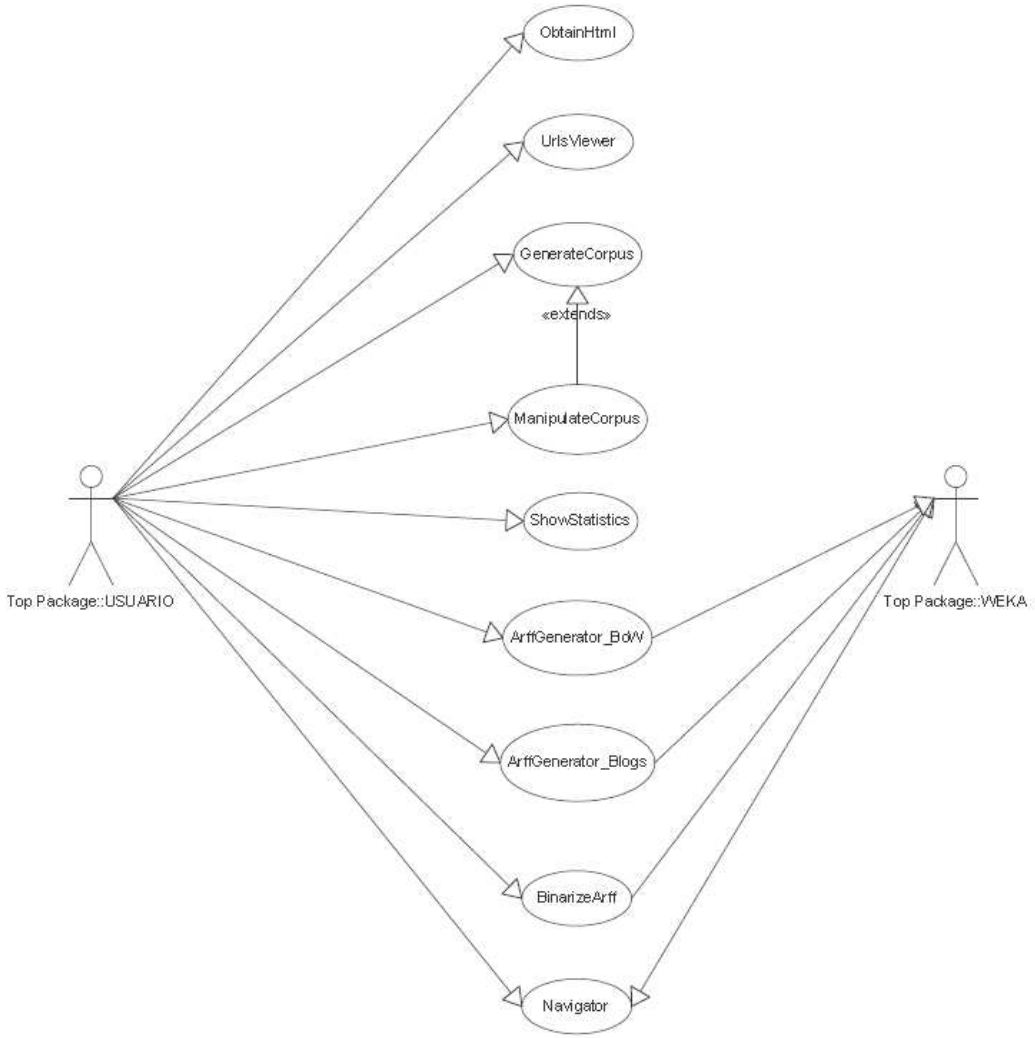


FIGURA I.2: Diagrama de casos de uso

Con ello, los diferentes casos de uso que se darán serán los siguientes:

Obtención de documentos html

PROPÓSITO	Obtener los documentos html requeridos desde Internet y almacenarlos en un repositorio local al sistema para su posterior tratamiento de manera local.
PRECONDICIONES	Conexión a Internet
ACTIVACIÓN	A discreción del actor USUARIO en rol TÉCNICO

FLUJO PRINCIPAL	VARIACIONES	EXCEPCIONES
	USUARIO decide cargar un repositorio inicial para completarlo con una nueva extracción. SISTEMA incluye las nuevas Urls en el repositorio actual en lugar de crear uno nuevo	
SISTEMA requiere un listado de Urls a extraer en un formato concreto		
USUARIO introduce un nivel de crawl máximo		
USUARIO indica comienzo de extracción		
SISTEMA recorre la lista de Urls, extrae los documentos, y genera un repositorio en memoria. Para cada URL actualiza el estado de progreso	USUARIO decide parar el proceso, con lo que SISTEMA detiene la extracción y se mantienen los elementos extraídos actuales en el repositorio de memoria	No se puede extraer una URL por cualquier motivo (no existe, no hay conexión a Internet, el servidor está caído...) SISTEMA continúa con la siguiente URL
USUARIO, al finalizar SISTEMA, indica que desea grabar el repositorio en disco		

Y su diagrama del flujo de trabajo:

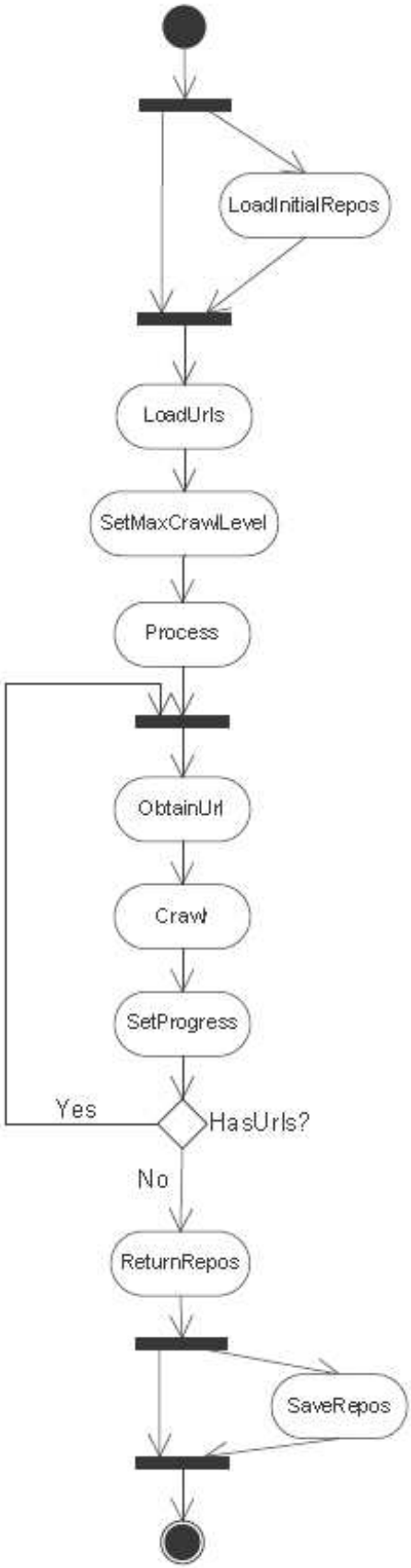


FIGURA I.3: Diagrama del flujo de trabajo ObtainHtml

Visualización de datasets de documentos

PROPÓSITO	Permitir la visualización y manipulación del dataset de páginas web extraídas para un correcto preprocesado de las mismas
PRECONDICIONES	
ACTIVACIÓN	A discreción del actor USUARIO en rol TÉCNICO

FLUJO PRINCIPAL	VARIACIONES	EXCEPCIONES
SISTEMA requiere el repositorio de Urls extraídas		
USUARIO selecciona repositorio		
SISTEMA muestra en una rejilla las Urls, junto con la clase y el contenido html		
USUARIO realiza las manipulaciones que desee sobre las mismas	USUARIO añade Urls desde otro repositorio. SISTEMA requiere dicho repositorio. USUARIO lo selecciona. SISTEMA combina las Urls del repositorio actual con el nuevo y las muestra en la rejilla	
USUARIO decide guardar el repositorio manipulado		
SISTEMA requiere un fichero destino		
USUARIO selecciona el fichero destino		
SISTEMA, previa eliminación de duplicados, almacena el repositorio en disco		

Y su diagrama del flujo de trabajo:

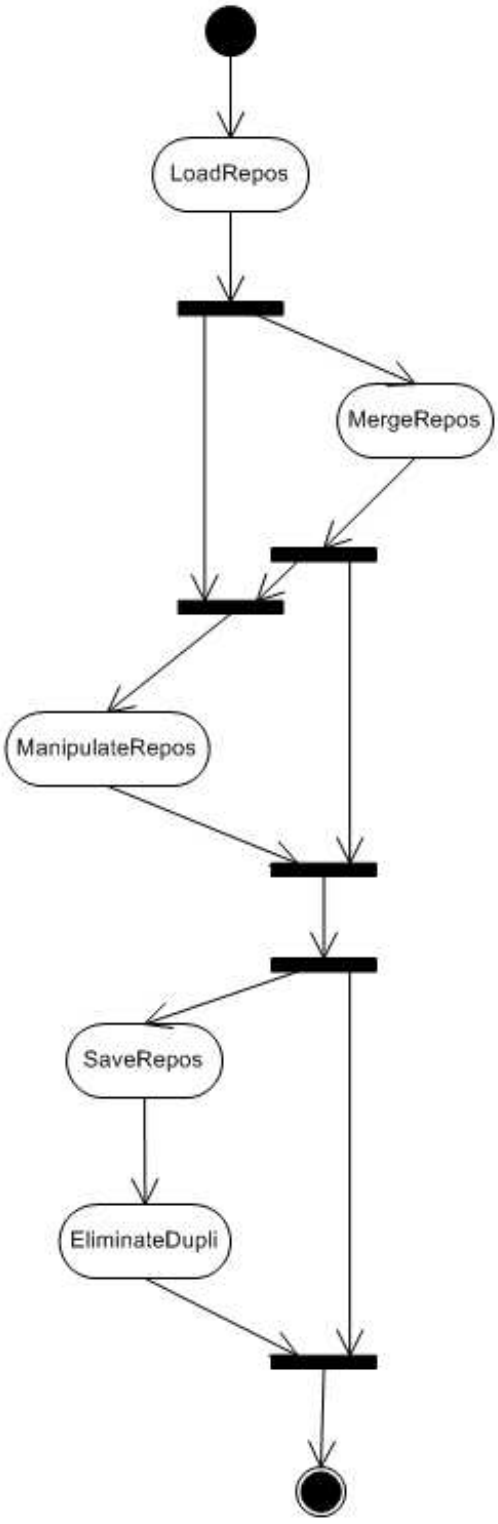


FIGURA I.4: Diagrama del flujo de trabajo UrlsViewer

Generación de corpus

PROPÓSITO	Permitir la generación de un corpus lingüístico a partir de un conjunto de documentos preseleccionados
PRECONDICIONES	Listado de SWL Listado de parejas lingüísticas stem
ACTIVACIÓN	A discreción del actor USUARIO en rol TÉCNICO

FLUJO PRINCIPAL	VARIACIONES	EXCEPCIONES
SISTEMA requiere de un fichero con el repositorio de páginas previamente extraídas		
USUARIO selecciona el repositorio		
USUARIO selecciona generar corpus		
SISTEMA requiere de un fichero de salida para el corpus generado		
USUARIO selecciona el fichero destino		
SISTEMA extrae las palabras de los documentos, rellena el corpus y lo almacena en el fichero destino		

Y su diagrama del flujo de trabajo:

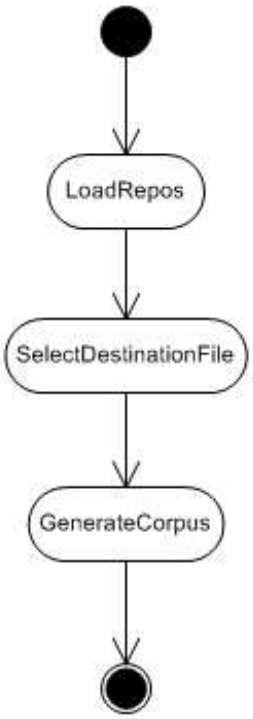


FIGURA I.5: Diagrama del flujo de trabajo GenerateCorpus

Manipulación de corpus

PROPÓSITO	Permitir la manipulación de un corpus lingüístico previamente generado para su correcto ajuste a las necesidades
PRECONDICIONES	
ACTIVACIÓN	A discreción del actor USUARIO en rol TÉCNICO

FLUJO PRINCIPAL	VARIACIONES	EXCEPCIONES
SISTEMA requiere de un fichero con el corpus almacenado		
USUARIO selecciona el corpus		
USUARIO realiza las operaciones que desee sobre el mismo, como eliminar o modificar	USUARIO puede seleccionar otro corpus para obtener la ratio de aparición en el corpus actual frente al nuevo corpus	
USUARIO selecciona almacenar las modificaciones		
SISTEMA requiere de un fichero destino		
USUARIO selecciona fichero destino		

SISTEMA almacena el corpus en local		

Y su diagrama del flujo de trabajo:

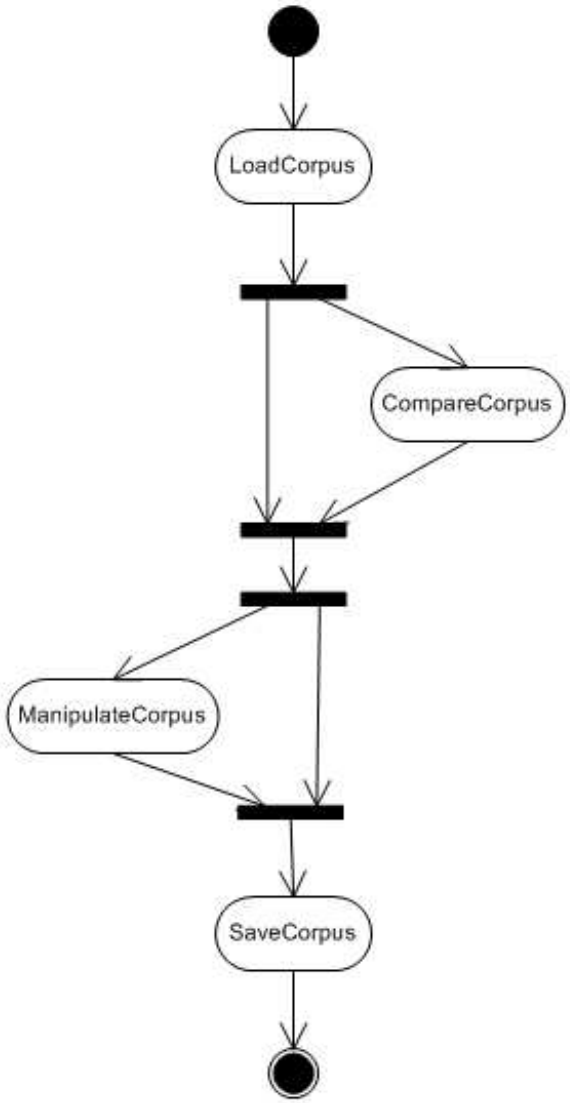


FIGURA I.6: Diagrama del flujo de trabajo ManipulateCorpus

Visualización de estadísticas de clases

PROPÓSITO	Mostrar la distribución de páginas por clases
PRECONDICIONES	
ACTIVACIÓN	A discreción del actor USUARIO en rol TÉCNICO

FLUJO PRINCIPAL	VARIACIONES	EXCEPCIONES
SISTEMA requiere repositorio de páginas para evaluar		

USUARIO selecciona repositorio de páginas		
SISTEMA muestra estadísticas		

Y su diagrama del flujo de trabajo:

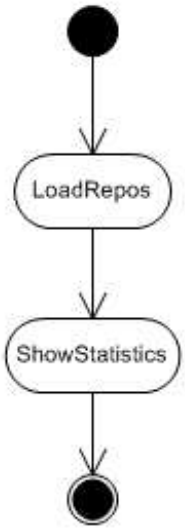


FIGURA I.7: Diagrama del flujo de trabajo EstadísticasClase

Generación de ficheros arff BoW

PROPÓSITO	Obtener un fichero ARFF con los atributos específicos para la clasificación BoW y los valores correspondientes para las Urls utilizadas
PRECONDICIONES	
ACTIVACIÓN	A discreción del actor USUARIO en rol TÉCNICO

FLUJO PRINCIPAL	VARIACIONES	EXCEPCIONES
SISTEMA requiere corpus		
SISTEMA requiere repositorio de Urls		
USUARIO selecciona corpus		
USUARIO selecciona repositorio de Urls		
USUARIO selecciona crear	USUARIO selecciona BoW estándar USUARIO selecciona BoW mejorado: SISTEMA requiere las ponderaciones para los diferentes	

	elementos que el USUARIO introduce USUARIO selecciona BoW sobre URL USUARIO selecciona BoW combinado	
SISTEMA requiere fichero de salida ARFF		
USUARIO selecciona fichero de salida ARFF		
SISTEMA extrae las características de las Urls y genera las líneas del fichero ARFF		

Y su diagrama del flujo de trabajo:

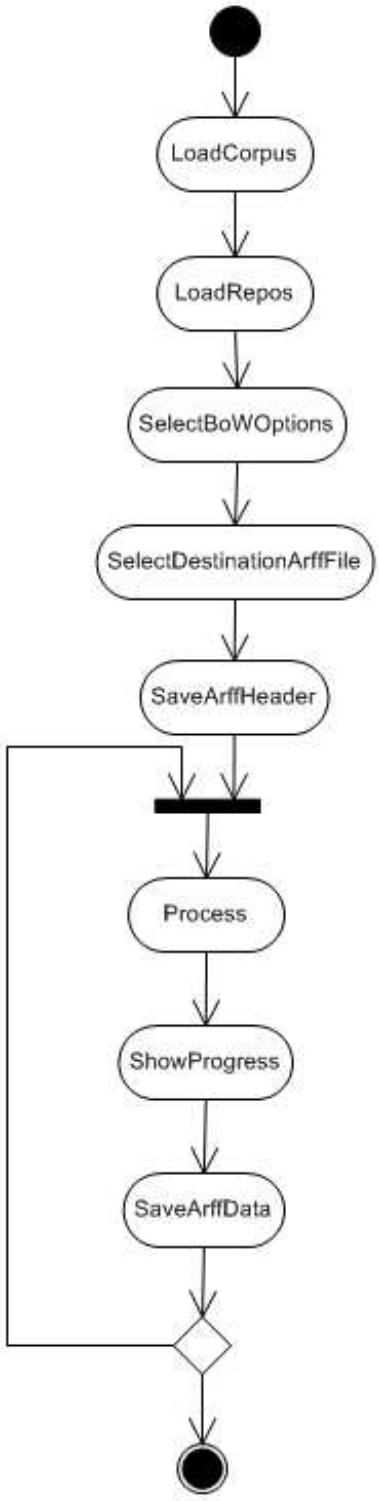


FIGURA I.8: Diagrama del flujo de trabajo ArffCreator_BoW

Generación de ficheros arff Blogs

PROPÓSITO	Obtener un fichero ARFF con los atributos específicos para la clasificación como Blog y los valores correspondientes para las Urls utilizadas
PRECONDICIONES	
ACTIVACIÓN	A discreción del actor USUARIO en rol TÉCNICO

FLUJO PRINCIPAL	VARIACIONES	EXCEPCIONES
SISTEMA requiere repositorio con Urls		
USUARIO selecciona dicho repositorio		
USUARIO selecciona comenzar proceso		
SISTEMA requiere fichero salida ARFF		
USUARIO selecciona fichero salida ARFF		
SISTEMA extrae características de las Urls y añade las líneas al fichero ARFF		

Y su diagrama del flujo de trabajo:

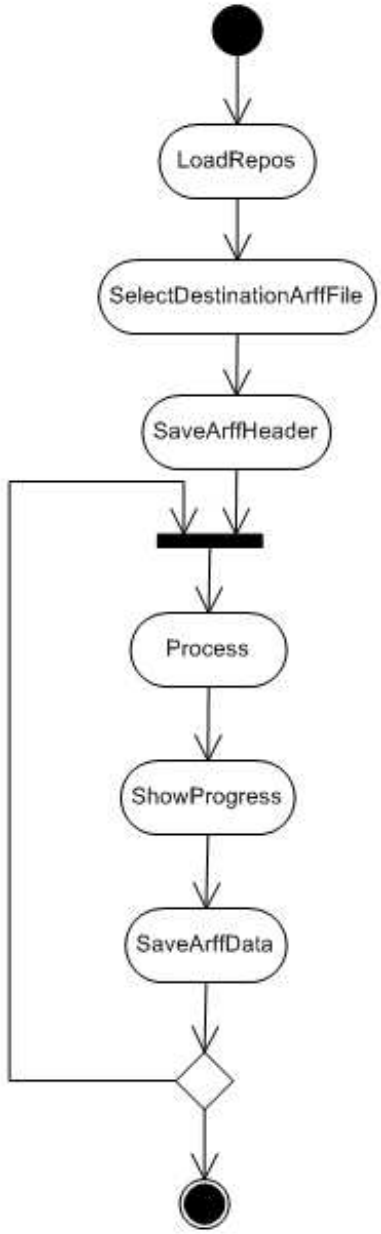


FIGURA I.9: Diagrama del flujo de trabajo ArffCreator_Blogs

Binarizador

PROPÓSITO	Obtener a partir de un fichero ARFF con todas las N categorías, N ficheros con la pertenencia o no a la categoría concreta
PRECONDICIONES	
ACTIVACIÓN	A discreción del actor USUARIO en rol TÉCNICO

FLUJO PRINCIPAL	VARIACIONES	EXCEPCIONES
SISTEMA requiere del usuario la selección de un fichero arff inicial para binarizar		

USUARIO selecciona fichero arff original para binarizar		
SISTEMA genera tantos ficheros binarios como categorías existen		

Y su diagrama del flujo de trabajo:

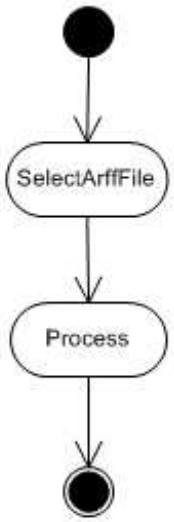


FIGURA I.10: Diagrama del flujo de trabajo BinarizeArff

Navegación/categorización

PROPÓSITO	Permitir la navegación por Internet y la categorización de las páginas accedidas en tiempo real.
PRECONDICIONES	Precargados modelos de clasificación Conexión a Internet
ACTIVACIÓN	A discreción del USUARIO en rol USUARIO

FLUJO PRINCIPAL	VARIACIONES	EXCEPCIONES
USUARIO solicita la navegación a una url	USUARIO navega dentro de la propia página y SISTEMA obtiene cada pulsación en enlace y la procesa como una petición de navegación	
SISTEMA navega a la url solicitada		No hay conexión a Internet: ERROR No existe la URL: ERROR
SISTEMA clasifica y muestra la clasificación		No existen modelos: SISTEMA no clasifica pero permite la navegación

Y su diagrama del flujo de trabajo:

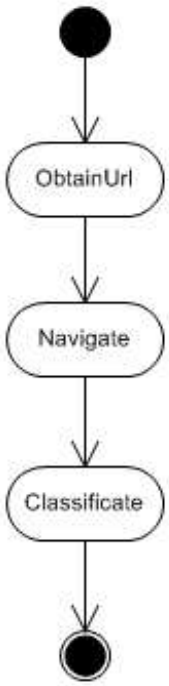


FIGURA I.11: Diagrama del flujo de trabajo Navigate&Classificate

I.3 Análisis de la interfaz de usuario

Vista la delimitación del sistema y su relación o interacción con los actores que intervienen, se debe especificar la interfaz mediante la cual se producirá dicha interacción.

Debida a la especificación de requisitos del sistema, la interacción del usuario con el sistema se limita al intercambio de información entre ambos para conducir el proceso desde los datos que se poseen a la entrada hasta los datos que se espera obtener en cada proceso a la salida.

La interfaz elegida para el diseño del sistema corresponde a la interfaz estándar de aplicaciones Windows MDI (Multi-Document Interface) o interfaz de documentos múltiples, de manera que sobre el mismo interfaz se podrá tener diferentes ventanas con diferentes procesos abiertos y trabajando de manera concurrente, a los cuales se accederá a partir de un menú estándar.

Esto es así no por requerimiento propio del sistema, pues cada interacción con el mismo, como ya se ha indicado, conduce a un proceso independiente de los demás y guiado de principio a fin, sino para facilitar la posibilidad, favorecida por dicha independencia y debido al tiempo de ocupación que requieren alguno de los procesos por parte del sistema, de realizar diferentes tareas de manera concurrente. Así pues, se

pueden extraer páginas desde Internet a la par que se visualiza estadísticas de un repositorio actual, se estudia un corpus o incluso se generan o binarizan ficheros arff.

El análisis de un interfaz requiere de la definición de la interacción en el sentido de la entrada y la salida. Como ya se ha indicado, la interfaz será estándar de Windows con lo cual toda interacción vendrá determinada por su modelo de componentes gráficos, que facilitan la usabilidad y la visualización de los datos.

Así pues, la definición concreta de las entradas y las salidas se define a continuación, para cada uno de los posibles casos de uso:

Caso de uso: Obtener Html		
ENTRADAS	**SALIDAS**	**REALIMENTACIÓN VISUAL**
DataSet inicial: Se seleccionará mediante un control de selección de ficheros a cargar al que se accederá desde un botón y se cargará un fichero con un formato prefijado	DataSet final: Se seleccionará mediante un control de selección de fichero a grabar accesible desde un botón y se almacenará en un fichero con un formato prefijado	Rutas a los ficheros cargados y número de elementos que contienen en etiquetas
Repositorio Xml: Se seleccionará mediante un control de selección de ficheros a cargar al que se accederá desde un botón y se almacenará en un fichero con un formato prefijado		Total de Urls a procesar y url procesadas hasta el momento en una etiqueta
Niveles de crawl: Se indicará mediante un control caja de texto editable		Barra de progreso que resume visualmente lo anterior
		Nivel de crawl actual y máximo a procesar en una etiqueta
		Url y clase procesándose Conjunto de etiquetas con la distribución por clases de las Urls procesadas hasta el momento

Caso de uso: Visualización del repositorio de documentos		
ENTRADAS	**SALIDAS**	**REALIMENTACIÓN VISUAL**
Repositorio inicial: Se seleccionará mediante un selector de ficheros a cargar al que se accede desde un botón y que lee un fichero	Repositorio final: Se seleccionará mediante un selector de ficheros a grabar al que se accede desde un botón y que almacenará el	Rutas a los ficheros a cargar y grabar que se mostrarán en etiquetas

con un formato prefijado	repositorio en un fichero con un formato prefijado	
Añadir repositorio: Se seleccionará mediante un selector de ficheros a cargar al que se accede desde un botón y que leerá un fichero con un formato prefijado		Rejilla que mostrará el listado de Urls cargadas y que permitirá la siguiente interacción: • Ordenación seleccionando el campo en la cabecera • Eliminación, seleccionando el registro en el selector de registros y eliminando con la tecla "supr" • Modificación, seleccionando la celda a modificar y realizando las modificaciones deseadas • Navegación, haciendo doble click sobre la fila a navegar, abrirá el navegador integrado con la url seleccionada

Caso de uso: Generación de corpus		
ENTRADAS	**SALIDAS**	**REALIMENTACIÓN VISUAL**
Repositorio de Urls: se cargará a partir de un formulario de selección de fichero a cargar al que se accederá desde un botón y que leerá el repositorio desde un fichero con el formato prefijado	Corpus generado, que se almacenará en un fichero con un formato prefijado seleccionado mediante un control de selección de fichero a grabar, al cual se accede desde el botón de comenzar la creación del corpus	Ruta del repositorio mostrada en una etiqueta
¿Cuerpo o encabezados y enlaces?, opción disjunta a la que se accederá desde controles radio		Número de Urls procesadas y de atributos (palabras) extraídas hasta el momento Barra de progreso con las Urls extraídas hasta el momento frente al total
		Rejilla donde se mostrará el corpus obtenido, con las

		palabras y su número de apariciones. Se permite la eliminación y modificación de líneas
		Número de palabras del corpus que se mostrará encima del encabezado de la rejilla

Caso de uso: Manipular corpus		
ENTRADAS	**SALIDAS**	**REALIMENTACIÓN VISUAL**
Corpus, que bien vendrá dado por el proceso de generación de corpus descrito anteriormente, bien por la carga de un corpus externo desde un fichero con un formato prefijado seleccionado desde un control de selección de fichero a cargar al que se accede desde un botón	Corpus manipulado, que se almacenará en un fichero con un formato prefijado y que se seleccionará desde un control de selección de fichero a grabar accedido desde un botón.	Rejilla donde se mostrará el corpus obtenido, con las palabras y su número de apariciones. Se permite la eliminación y modificación de líneas. Si se compara el corpus con otro corpus se mostrará la ratio de apariciones de cada palabra en ambos corpus
Corpus de comparación, que se cargará desde un fichero con un formato prefijado seleccionándolo en un control de selección de fichero a cargar al que se accederá desde un botón		Número de palabras del corpus que se mostrará encima del encabezado de la rejilla

Caso de uso: Visualización de estadísticas de clase		
ENTRADAS	**SALIDAS**	**REALIMENTACIÓN VISUAL**
Repositorio de Urls, que se cargará de un fichero con un formato prefijado desde un selector de ficheros a cargar al que se accederá desde un botón		Ruta del repositorio de Urls mostrado en una etiqueta Número total de Urls mostrado en una etiqueta Número de Urls por clase mostrado en una etiqueta por clase

Caso de uso: Generación de ficheros ARFF BoW		
ENTRADAS	**SALIDAS**	**REALIMENTACIÓN VISUAL**
Fichero con corpus, cargado desde un fichero con un formato prefijado que se selecciona con un control de selección de fichero a		Etiqueta con la ruta al corpus

cargar accesible desde un botón		
Fichero con repositorio de Urls, cargado desde un fichero con formato prefijado que se selecciona con un control de selección de fichero a cargar accesible desde un botón		Etiqueta con la ruta al repositorio Etiqueta con el número de Urls del repositorio
Tipo de BoW en un control disjunto de botones radio: • Estándar • Mejorado: Accede a un formulario dónde introducir, en cajas de texto, los valores de la ponderación para cada elemento del html (título, enlaces, encabezados...) • Urls • Combinado		
	Crear fichero ARFF salida, seleccionado desde un control de selección de fichero a grabar accesible desde un botón de comenzar proceso	Urls procesadas hasta el momento en una etiqueta Urls sin contenido hasta el momento en una etiqueta Progreso url actual/total

Caso de uso: Generación de ficheros ARFF Blogs		
ENTRADAS	**SALIDAS**	**REALIMENTACIÓN VISUAL**
Fichero con repositorio de Urls, cargado desde un fichero con formato prefijado que se selecciona con un control de selección de fichero a cargar accesible desde un botón		Etiqueta con la ruta al repositorio Etiqueta con el número de Urls del repositorio
	Fichero ARFF salida, seleccionado desde un control de selección de fichero a grabar accesible desde el botón de comenzar proceso	Urls procesadas hasta el momento en una etiqueta Progreso url actual/total

Caso de uso: Binarizador		
ENTRADAS	**SALIDAS**	**REALIMENTACIÓN VISUAL**

Fichero ARFF a binarizar, que se seleccionará con un control de selección de fichero a cargar accesible desde un botón	N ficheros ARFF, uno por cada categoría, donde se almacenará el valor de clase a 1 si pertenece a esa categoría y 0 en caso contrario	Ruta al fichero ARFF en una etiqueta Clase actual/clases totales en una etiqueta Progreso visual clase actual/clases totales

Caso de uso: Navegador/Categorizador		
ENTRADAS	**SALIDAS**	**REALIMENTACIÓN VISUAL**
URL en una caja de texto editable	Página Web a la que se navega en un control explorador	
Navegar a URL, pulsando enter en la caja de texto de la url o pulsando el botón navegar	Categorías asignadas automáticamente a la página en controles de chequeo	
Navegar por la web, pulsando los enlaces de las páginas		

I.4 Modelo de objetos

El modelo de objetos es la siguiente fase en el análisis del sistema. Es un modelo estático del sistema, pues describe los objetos que en él aparecen, no como interactúan o se comportan. El modelo de objetos nos aporta información de los objetos que aparecen en el sistema, su significado o semántica, así como la relación con otros objetos. Por tanto los pasos a seguir en este proceso de modelado estático son:

- Identificar los objetos que aparecen en el sistema agrupándolos mediante clases de objetos

- Identificar las relaciones entre objetos, mediante asociaciones y agregados

- Identificar los atributos de las clases que aparecen, es decir, las características propias de las clases y que nos van a determinar su semántica

- Identificar las operaciones de los objetos de la clase, lo que determinará el comportamiento de dichos objetos.

- Realizar el diagrama de clases donde se muestra de manera gráfica todos los elementos anteriores.

I.4.1 Identificación de clases relevantes al problema

Para que el diseño sea modular y escalable es interesante realizar una división entre la funcionalidad y la interfaz final, como se verá en el apartado de diseño del sistema, pero ya desde el análisis se intenta plasmar esta diferenciación.

Así pues, en primer lugar se identificarán las clases de soporte encargadas de realizar toda la lógica de negocio, en este caso obtener el html y crear el dataset, obtener su representación, generar los ficheros de entrenamiento... y por otro lado las encargadas de interactuar con el usuario para guiarle en la tarea de preparación, generación y uso.

Clases de soporte

Crawler
La clase Crawler es la encargada de facilitar el acceso a la web, extraer las páginas indicadas y generar un repositorio o dataset con las mismas. La clase Crawler será estática permitiendo un acceso directo a sus funcionalidades públicas ya que no mantiene un estado interno sino que se limita a devolver unos resultados a partir de unos parámetros de entrada.

BoWMngr
La clase BoWMngr es la encargada de generar el vector de palabras de un documento dado. Para ello se encarga de resolver los problemas lingüísticos más básicos a partir de las listas de palabras vacías y el proceso de stemming. La clase provee toda la funcionalidad necesaria para crear un vector de palabras ponderados sus valores por frecuencia de aparición y un peso asignable.

HtmlMngr
La clase HtmMngr provee de una completa colección de métodos para acceder y tratar tanto el contenido como la estructura de los documentos html, permitiendo obtener gran variedad de características de los mismos y permitiendo generar diferentes modelos de representación a partir del html facilitado

ArffMngr
La clase ArffMngr permite la generación de ficheros Arff para Weka a partir de los atributos y valores de la representación de los documentos. Provee funcionalidad para crear la cabecera, con los atributos necesarios, y los datos a partir de la representación concreta (BoW, BoW combinado, BlogStruct)

Clases de interfaz

Las clases de interfaz han sido organizadas en tres grupos para mayor claridad y separación de su cometido.

Por un lado aparecen las herramientas previas a la creación de la representación, siendo aquellas las que permiten obtener las páginas desde Internet, los corpus necesarios y su comparación y extracción de estadísticas, la visualización de las páginas obtenidas y el cálculo de estadísticas con las mismas referente a las clases.

ObtainHtml
La clase ObtainHtml proporciona el interfaz necesaria para que el usuario pueda extraer aquellas páginas de Internet que requiera para su posterior tratamiento. Facilita información del progreso, así como estadísticas de las clases extraídas hasta el momento. Permite un proceso automático, dónde se indica el conjunto de Urls a extraer así como el nivel de crawl, y un proceso manual dónde se le indica una única url, así como el nivel de crawl, para añadir a la colección actual.

UrlsViewer
El visor de Urls permite visionar en una rejilla las Urls extraídas así como su clase y el html que las compone. Permite la eliminación automática de páginas duplicadas, la eliminación manual de cualquier página y el mezclado con datasets previamente obtenidos.

GenerateCorpus
La interfaz para generación de corpus permite la extracción, a partir del conjunto de páginas que conforman el dataset de entrada, de aquellos términos que deban formar parte del corpus. Se permite la extracción de corpus a partir del cuerpo del html, o de los enlaces y url. Se permite la manipulación del corpus, la comparación con otros corpus y la extracción de frecuencias de aparición, de manera que se pueda construir el corpus final

EstadisticaClases
Permite observar la distribución de las clases en el dataset previamente obtenido de la web.

Por otro lado se agrupan las herramientas necesarias para obtener la representación formal del documento en sus diferentes modalidades y almacenarla en disco para su posterior utilización por una herramienta de aprendizaje como Weka.

ArffCreator_BoW
Interfaz para la creación, a partir de un corpus y un dataset de Urls clasificadas, del fichero de entrenamiento para la aplicación Weka. Permite crear tres ficheros diferentes para tres representaciones diferentes, la BoW estándar, la BoW mejorada con elementos contextuales que se podrán configurar, BoW únicamente de las Urls y la BoW que ocupa el planteamiento del trabajo, dónde se tienen en cuenta los elementos de los metatags de la cabecera, la url y los enlaces.

BOWOptions
Permite, para el caso que se desee mejorar la BoW estándar, definir los valores de ponderación de cada posible elemento como el título, la url, los encabezados...

ArffCreator_Blogs
Interfaz para la creación del fichero de entrenamiento para el caso concreto de los Blogs

BinarizeArff
Interfaz para binarizar el fichero de entrenamiento creando tantos ficheros de

entrenamiento como categorías existen en el problema.

Por último el interfaz de navegación con categorización integrada que permite, a la par que se navega por las páginas web, obtener una categorización de las mismas en tiempo real, previa carga de los modelos de clasificación binaria obtenidos en las etapas anteriores.

Navigator
El navegador, previa carga de los modelos entrenados y listos para predecir, permitirá navegar al usuario a la página deseada y a través de sus enlaces, mostrando en todo momento, y en tiempo real, las categorías que se asignan a la página en cuestión

Y las clases para el manejo de los modelos adecuados a los Blogs y al resto de categorías:

ClsBlogs
La clase auxiliar para el manejo de Blogs permite cargar el modelo preentrenado para la clasificación de los mismos, así como la utilización de dicho modelo para clasificar nuevas instancias, que se crearán a partir del documento web.

ClsClass
La clase auxiliar para el manejo de las demás categorías permite cargar el modelo pre-entrenado para la clasificación de la categoría en cuestión, así como la utilización de dicho modelo para clasificar nuevas instancias, creadas también a partir del documento web.

Vistas las clases de objetos que pueden aparecer en el sistema, se procederá a la identificación de las relaciones entre éstas.

I.4.2 Asociaciones y agregados entre clases

Las asociaciones muestran que clases son interdependientes y por lo tanto muestran la relación entre las diferentes clases. Generalmente éstas asociaciones aparecen en la especificación del diseño

- La obtención del html requerirá del crawler para obtener el mismo desde la Web
- El navegador requerirá del crawler para obtener el html de la página actual para poder clasificarla
- La clase Blog hará un uso puntual del Crawler en caso de no tener el html de la página dada poder obtenerlo
- El crawler utilizará el manejador de html para obtener la lista de enlaces de la página actual, para poder continuar en la siguiente iteración
- El generador del corpus utilizará al manejador de html para obtener las palabras que lo compondrán
- Tanto el creador del arff de BoW como el de Blogs requerirán la clase de manejo de html para obtener las características necesarias para su cometido.

- El manejador de html requerirá de la clase de manejo de BoW para obtener la representación BoW de los elementos que lo necesiten
- La creación de las representaciones BoW como Blogs requerirán del manejador de Arff para crear los ficheros de entrenamiento
- La interfaz de creación de BoW requerirá de la interfaz de parametrización de pesos en el caso mejorado
- El navegador requerirá de la clase de manejo de los Blogs y demás categorías para realizar la clasificación

Como se puede observar, por el tipo de aplicación final dónde las tareas están muy definidas y son muy independientes entre sí, la relación entre clases se reduce al apoyo de las clases de interfaz en las clases de soporte para realizar sus tareas, así como entre las clases de soporte para realizar pasos sucesivos en la abstracción del problema.

I.4.3 Atributos de las clases

Los atributos de las clases, como se dijo con anterioridad, nos van a determinar la semántica de dicha clase, especificando de este modo el significado hasta el punto que requiera el sistema.

Atributos de la clase Crawler
No tiene atributos de clase. Es una clase estática que proporciona soporte para el acceso a la Web y la obtención de los documentos

Atributos de la clase HtmlMngr
• BoWMngr: Instancia del manejador de BoW para obtener la representación como vector de elementos varios

Atributos de la clase BoWMngr
• SWL: Tabla hash con las palabras vacías utilizadas • Stem: Tabla hash con las correspondencias entre las palabras completas del diccionario y su correspondiente reducción a los lemas comunes

Atributos de la clase ArffMngr
• ArffFileName: Nombre del fichero arff a generar

Atributos de la clase ObtainHtml
• UrlsRepos: Dataset con las Urls clasificadas o no que se desea extraer de manera automática desde la Web • HtmlRepos: Dataset con las páginas extraídas o que se extraerán desde la Web

Atributos de la clase UrlsViewer
• HtmlRepos: DataSet con las páginas previamente extraídas de la Web sobre las que se quieren realizar tareas de visualización

Atributos de la clase GenerateCorpus
• HtmlRepos: DataSet con las páginas que servirán para la generación del corpus

- Corpus: Tabla Hash con el corpus generado a modo de parejas palabra/frecuencia de aparición
- ProcessEnded: Booleano indicando el fin de la generación, para controlar el aborto de la misma por parte del usuario

Atributos de la clase EstadisticaClases
No tiene atributos

Atributos de la clase ArffCreator_BoW

- HtmlRepos: DataSet con las páginas clasificadas que servirán para la generación del fichero de entrenamiento o prueba
- Corpus: Tabla hash con el corpus sobre el que trabajar
- StrBoWOptions: Estructura de ponderaciones que definen las mejoras introducidas sobre el BoW estándar
- ArffMngr: Manejador de fichero arff utilizado para la generación del mismo
- ReposReaded: Booleano indicando si el dataset de páginas ha sido leído, y por tanto puede comenzar el proceso
- ProcessEnded: Booleano para manejar la cancelación por parte del usuario del proceso de generación

Atributos de la clase BOWOptions

- StrBoWOptions: Estructura de ponderaciones que definen las mejoras introducidas sobre el BoW estándar

Atributos de la clase ArffCreator_Blogs

- HtmlRepos: Conjunto de páginas que servirán para la creación del fichero de entrenamiento/test para los Blogs
- ReposReaded: Booleano indicando si el dataset de páginas ha sido leído, y por tanto puede comenzar el proceso
- ProcessEnded: Booleano para manejar la cancelación por parte del usuario del proceso de generación

Atributos de la clase BinarizeArff
No tiene atributos

Atributos de la clase Navigator

- Url: Dirección Web a la que navegar y clasificar
- ClsBlog: Clase de manejo y clasificación de los Blogs
- ClsClass: Clase de manejo y clasificación del resto de categorías

Atributos de la clase ClsBlogs

- StrBlogs: Estructura con las características o atributos para definir un Blog
- Classifier: Clasificador weka para la clase Blogs

Atributos de la clase ClsClass

- Class: Nombre de la clase o categoría de la instancia concreta de ClsClass
- Classifier: Clasificador weka para la clase dada

I.4.4 Operaciones

Las operaciones de las clases determinarán su comportamiento. En el sistema se encuentran principalmente las siguientes.

Operaciones de la clase Crawler
• GetHtml: Obtiene el html de una página a partir de su Url. Si se le adjunta un dataset con páginas no busca en Internet si no que busca en dicho dataset • Crawl: Realiza un crawl en profundidad a partir de la url dada, sin salirse del dominio de la url base indicada, y hasta la profundidad máxima definida, devolviendo un dataset con los htmls obtenidos. Si se le adjunta un dataset con documentos Web, se mezcla con los obtenidos en la etapa actual del crawl.

Operaciones de la clase HtmlMngr
• GetHtmlDoc: Obtiene un documento html a partir de un texto en html, para su mejor tratamiento mediante métodos y atributos propios de la clases mshtml de .Net • TokenizeUrl: Elimina los tokens que no aportan información de las Urls como ? & _ - / : acentos, números y elementos sustitutos de acentos y demás... • GetBISHList: Obtiene un listado de las palabras que aparecen bajo el tag de encabezado indicado • GetLinks: Obtiene el conjunto de Urls que aparecen en los enlaces de la página dada • GetHead: Obtiene el fragmento de html definido entre <head></head> • GetNWordsBody: Obtiene el número de palabras que conforman el cuerpo del html • GetNWordsUrl: Obtiene el número de palabras que conforman la url, previa tokenización • GetNWordsLinks: Obtiene el número de palabras de los enlaces de la página • GetNWords: Dada una lista de palabras o un texto obtiene el número de palabras individuales que lo componen • GetNumOccurs: Devuelve el número de ocurrencias de un texto en otro texto • GetHtmlTag: Obtiene el fragmento de html contenido dentro de los tags dados como parámetro • GetNHtmlTag: Obtiene el número de elementos definidos por un tag dado que existen en el documento html • GetNComments: Obtiene el número de elementos comentario que aparecen en el documento html • GetNCommentsInLinks: Obtiene el número de elementos comentario que aparecen en los enlaces del documento html • GetNLinks: Obtiene el número total de enlaces que hay en un documento html, así como los que son al propio dominio y los que son a dominios externos • GetNLinksIslands: Obtiene las islas de enlaces del documento html, devolviendo un booleano indicando la existencia de alguna, el número de enlaces totales que contienen las mismas, el número de enlaces a páginas internas del dominio y el número de enlaces externas al dominio • GetNDates: Obtiene el número de fechas contenidas en el documento, teniendo en cuenta diferentes formatos de las mismas • GetBodyBow: Obtiene la representación BoW en una tabla hash para el contenido del cuerpo del documento html con la ponderación dada

- GetTitleBoW: Obtiene la representación BoW en una tabla hash para el contenido del título del documento html con la ponderación dada
- GetHeadBoW: Obtiene la representación BoW en una tabla hash para el contenido del encabezado del documento html con la ponderación dada
- GetUrlBoW: Obtiene la representación BoW en una tabla hash para el texto contenido en la url del documento html con la ponderación dada
- GetLinksUrlsBoW: Obtiene la representación BoW en una tabla hash para el texto de la url de los enlaces que contiene el documento html con la ponderación dada
- GetLinksBoW: Obtiene la representación BoW en una tabla hash para el texto de los enlaces que contiene el documento html con la ponderación dada
- GetBISHBoW: Obtiene la representación BoW en una tabla hash para el texto contenido en la etiqueta de encabezado indicada con la ponderación dada
- GetBoW: Obtiene la representación BoW en una tabla hash para una combinación ponderada de todas las representaciones anteriores para el documento html dado.

Operaciones de la clase BoWMngr

- GetBoW: Obtiene una tabla hash con los valores de frecuencia de aparición de las palabras del documento html. Permite la ponderación del peso de las palabras, así como la discriminación o no de las palabras que no se encuentren en el stemmer.
- ReadStems: Lee de un fichero externo las parejas de palabras y su lema stem
- ReadSWL: Lee de un fichero externo las palabras vacías
- IsSW: Dada una palabra indica si es vacía
- Normalize: Dado un texto elimina los espacios por el principio y final y realiza un stemming del mismo
- Lemmatize: Realiza un stemmin del texto dado a la entrada. Permite indicar si devuelve o no aquellas palabras que no existan en las tablas del stem.

Operaciones de la clase ArffMngr

- CrearCabeceraArff: Genera la cabecera de un fichero Arff con los atributos definidos en el corpus. Se le puede indicar si creará atributos simples, sólo el corpus, o combinados, corpus para encabezado, enlaces y url.
- CrearCabeceraArffBlogs: Genera la cabecera de un fichero Arff con los atributos predefinidos para la representación de los Blogs
- AñadirDatosArff: Añade una línea de datos con los valores obtenidos para el corpus simple, o los tres corpus en el caso de la clasificación combinada
- AñadirDatosArffBlogs: Añade una línea de datos con los valores obtenidos para la representación de los Blogs

Operaciones de la clase ObtainHtml

- LoadRepos: Permite cargar un dataset con páginas previamente extraídas de la Web
- SaveRepos: Permite grabar el dataset de páginas extraídas en disco
- LoadUrls: Permite cargar el conjunto de Urls clasificadas para realizar la extracción del contenido html de las mismas desde la web
- ObtainHtml: A partir de las Urls clasificadas, realiza un crawl con los parámetros seleccionados por el usuario referentes a niveles de crawl, y rellena el dataset de páginas.
- ShowStatistics: Muestra el progreso de la extracción y las estadísticas de clases que extraídas hasta el momento

Operaciones de la clase UrlsViewer
• LoadRepos: Permite cargar un dataset con páginas previamente extraídas de la Web para su visualización • SaveRepos: Permite grabar el dataset visualizado en la rejilla a almacenamiento en disco • MergeRepos: Permite mezclar el dataset cargado con otro dataset previamente grabado en disco • DeleteDupli: Elimina aquellas páginas que compartan url y clase, eliminando por tanto posibles duplicados. • ShowUrls: Lanza el navegador integrado con la url sobre la cuál se haya efectuado doble click

Operaciones de la clase GenerateCorpus
• LoadRepos: Permite cargar el dataset de páginas previamente extraído de la web • GenerateCorpus: Genera el corpus a partir de las palabras contenidas en las Webs extraídas en el dataset, a partir del cuerpo de las mismas, o a partir del encabezado y enlaces, según la opción seleccionada • ReadCorpus: Permite cargar un corpus previamente almacenado • SaveCorpus: Permite salvar el corpus visualizado en almacenamiento en disco • CompareCorpus: Permite comparar un corpus con otro obteniendo ratio de aparición de cada palabra en el corpus principal frente a la aparición en el secundario. • ShowCorpus: Muestra el corpus en la rejilla de visualización y muestra el número de palabras que lo componen

Operaciones de la clase EstadisticaClases
• LoadRepos: Permite cargar el dataset de páginas • ShowStatistics: Muestra el número de páginas que componen cada clase

Operaciones de la clase ArffCreator_BoW
• ReadCorpus: Permite cargar el corpus que servirá para la creación del fichero de entrenamiento/test • LoadRepos: Permite cargar el fichero que contiene las páginas previamente descargadas de la web y clasificadas para la generación del fichero de entrenamiento/test • ObtainImproveOptions: Muestra el formulario de opciones BoW para mejorar la ponderación de determinados elementos • CreateArff: Comienza la creación del fichero Arff a partir de todos los datos obtenidos por las operaciones anteriores • ShowProgress: Muestra el progreso de la creación del fichero

Operaciones de la clase BOWOptions
• LoadOptions: Carga las opciones previamente introducidas por el usuario y las muestra en los controles apropiados • SaveOptions: Almacena en la estructura de opciones los valores introducidos por el usuario

Operaciones de la clase ArffCreator_Blogs
• LoadRepos: Permite cargar el fichero que contiene las páginas previamente descargadas de la web y clasificadas para la generación del fichero arff • CreateArff: Comienza la creación del fichero Arff • ShowProgreso: Muestra el progreso de la creación del fichero

Operadores de la clase BinarizeArff
• BinarizeFile: Permite cargar un fichero Arff generado para todas las categorías conjuntamente y generar un fichero de clasificación binaria para cada categoría independientemente

Operaciones de la clase Navigator
• Navegar: Navega a la página indicada por el usuario • Clasificar: Clasifica la página indicada por el usuario

Operaciones de la clase ClsBlogs
• LoadClassifier: Carga un modelo de clasificador previamente entrenado • IsBlog: Dada una url y opcionalmente su contenido html indica si se puede clasificar como blog • GetInstance: A partir de una url y su html asociado devuelve una instancia de Weka con las características y valores para un elemento de la clase blog • GetBlogAttributtes: A partir de una url, su html y su clase devuelve el conjunto de características propias definidas para un blog

Operaciones de la clase ClsClass
• LoadClassifier: Carga un modelo de clasificador del tipo concreto previamente entrenado • IsClass: Dada una url y opcionalmente su contenido html indica si se puede clasificar como la clase para la que fue instanciado el objeto • GetInstance: A partir de una url y su html asociado devuelve una instancia de Weka con las características y valores para un elemento del tipo de la clase • GetClassAttributtes: A partir de una url, su html y su clase devuelve el conjunto de características propias definidas para dicha clase

I.4.5 Diagramas de clases

El diagrama de clases muestra de manera gráfica todas las clases de objetos, con sus atributos y operaciones y la interacción entre ellas. Su realización concluiría con la fase de modelado de objetos.

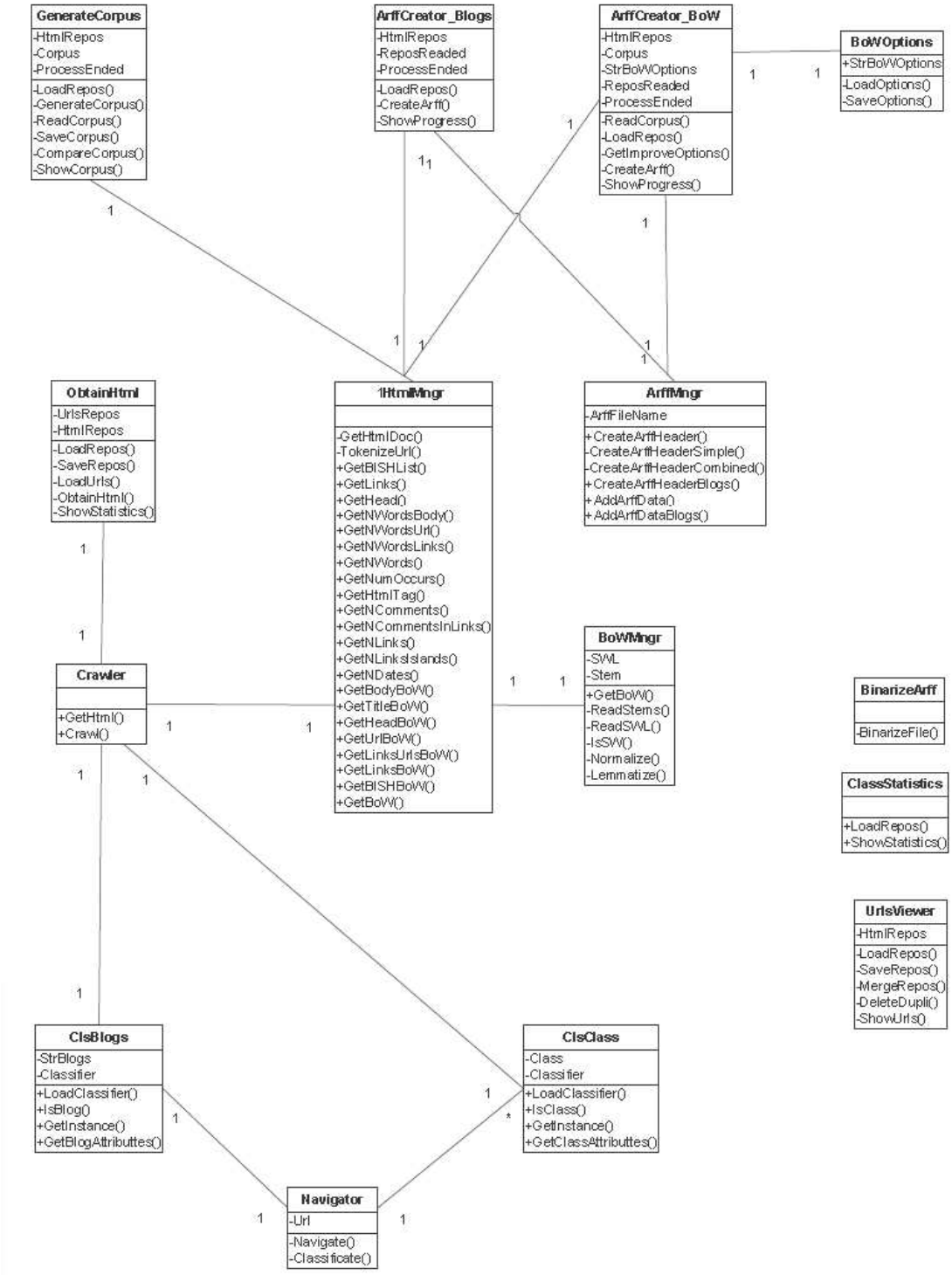


FIGURA I.12: Diagrama de clases

I.5 Modelo dinámico

Es la siguiente fase en el modelado UML para análisis de sistemas. En el modelo dinámico se estudia el comportamiento de los objetos durante su ciclo de vida, describiéndose por un lado la interacción entre los diversos objetos, y por otro lado la evolución interna de cada uno de ellos.

En éste punto se debe tener una visión global del funcionamiento del sistema, para plasmarlo mediante los tres apartados siguientes:

- Escenarios o diagramas de secuencia
- Diagramas de estados
- Diagramas de colaboración

I.5.1 Escenarios o diagramas de secuencia

Un escenario muestra de qué manera interactúan los distintos objetos dentro del flujo principal de eventos de un Caso de Uso, representando las interacciones de objetos ordenadas en una serie temporal que muestra su ciclo de vida.

Por tanto, habrá tantos escenarios como casos de uso se identificaron.

Obtención de documentos html

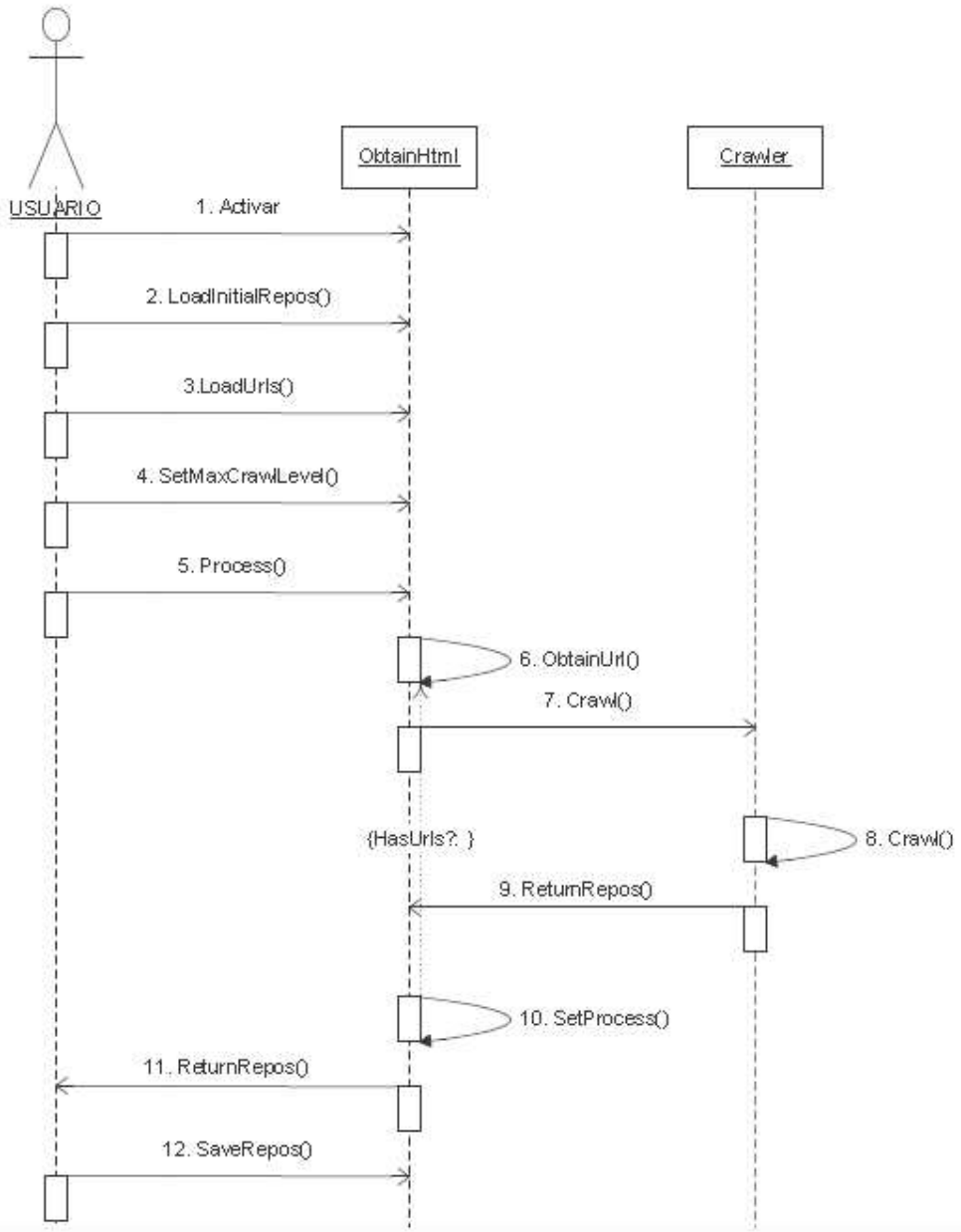


FIGURA I.13: Diagrama de secuencia ObtainHtml

Visualización del repositorio de documentos

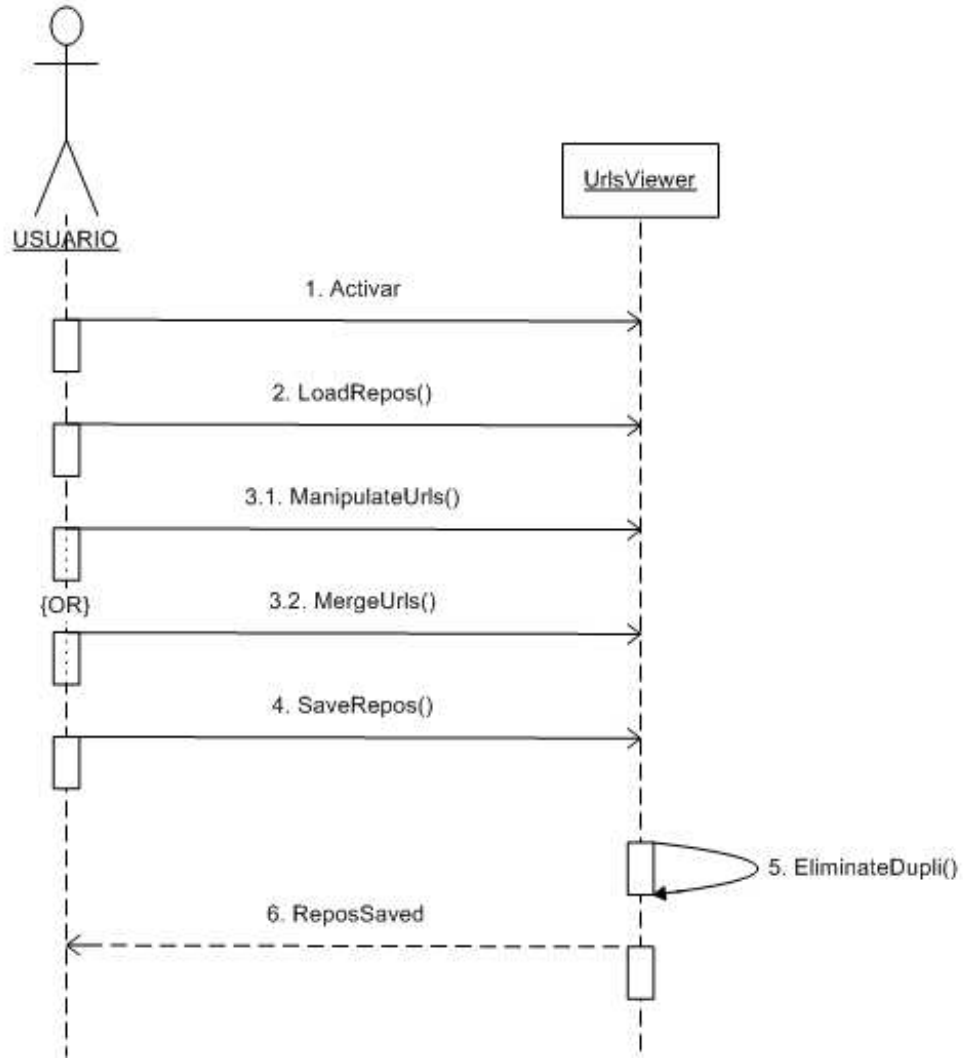


FIGURA I.14: Diagrama de secuencia UrlsViewer

Generación del corpus

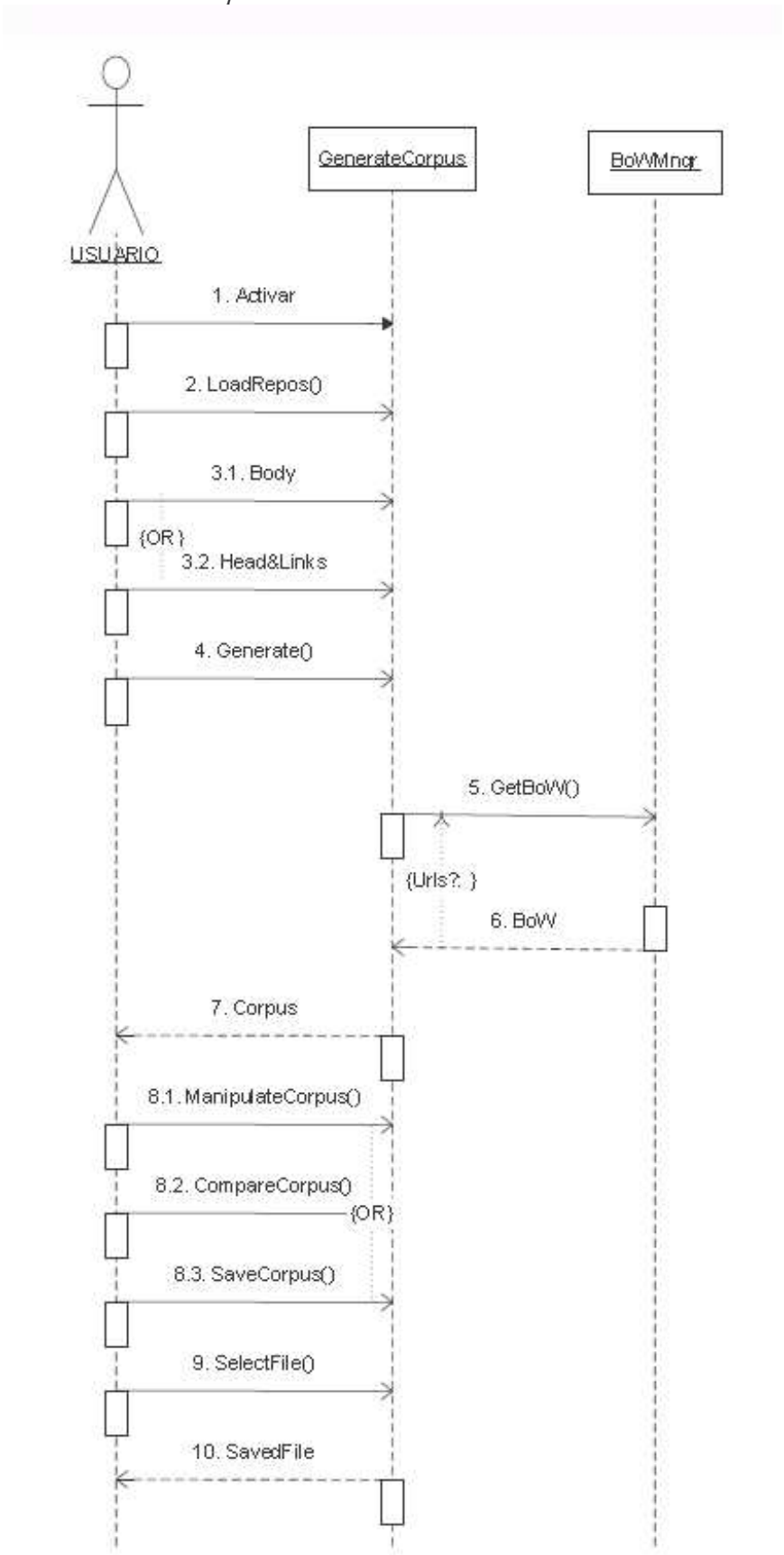


FIGURA I.15: Diagrama de secuencia GenerateCorpus

Manipulación del corpus

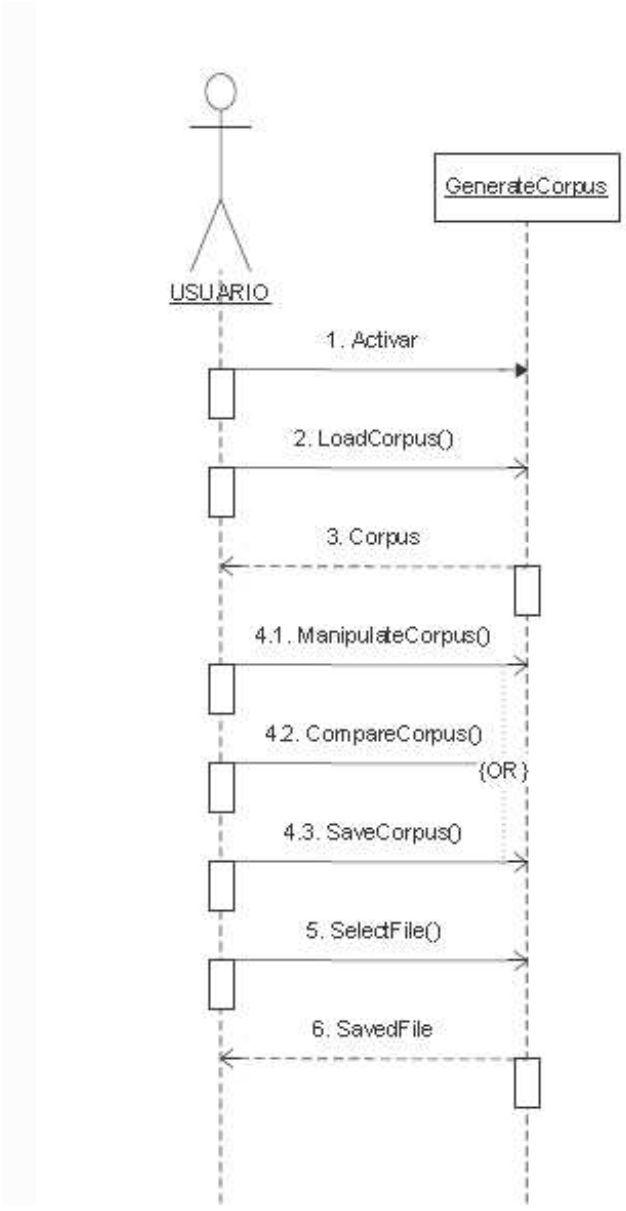


FIGURA I.16: Diagrama de secuencia ManipulateCorpus

Visualización de estadísticas de clases

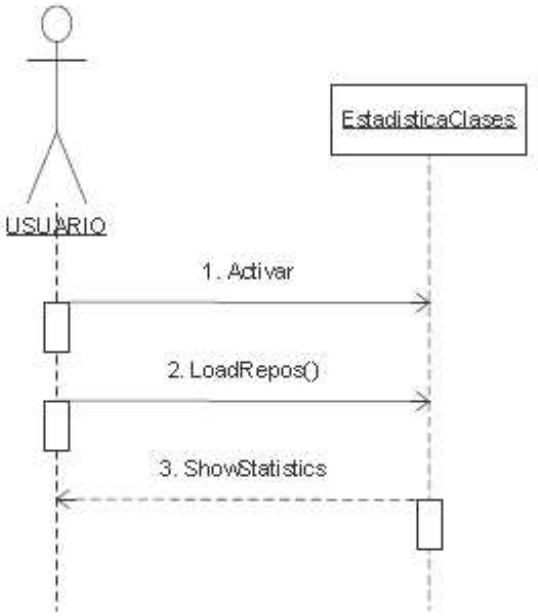


FIGURA I.17: Diagrama de secuencia EstadísticaClases

Generación de ficheros ARFF BoW

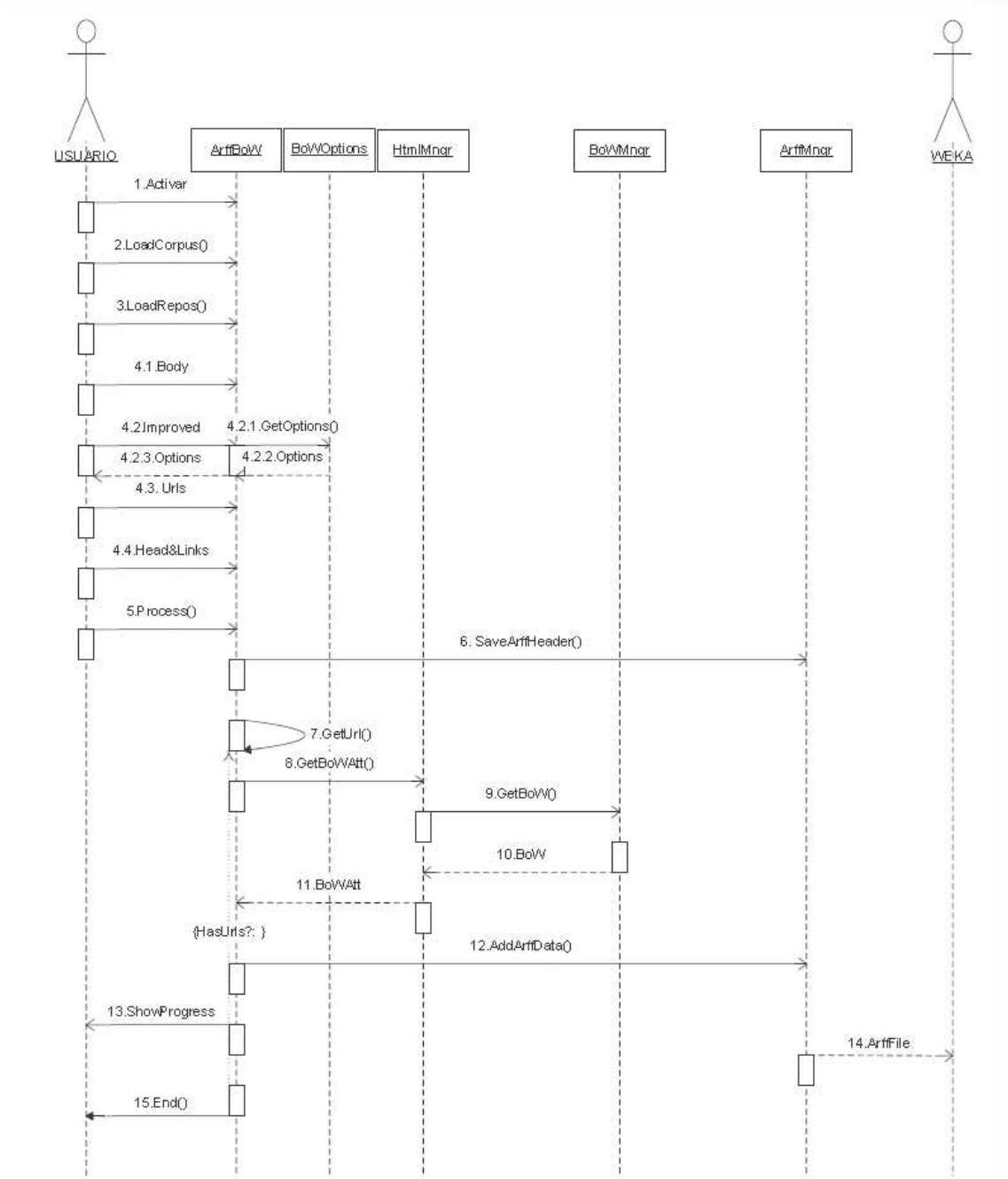


FIGURA I.18: Diagrama de secuencia ArffCreator_BoW

Generación de ficheros ARFF Blogs

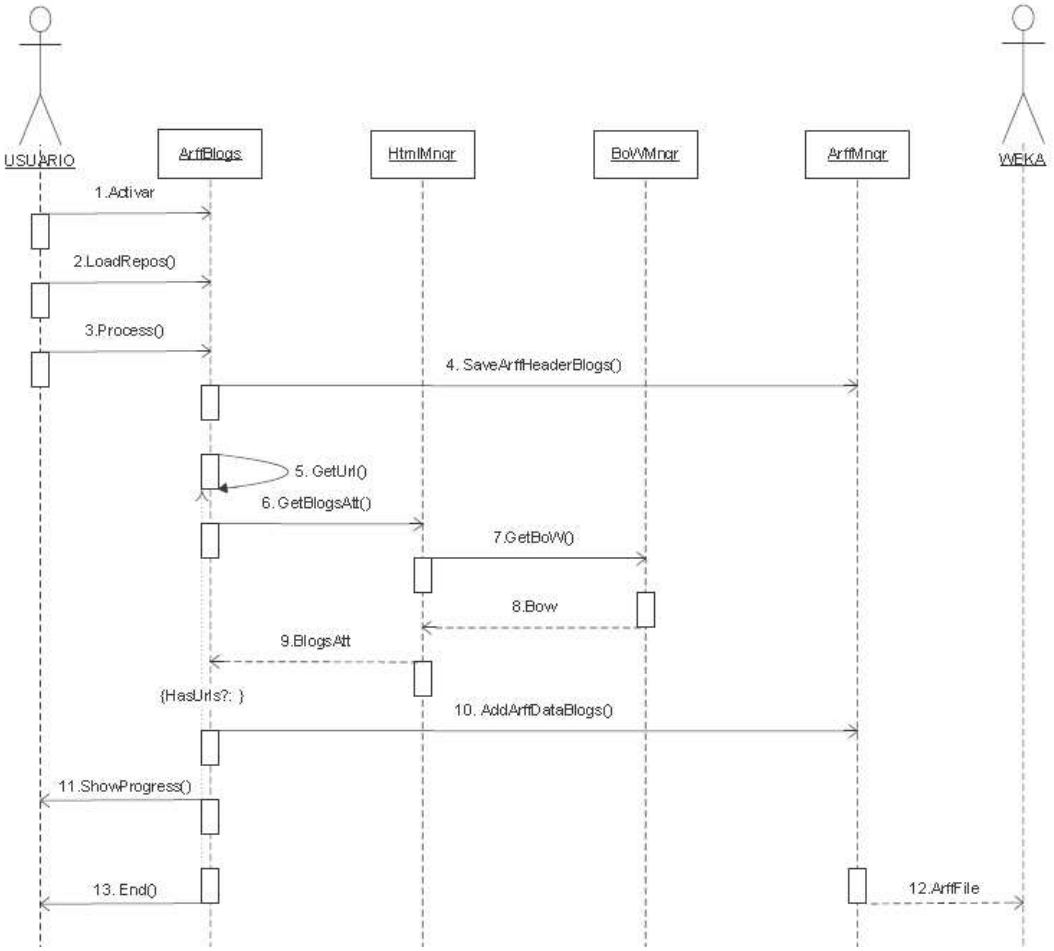


FIGURA I.19: Diagrama de secuencia ArffCreator_Blogs

Binarizador

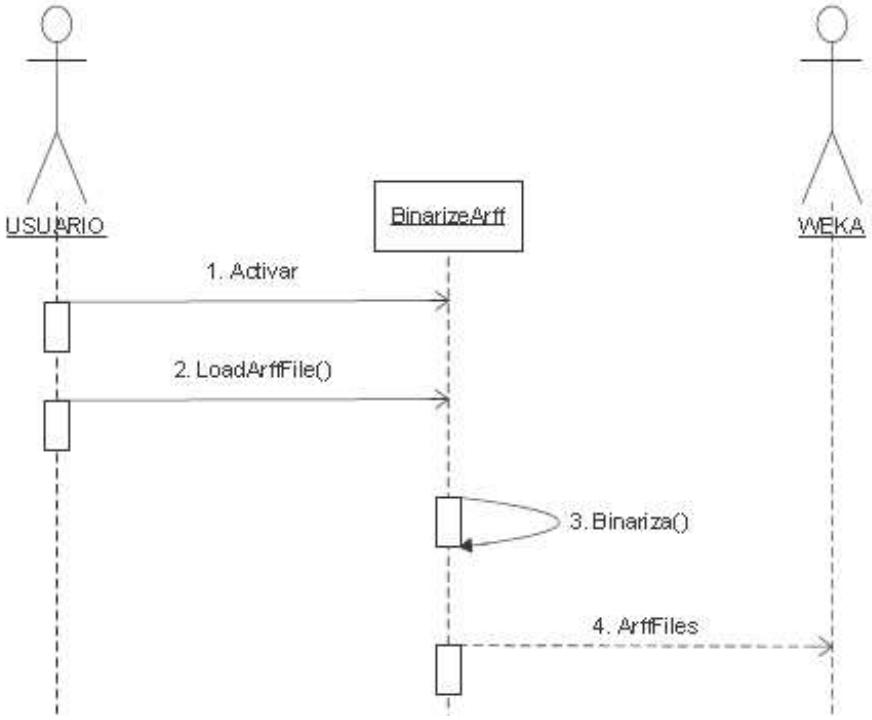


FIGURA I.20: Diagrama de secuencia BinarizeArff

Navegador/categorizador

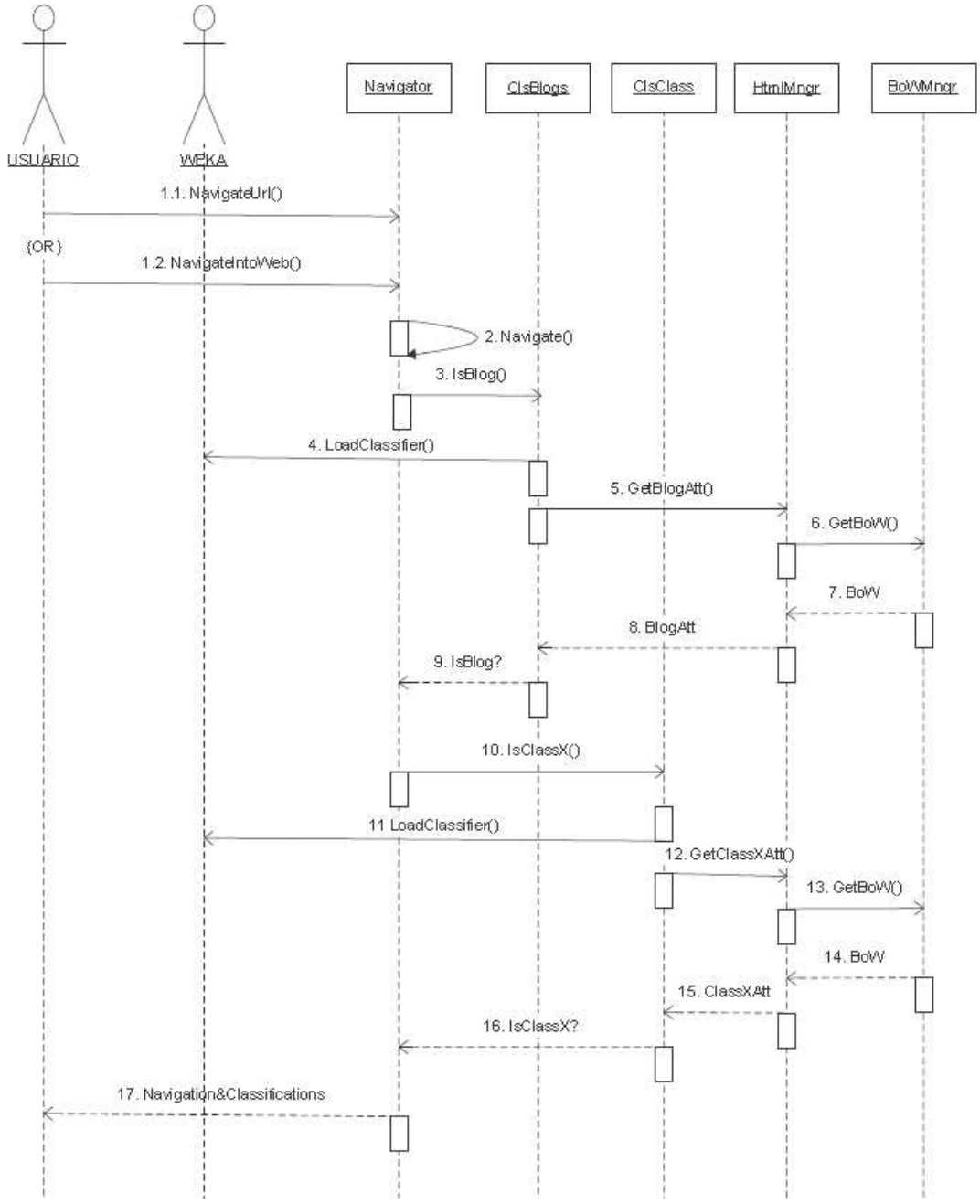


FIGURA I.21: Diagrama de secuencia Navigate&Classificate

I.5.2 Diagramas de estados

El diagrama de estados es un grafo que indica el comportamiento temporal de una determinada clase, donde los estados posibles se representan mediante nodos, y los cambios de estado, así como los sucesos que los provocan serían los arcos que unen dichos nodos.

El conjunto de estados en los cuales puede estar una determinada clase en los sistemas en estudio es tan reducido que no merece la pena mostrarlos, debido a que a lo sumo, una clase podrá tener dos estados, activada y desactivada, es decir, los objetos de las clases se activarán para realizar un proceso secuencial y a su finalización devolverán unos resultados y se desactivarán.

Su comportamiento queda mejor especificado mediante los diagramas de colaboración.

I.5.3 Diagramas de colaboración

Los diagramas de colaboración entre objetos representan la cooperación entre los objetos presentes en el sistema para realizar una funcionalidad. Dicha representación se realiza desde un punto de vista estático y dinámico, e incluye aquellos objetos que implementen una determinada función en el sistema.

Su finalidad es describir los mensajes que intercambian los distintos objetos para cumplir con las responsabilidades definidas en un escenario concreto de un Caso de Uso.

Por lo tanto, se muestran a continuación los diagramas de colaboración para cada uno de los Casos de Usos definidos anteriormente:

Obtención de documentos html

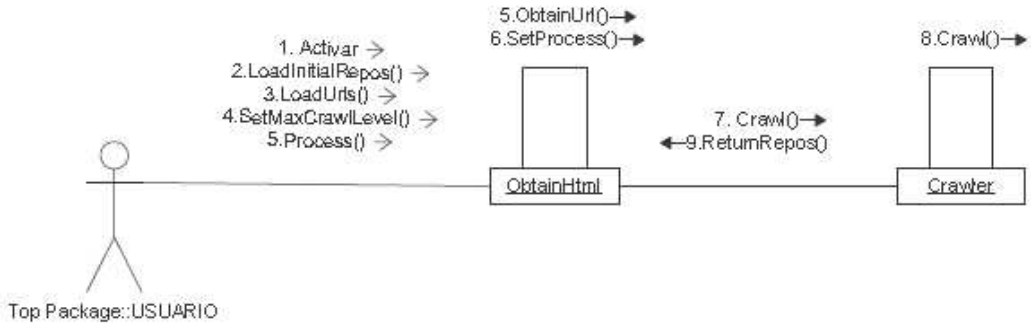


FIGURA I.22: Diagrama de colaboración ObtainHtml

Visualización del repositorio de documentos

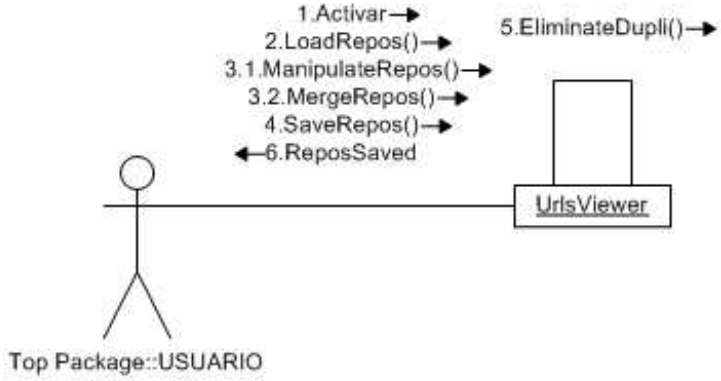


FIGURA I.23: Diagrama de colaboración UrlsViewer

Generación del corpus

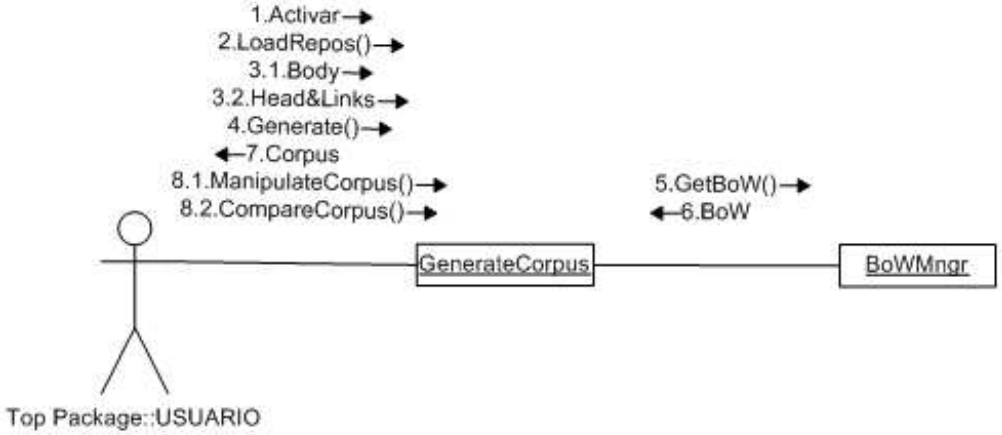


FIGURA I.24: Diagrama de colaboración GenerateCorpus

Manipulación del corpus

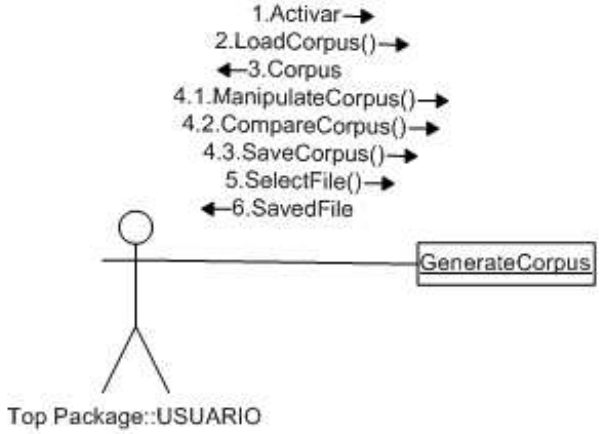


FIGURA I.25: Diagrama de colaboración ManipulateCorpus

Visualización de estadísticas de clases

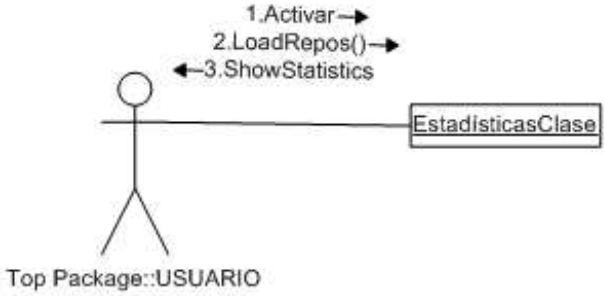


FIGURA I.26: Diagrama de colaboración EstadísticaClases

Generación de ficheros ARFF BoW

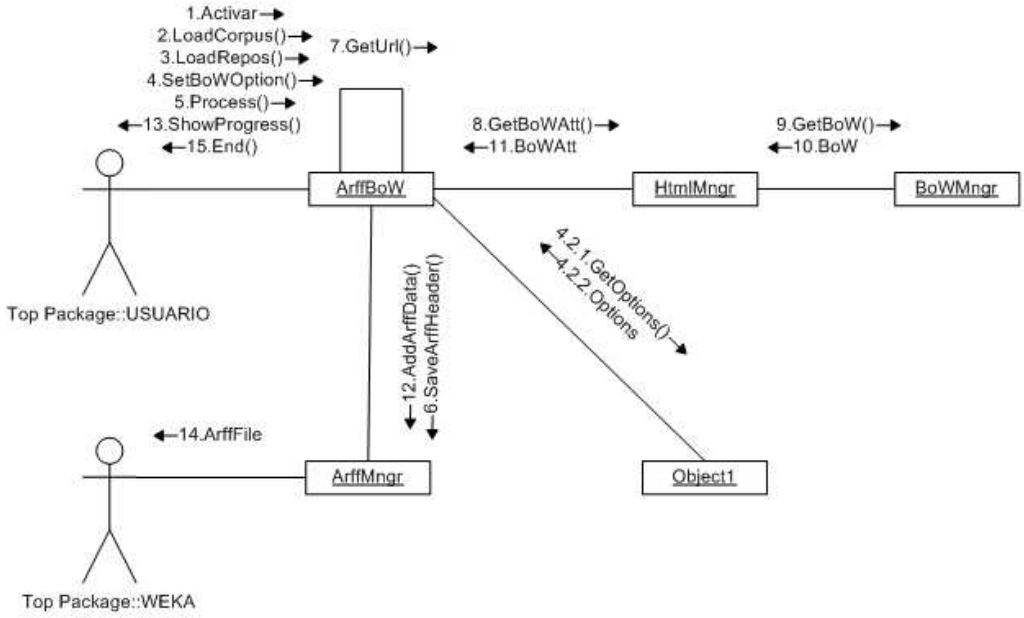


FIGURA I.27: Diagrama de colaboración ArffCreator_BoW

Generación de ficheros ARFF Blogs

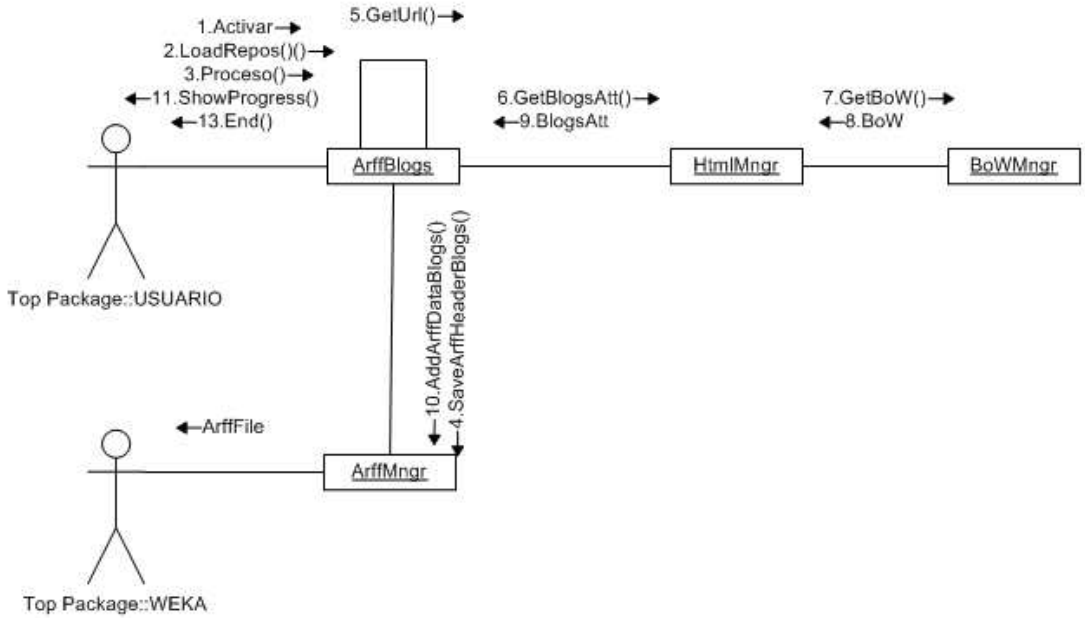


FIGURA I.28: Diagrama de colaboración ArffCreator_Blogs

Binarizador

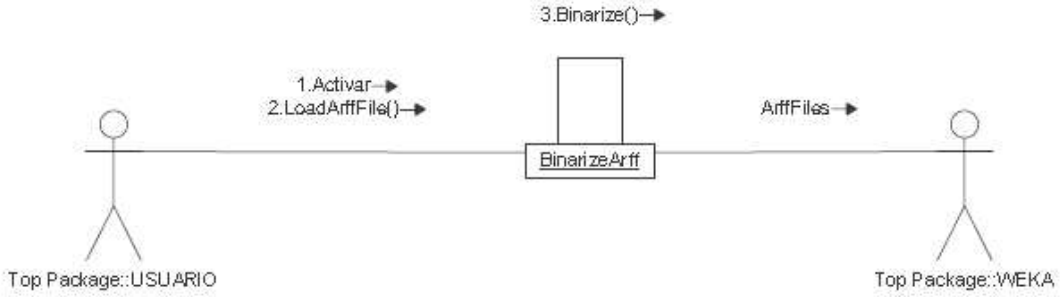


FIGURA I.29: Diagrama de colaboración BinarizeArff

Navegador/categorizador

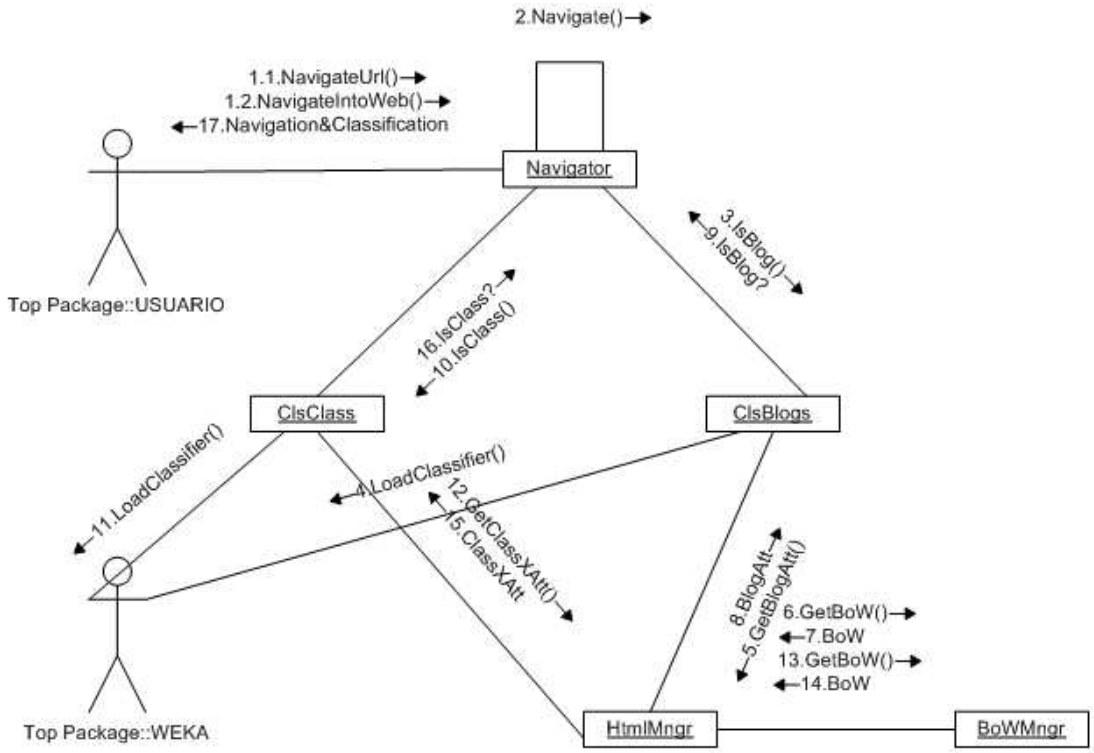


FIGURA I.30: Diagrama de colaboración Navigate&Classificate

ANEXO II. DISEÑO DEL SISTEMA CLASIFICADOR

Una vez realizado el análisis del sistema final, se procede a su diseño atendiendo a diferentes criterios que definirán la consecución de los objetivos. En primer lugar se tendrá en cuenta la descomposición del sistema en diferentes subsistemas o módulos que ayuden a la obtención del sistema global como un conjunto de subsistemas sencillos conceptualmente y cuya integración determinan el todo. En segundo lugar se tendrán en cuenta las prioridades a la hora de tomar una decisión de diseño de modo que se obtengan de la manera más ajustada posible los prerrequisitos impuestos al proyecto.

El diseño por tanto consistirá en dar respuesta de una manera técnica al cómo realizar el sistema modelado en la fase de análisis anterior. En primer lugar se dividirá el sistema en los distintos subsistemas que se han considerado adecuados, y en segundo lugar se hará hincapié en las prioridades a seguir a la hora de tomar decisiones. Por último se diseñarán los objetos que realizarán las acciones necesarias para la consecución de los objetivos.

II.1 Descomposición modular

Como se ha podido comprobar a lo largo de la descripción del proyecto el mismo responde a una descomposición modular bastante simple y poco conectada entre sus módulos. Así pues sí que es necesaria la secuenciación en las tareas para ir desde los datos facilitados a la entrada, el conjunto de Urls anotadas, hasta los resultados esperados, la construcción de un modelo y su evaluación por un sistema como Weka, así como su posterior uso en un entorno que podría ser real.

La descomposición en módulos es la siguiente:

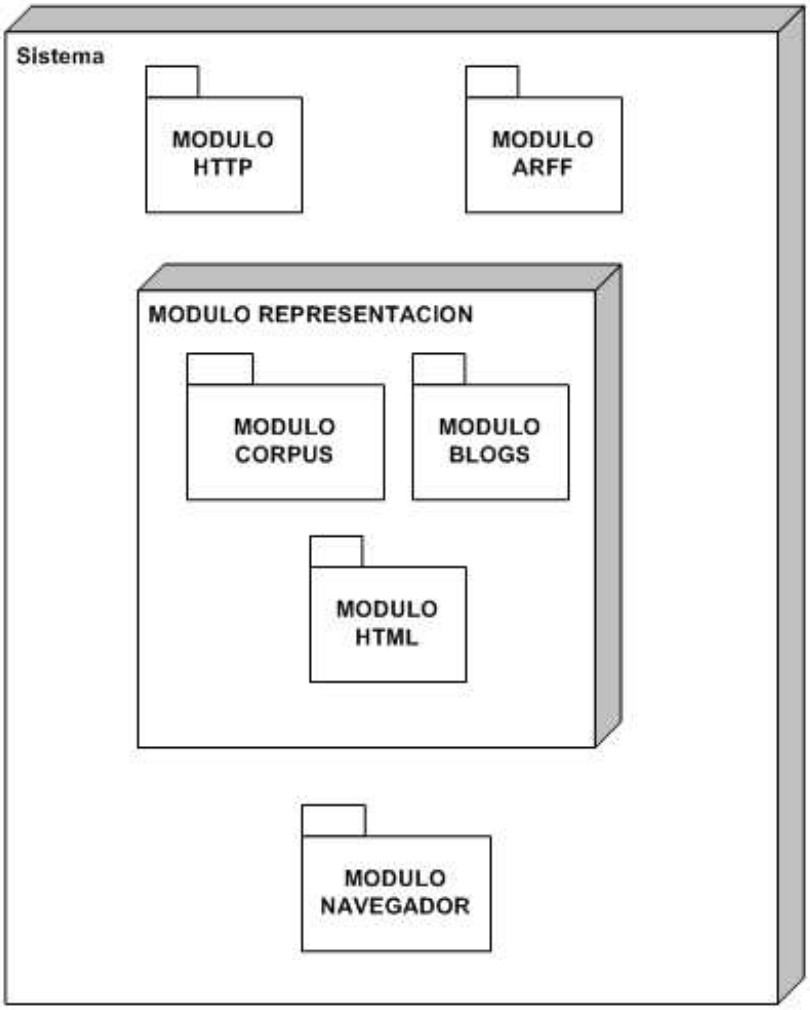


FIGURA II.1: Arquitectura modular del sistema

II.2 Prioridades del diseño

En cuanto a las prioridades del diseño, habrá que remontarse a la especificación de los requisitos y al objetivo inicial de la investigación.

El objetivo inicial de la investigación es obtener una representación lo suficientemente ajustada a las páginas a clasificar de modo que permita un aprendizaje inductivo y su posterior utilización mejorando el estado de arte actual y presentando alguna novedad en su diseño.

Dados las anteriores prioridades, todas las decisiones de diseño e implementación de ahora en adelante se deberán tomar siguiendo éstas, asegurando de este modo el cumplimiento de los requisitos y la necesidad de adaptar lo mejor posible la solución a las necesidades a satisfacer.

II.3 Diseño del interfaz hombre-máquina

La evolución de los interfaces, su homogeneización, la utilización de controles y widgets estándar, hacen de esta tarea de diseño una etapa más o menos sencilla en comparación con los tiempos en el que había que definir hasta el modo de interacción.

Pese a ello, hay que tomar ciertas decisiones sobre el modo de presentar la interfaz al usuario así como la manera en que ambos interactuarán.

Por otro lado, la presente investigación requiere del trabajo con dos interfaces diferentes, por un lado la propia del programa a desarrollar, y por otra con la interfaz del programa Weka. La comunicación entre ambos se realizará mediante el traspaso de ficheros que ambos sistema sean capaces de entender, los ficheros Arff.

Así mismo, por las características de los datos a tratar, presentes de manera distribuida en Internet, se hace necesaria la posibilidad de almacenamiento en dispositivo local, de manera que habrá que definir la forma y método para ello.

Con ello, el siguiente punto 4.3.1 describe la interfaz de la aplicación y el 4.3.2 el formato de los diferentes ficheros de intercambio y tratamiento.

II.3.1. Diseño de la UI

La UI se define siguiendo el estándar MDI Forms de Windows. Los MDI o Multi Document Interface permiten la apertura de múltiples documentos a modo de ventanas bajo el entorno general.

Esto permite la rápida selección de una ventana sobre la que trabajar, así como la realización de tareas en paralelo.

Por las características de alguno de los módulos, la realización de tareas en paralelo puede significar un aumento en el rendimiento del usuario. Así por ejemplo, mientras el sistema está realizando un crawl de las páginas anotadas el usuario puede ir realizando un estudio de un corpus previamente creado, visualizar un repositorio de Urls o generar un fichero arff para su entrenamiento y validación.

Respecto a los controles se utiliza el conjunto de controles estándar de Microsoft de los cuales cualquier usuario está familiarizado con su uso.

Una serie de ejemplos de ejecución se facilitan en el ANEXO III

II.3.2. Diseño de los ficheros de intercambio

La investigación se inicia con un fichero de Urls anotadas dónde se indica a qué categoría de teatro pertenecen.

En el análisis se tomó la decisión de obtener, a partir de este fichero de anotaciones, el conjunto de páginas web y almacenarlas en un repositorio local.

Así mismo la generación y evaluación de los modelos se realizará con una herramienta externa, Weka, la cuál necesita un tipo determinado de ficheros que también se describirá a continuación.

II.3.2.1. Fichero de anotaciones

El fichero de anotaciones es un fichero xml estructurado del siguiente modo que contiene el conjunto de Urls previamente catalogadas, junto con las clases a las que pertenece cada Url.

Así mismo, las Urls pueden tener un símbolo * tras ellas, lo que indica que todas las páginas que comienzan por el mismo prefijo anterior al * pertenecen a esa misma categoría. Esta característica es la que nos permite, mediante un crawl, expandir el repositorio de páginas para obtener una colección más o menos completa para realizar el entrenamiento.

El formato del fichero es el siguiente:

```
<Annotations num="426" total="426">

<Annotation about="http://www.mi-dominio.xxx/file.ext">
    <Label name="class1" />
    <Label name="class2" />
  </Annotation>

...
```

o bien

```
<Annotation about="http://www.mi-dominio.xxx/path/*">
    <Label name="class5" />
    <Label name="class1" />
  </Annotation>
...

</Annotations>
```

La primera de las anotaciones indica una url única que pertenece a las clases 1 y 2. Podría pertenecer a más clases o a una única clase, con lo que contendrá tantas etiquetas label como necesite.

La segunda de las anotaciones indica una url a partir de la cual todas las Urls que compartan ese fragmento de url, accedidas desde ella o desde cualquier otra url, pertenecerán a las clases indicadas en las etiquetas label.

Mediante la clase ObtainHtml que se diseñada en el punto 4.5.5 se realiza el recorrido de las Urls de este fichero anotado y se realiza las peticiones adecuadas a la web para obtener el conjunto de páginas necesarias para construir el repositorio. Como allí se verá, la aparición del * indicará la necesidad de realizar un crawl. En el siguiente apartado se define el fichero xml que servirá como repositorio de páginas web.

II.3.2.2. Fichero repositorio

El repositorio de páginas web se podría haber obtenido a BBDD. En un producto comercial se debería incluso dotar de la posibilidad de enlazar con diferentes BBDD comerciales. Para el propósito de la investigación la construcción del repositorio en disco, mediante un fichero xml estructurado es suficiente.

El fichero xml del repositorio deberá incluir la url extraída, junto con la clase a la que pertenece y el html de la web a la que apunta. El proceso de crawl se encargará de obtener sub-repositorios a partir de una url anotada con el símbolo * y combinarlos con el repositorio actual.

El formato del xml del repositorio es el siguiente:

```
<Repos>
  <URLS>
      <URL></URL>
      <CLASS></CLASS>
      <HTML>/<HTML>
  </URLS>
</Repos>
```

En este fichero, en cada etiqueta URL se anota una única url, sin símbolos como *. Es decir, una url anotada en el fichero descrito en el punto anterior mediante * dará lugar a tantas etiquetas Urls en el fichero repositorio como se extraigan en el proceso de crawl.

Así mismo, si una url estaba anotada para más de una categoría, aquí habrá tantas etiquetas URLS como categorías hubiera, cambiando el valor de la etiqueta CLASS y manteniendo constante el valor de la etiqueta URL y HTML. Se diseña el proceso de manera eficiente para no realizar peticiones repetidas para una misma página Web.

Con todo ello sólo queda comprobar las necesidades de recursos de este repositorio. Realizado un crawl completo hasta 3 niveles de profundidad, a partir de las 176 Urls anotadas, resulta en un total de 4801 Webs, con un tamaño total en disco de aproximadamente 73Mb, algo muy razonable para las capacidades computacionales actuales.

II.3.2.3. Fichero Arff

Como ya se ha visto en varias ocasiones el entrenamiento y posterior validación de los modelos se realizará desde la propia herramienta Weka, que proporciona muchas más posibilidades de análisis y cuya implementación escapan a los objetivos de la investigación.

Weka requiere un tipo de fichero con un formato concreto, los ficheros Arff El diseño de la estructura del fichero vendrá determinado por las necesidades específicas de dichos ficheros, pero el diseño de los tipos de datos y los valores se debe especificar en el actual punto.

Los ficheros Arff tienen dos secciones bien diferenciadas, la cabecera, dónde se indican los atributos que tendrá, y el cuerpo, dónde se enumeran los valores para cada instancia.

En la cabecera se comienza con la descripción de la RELATION. En ella se indicará un identificador para realizar dicha función identificativa del fichero.

@RELATION my-RelationId

Bajo la identificación del fichero se procede a enumerar, una por línea, todas las características (y el orden) que definirán a las instancias posteriores. La definición de las características es del tipo:

@ATTRIBUTE my-AttId my-AttType

Dónde my-AttId será el identificador que deseemos dar al atributo en cuestión, que por regla general, aunque Weka no hace uso más que para la visualización, se corresponderá con la característica a la que defina, y en el caso de una representación BoW, a la palabra en cuestión.

my-AttType es el tipo de datos admitido por Weka para representar el valor de dicha característica en cada una de las instancias. El tipo de datos a utilizar será el REAL, pues en él estamos indicando la frecuencia relativa de aparición de una determinada palabra.

Por último un atributo especial es el atributo de la clase. A Weka se le puede indicar posteriormente qué atributo realizará la función de clase, aunque es común definirlo como el último atributo de la lista. En nuestro caso lo hemos denominado class y el tipo de datos será una enumeración:

@ATTRIBUTE class {0, 1}

Aunque por motivos históricos, a partir de los cuales nace el proyecto de investigación actual, aquí se enumeraban todas y cada una de las categorías. Por ello es necesario, a excepción del generador de características de los Blogs, realizar una binarización del fichero una vez extraído. Esto se explicará en el diseño de BinarizeArff.

A continuación se define el comiendo de la sección de datos, mediante:

DATA

Y se continúa enumerando los mismos, para lo cuál se anota, separado por comas, cada uno de los valores de la instancia actual para la característica definida en la posición que toca, así como por último el valor de la clase, como un valor más, pues así lo toma Weka.

En la aplicación actual el tamaño de estos ficheros para las representaciones BoW es bastante elevado, dependiendo del numero de características, que por su parte depende del número de palabras del corpus utilizado; así pues existen ficheros arff de

hasta 43Mb, por contener la definición de más de 3000 característica y sobre 4000 instancias.

En cuanto al método propuesto en la investigación, basado en H&L&U, obtiene ficheros Arff ligeramente más pequeños, sobre 30Mb, definiendo aproximadamente 707 características y sobre 4000 instancias también.

Por último los ficheros generados para la representación específica de los Blogs tienen un tamaño mucho más manejable, sobre 300Kb, porque aunque definen también aproximadamente 4000 instancias, únicamente requieren 15 atributos para ser definidos, lo que limita enormemente el tamaño y acelera en gran medida el aprendizaje.

En cualquier caso, la utilización de Weka y del algoritmo Naïve Bayes, además de conseguir un buen aprendizaje, obtiene buenas velocidades en la creación de los modelos que hacen que el tratamiento de los mismos no sea un proceso tedioso y lento.

II.3.3. Diseño del stemmer de Porter

El proceso de tratamiento lingüístico ha quedado bastante claro en diversos artículos [Forman][Guyon] que es necesario no sólo para reducir la dimensionalidad del problema, sino además para obtener mejores resultados, reuniendo palabras con una misma raíz que de otro modo aparecerían por separado, dividiendo así su importancia.

El tratamiento lingüístico es algo complejo y lejos del alcance de la investigación por lo que se ha acudido al estado del arte para obtener algún tipo de aplicativo que lo contemple.

Una manera de relajar este análisis lingüístico es mediante los denominados algoritmos de stem, que mediante una serie de reglas computacionales reducen las palabras a sus lemas comunes mediante eliminación de prefijos, sufijos y características similares.

En la literatura es común oír hablar del stemmer de Porter, que es un proceso para eliminar las finalizaciones morfológicas e inflexivas más comunes de las palabras en idioma inglés. Su principal uso ha sido para la normalización de términos en las tareas de recuperación de información.

La implementación del algoritmo de [Porter] está realizada en diversos lenguajes de programación. Así mismo aparecen variaciones para contemplar otras lenguas diferentes del inglés, como en nuestro caso el castellano.

En un repositorio sobre recursos lingüísticos relacionados con este área se encuentra una serie de ficheros, por idiomas, dónde se relacionan par a par las palabras del idioma con su raíz previamente *stemmizada,* por lo que se decide utilizar este fichero, cargándolo como una tabla hash de acceso rápido y directo, en lugar de implementar ningún algoritmo.

Puesto que la clase BoWMngr es la que encapsula la clase del stemmer, sería rápidamente modificable para incorporar un stemmer hecho a medida, tarea que se deja como ampliación futura.

II.4. Diseño de las características

En el análisis de alternativas se estudió las diferentes aproximaciones a la representación del problema según el estudio del estado del arte. Se describió cada una de ellas y se disertó acerca de la idoneidad de las mismas para ser usadas comparativamente en la aproximación seguida, así como se describió brevemente la aproximación que se iba a seguir de manera general, y la aproximación específica para la clasificación de las páginas de tipo blog.

En el análisis del sistema se revisó las características de los diferentes tipos de páginas, observando sus propiedades y atributos más representativos, mediante un análisis del conjunto de corpus obtenidos para los mismos, la frecuencia de aparición de las palabras y su comparación con un corpus general del dominio del teatro. Se hizo hincapié en la necesidad de obtener aquellas características suficientemente representativas de la categoría actual y que además fuesen suficientemente diferenciadoras del resto de categorías, y sobre la base de las observaciones, se defendió el por qué podría llegar a ser buena la representación h&l&u propuesta. Así mismo en aquél punto se realizó un análisis detallado de las características de las páginas web de tipo blog que serviría para diseñar una representación formal de los mismos capaz de obtener resultados superiores al resto de representaciones.

En el apartado actual de diseño de las características se expondrán las decisiones de diseño referentes a las palabras del corpus que serán tomadas en cuenta y el modo de extraerlas y ponderarlas. Así mismo, se hará especial hincapié en el diseño de las características definitorias de los Blogs y cómo se obtendrán para construir un modelo de aprendizaje inductivo.

II.4.1. Características BoW: BoW Standar, BoW Improv y BoWUrl

La representación BoW consistirá en obtener las palabras de las páginas en cuestión que formen parte del corpus dado para la categoría actual, obteniendo una frecuencia de aparición y formando esta el valor de la categoría en cuestión.

Ahora bien existen diversos modos de obtener estas palabras así como diversos modos de ponderar el valor de las diferentes características.

II.4.1.1. ¿Obtener palabras o contar apariciones?

La obtención de las palabras es una tarea relativamente sencilla. Con la aplicación de una expresión regular, previamente eliminadas las etiquetas html, se extrae el conjunto de palabras de un documento.

La expresión regular concreta es la siguiente:

```
\b(?<word>[a-zñáàéèíóòúëï]+)\b
```

En ella se observa que se obtiene la aparición 1 o más veces de todo aquél conjunto delimitado por separadores de caracteres alfabéticos, con y sin acentos y con o sin diéresis. No se toman palabras que contengan números ni signos de cualquier tipo.

Mediante un proceso de stemming se obtiene el conjunto de lemas de la misma y se compara con el conjunto de lemas del corpus, anotando la frecuencia de aparición de cada palabra del corpus dividiendo por el número total de palabras.

El proceso es sencillo, pero con las Urls tiene una pega. Las Urls no suelen estar formadas por palabras completas y separadas por separadores estándar, como puedan ser en un documento textual la coma, el punto, el punto y coma, el espacio, etcétera. Es muy común que una url sea del tipo revistadeteatro.com o festivaldemerida.es/festivalesdeverano.

En estos casos el proceso de extracción de palabras resultaría en tres palabras únicamente, *revistadeteatro*, *festivaldemerida* y *festivalesdeverano*, que por otro lado, al no tener correspondencia en el stemmer, se perderían.

Por ello, en la comparación con las Urls es más interesante el método de comparar que de obtener. Comparar significa obtener cada una de las palabras del corpus y contar el número de apariciones en el conjunto de palabras de la url. De este modo, la palabra revista, o su lema revist, aparecería una vez en la primera Url y la palabra festival o su lema festiv aparecería dos.

Por lo tanto, para la obtención de las características (palabras) de los documentos se utilizará el primer método (la aplicación de la expresión regular) y para la obtención de las características de los enlaces y la url se utilizará el segundo (contar el número de apariciones)

La implementación de todas estas opciones se realizará en la clase HtmlMngr.

II.4.1.2 ¿Ponderar o duplicar?

Cuando las características se toman de diferentes sitios de un documento, como por ejemplo las palabras del texto plano, del título, de la url, los enlaces o los encabezados, se debe tomar una decisión, ponderar los valores, mediante alguna distribución de pesos, para cada uno de los lugares anteriores o duplicar tantas veces como lugares haya el conjunto de las características.

En el primero de los casos, el valor de una característica equivaldrá a la suma ponderada de los resultados obtenidos en cada lugar:

$$X(i) = f(links)*w(links) + f(body)*f(body) + ...$$

En el segundo de los casos cada uno de estos lugares tendrá una característica:

$$XLinks(i) = f(links)$$
$$XBody(i) = f(body)$$
$$...$$

En el estado del arte estudiado parece que se utilizan ambas representaciones. La ventaja de la ponderación es que reduce considerablemente la dimensionalidad respecto de la representación duplicada. Su inconveniente es que al mezclar datos disminuye la

capacidad de los modelos inductivos de encontrar patrones que podría encontrar duplicando las características.

Por ello la decisión de utilizar la mezcla en la representación BoWImprov y la de triplicar las características en la representación H&L&U (descrita a continuación), de modo que se mantenga una dimensionalidad similar, ya que H&L&U se construye a partir de un corpus mucho más reducido, el obtenido con las palabras de la cabecera, los enlaces y la url.

II.4.1.3. Diseño de las tres representaciones

- La representación BoWStd se realiza a partir de la obtención, mediante la expresión regular, de las palabras que forman el conjunto de su cuerpo
- La representación BoWUrl se realiza a partir de la comparación de las palabras que forman su Url
- La representación BoWImprov se realiza a partir de la obtención, mediante la expresión regular, de las palabras que forman parte del contenido de su cuerpo y se pondera, con un valor igual a 1 para todas ellas, las palabras encontradas en el cuerpo, el título y los encabezados h1, h2 y h3.

II.4.3. Características H&L&U

La representación H&L&U es una representación BoW a partir del corpus obtenido desde la cabecera, los enlaces y la url del documento, previo proceso de stem y filtrado por stop word list, al igual que los anteriores.

El conjunto de características es mucho más reducido, como se vio anteriormente, se puede pasar de un corpus de unas 3000 palabras a uno de apenas 700. De este modo la duplicación, en este caso triplicación, de características no supone un incremento considerable manteniéndose en aproximadamente la misma cantidad que las representaciones anteriores.

Las características H&L&U se obtendrán de la siguiente manera:

En primer lugar se obtiene el conjunto de html de la cabecera del documento, mediante la expresión regular siguiente:

```
<head[^>]*>((.|\r|\n|\t)*?)(</head>|<body)
```

Puesto que muchos diseñadores olvidan, quizás intencionadamente, cerrar la etiqueta head, para evitar un time out del parser regex se permite el cierre de dicha etiqueta con la apertura del cuerpo del documento.

Una vez obtenidas el conjunto de palabras se procede a su separación mediante la expresión regular definida anteriormente para obtener palabras, y se crea el conjunto de características como el nombre de la misma seguido por la palabra HEAD.

Las palabras de la url se obtienen comprobando la ocurrencia de cada palabra del corpus en la misma, rellenándose con ellas las características con su mismo nombre seguidas de la palabra URL.

Para los enlaces se realiza la obtención desde dos propiedades de los documentos, concretamente de los enlaces que contiene, su texto y su url. Para ello se recorre el conjunto de enlaces de la página web y se obtiene por el primer método, el de la expresión regular, el conjunto de palabras de los mismos, y por el segundo método, el de comparar la aparición, el número de apariciones de las palabras en la URL. Ambos valores (frecuencias, al dividirse entre el número total de palabras) se suman, con una ponderación igual a uno, de modo que se toman con la misma importancia ambos, y se constituyen como una nueva característica denominada el nombre de la palabra seguida por LINK.

Con ello queda definida la estructura de la representación y el modo de obtenerla.

II.4.4. Características BlogSpecific

El diseño de las características que formarán parte de la representación específica de los Blogs viene directamente determinado por su análisis. Quizás más interesante será cómo obtener ciertas características.

En un principio el análisis realizado para la representación de los Blogs parecería indicar la comparación de plantillas, determinando si se adecua o no a la estructura dada.

En realidad determinar esta comprobación sería una tarea compleja ya que, aunque todos los Blogs comparten más o menos estas características, las formas de combinarlas pueden ser muchas y muy variadas.

Por ello se obtiene una representación en base a frecuencia de aparición de determinados elementos así como en ratios de aparición de unos frente a los otros.

En primer lugar es interesante enumerar las características a obtener, 15 en total, para describirlas y ver qué se necesita para obtenerlas. En segundo lugar se diseñará el modo de obtenerlas y en tercer lugar la manera en que algunas de ellas se componen mediante la combinación con otras.

Las características son:

- NBLOGINURL: Número de veces, o mejor dicho frecuencia de aparición de la palabra blog en la URL. Para obtenerlo se comprueba el número de apariciones de la palabra a lo largo de la URL y se divide entre el número total de palabras de que consta la misma.

- NBLOG: Frecuencia de aparición de la palabra blog en el documento. Para ello se obtiene, mediante la expresión regular de las palabras, las palabras que componen el documento y se obtiene la frecuencia de la palabra blog en las mismas.

- NPOST: Lo mismo que lo anterior pero con la palabra POST

- NRSS: Lo mismo pero con las palabras RSS y ATOM. La aparición de ambas se suma, suelen ser disjuntas por lo que si aparece una no aparecerá la otra, no alterando de este modo la frecuencia.

- CommentsVsDates: Ratio entre los comentarios aparecidos en la página y las fechas aparecidas. Para ello se hace uso de las siguientes expresiones regulares, para obtener los comentarios, en diferentes formatos y lenguas, y las fechas, también en diferentes formatos cortos y largos, así como de localización:

  Para los comentarios:

```
(?<comment>((\[?[0-9]*\\]?
*(comments|comentarios))|((comment|comentario) *\[?[0-9]*\]?)))
```

  Para las fechas:

```
// week day, month day, year
(?<date>(monday|tuesday|wednesday|thursday|friday|saturday|sunday)?,?
*(january|february|march|april|may|june|july|august|september|october|
november|december) *(0?[1-9]|[12][0-9]|3[01]),? *(19|20)?\\d\\d)

?<date>(monday|tuesday|wednesday|thursday|friday|saturday|sunday)?,?
*(jan|feb|mar|apr|may|jun|jul|aug|sep|oct|nov|dec) *(0?[1-9]|[12][0-
9]|3[01]),? *(19|20)?\\d\\d)

// día de mes de año
(?<date>(0?[1-9]|[12][0-9]|3[01]) *(-|de)?
*(enero|febrero|marzo|abril|mayo|junio|julio|agosto|septiembre|octubre
|noviembre|diciembre) *(-|de)? *(19|20)?\\d\\d)

(?<date>(0?[1-9]|[12][0-9]|3[01]) *(-|de)?
*(ene|feb|mar|abr|may|jun|jul|ago|sep|oct|nov|dic) *(-|de)?
*(19|20)?\\d\\d)

// DD/MM/AA
(?<date>(0?[1-9]|[12][0-9]|3[01])(/|-)(0[1-9]|1[1|2])(/|-
)(19|20)?\\d\\d)

// MM/DD/YY
(?<date>(0[1-9]|1[1|2])(/|-)(0?[1-9]|[12][0-9]|3[01])(/|-
)(19|20)?\\d\\d)
```

- LinkCommentsVsDates: Ratio entre los comentarios, obtenidos mediante la expresión regular anterior, que forman parte de un enlace y las fechas, obtenidas también mediante las expresiones regulares anteriores.

- LinkCommentsVsComments: Ratio entre los comentarios que forman parte de un enlace y el total de comentarios, obtenidos ambos mediante las expresiones regulares anteriores.

- CommentsRatio: Ratio de aparición de comentarios frente al número de encabezados de posts.

Esta característica, al igual que las dos siguientes, se obtiene por comparación con los tres tipos de encabezados que suelen aparecer en los Blogs, H1, H2 y H3, realizando el ratio con aquél de ellos cuya cantidad de apariciones sea más cercana al de la característica, es decir, si se obtienen los tres ratios posibles, nos quedamos con el más cercano a cero.

Esto es así porque en cualquier blog suelen haber tres tipos de encabezados, el del blog en general, el de cada post y el de otros menús como las islas de enlaces, u otros resaltados. Pero esto puede variar, con lo que la aplicación de los ratio puede servir de más ayuda a un modelo probabilístico de aprendizaje como el Naïve Bayes.

- LinkCommentsRatio: Ratio de aparición de comentarios en enlaces frente al número de encabezados de posts.

- DatesRatio: Ratio de aparición de fechas frente al número de encabezados de posts.

- HasIslands: Indica la aparición de islas de enlaces. Para ello hace uso de la siguiente expresión regular, que si devuelve alguna coincidencia indicará un valor positivo, 1, y en caso contrario negativo, 0.

```
<ul[^>]*>((.|\r|\n|\t)*?)</ul>
```

A partir de los enlaces aquí obtenidos se calculan las siguientes características

- NILinksRatio: Ratio entre el número de enlaces hacia el propio dominio aparecidos en las islas de enlaces frente al número total de enlaces de la página

- NOLinksRatio: Ratio entre el número de enlaces hacia fuera del dominio aparecidos en las islas de enlaces frente al número total de enlaces de la página

- NIOLinksRatio: Ratio entre el número de enlaces dentro y fuera del dominio aparecidos en islas de enlaces frente al número total de enlaces de la página

La implementación de lo aquí descrito se encuentra en la clase ClsBlog que hace uso de la clase HtmlMngr para obtener las características necesarias del Html.

II.5. Diseño de los objetos del sistema

Una vez definida la interfaz con el usuario, los ficheros necesarios para la realización de las tareas y las diferentes representaciones formales a obtener, se procede a diseñar cada una de las clases que realizarán las tareas necesarias para el tratamiento, creación y utilización de los anteriores ficheros y con ello para guiar hacia la consecución de los objetivos de la investigación.

II.5.1. Diseño del Crawler

La primera clase a diseñar es la de extracción de páginas desde Internet, capaz de recorrer, dada una url, todos los enlaces y hacer un crawl obteniendo el html de los enlaces visitados.

Para ello se diseña un proceso recursivo que recibe como parámetros la página base o fragmento de url que debe aparecer en el inicio de cada url para ser incluido en el repositorio, la url a extraer, la clase a la que pertenecen el conjunto de Urls que se extraigan, la profundidad actual y el máximo de profundidad que se debe alcanzar.

El proceso irá construyendo un repositorio con la información extraída de cada una de las páginas y devolverá el mismo al final del todo.

Su definición algorítmica es:

```
DataSet Crawl (baseUrl, currentUrl, urlClass, currentDepth, maxDepth)
        uriBase <- ObtainUri (baseUrl)
        if currentDepth <= maxDepth
                if Visited(currentUrl)
                        Visited(currentUrl = true
                        uriCurr <- ObtainUri(currentUrl)
                        if uriBase.Host = uriCurr.Host y currentUrl.StartsWith(baseUrl)
                                html <- GetHtml(currentUrl)
                                if html not empty
                                        Repos <- Repos + AddUrl(currentUrl, html, urlClass)
                                endif
                        endif
                        if html not empty
                                links = HtmlMngr.GetLinks(shtml)
                                foreach link in links
                                        if  link not null
                                                Crawl (baseUrl, link, urlClass, currentDepth + 1)
                                        endif
                                endforeach
                        endif
                endif
        endif
        return Repos
end Crawl
```

Los métodos ObtainUri y ObtainHtml se desarrollan para facilitar las tareas de extracción y únicamente hacen uso de las clases de .Net para tratamiento de Urls y obtención de recursos http.

II.5.2. Diseño del HtmlMngr

La clase HtmlMngr será con diferencia la clase que mayor funcionalidad aporte. Es la clase base que será utilizada por todas aquellas clases que requieran extraer conocimiento del contenido de una página web. La mayoría de sus métodos son públicos, como se vio en el diagrama de objetos, y además estáticos, permitiendo un acceso rápido sin necesidad de instanciación del objeto.

Los métodos proporcionados son los siguientes. En caso que sea necesario se adjuntará un algoritmo que clarifique la explicación, aunque en la mayoría de casos ésta es autosuficiente para describir el comportamiento del mismo, ya que su funcionalidad es muy acotada y simple:

- GetHtmlDoc: Obtiene un documento html a partir de un texto en html, para su mejor tratamiento mediante métodos y atributos propios de la clases mshtml de .Net

  ENTRADA: Cadena de texto con el html
  SALIDA: Documento html de la clase mshtml

- TokenizeUrl: Elimina los tokens que no aportan información de las Urls como ? & _ - / : acentos, números y elementos sustitutos de acentos y demás...

  ENTRADA: Cadena de texto con la url
  SALIDA: Cadena de texto con los caracteres anteriores eliminados

- GetBISHList: Obtiene un listado de las palabras que aparecen bajo el tag de encabezado indicado

  Hace uso de la expresión regular:

  ```
  <%%1%%[^>]*>((.|\r|\n|\t)*?)</%%1%%>
  ```

  reemplazando sucesivamente %%1%% por el tag B, I, S, H1, H2, H3 respectivamente y obtiene el conjunto de palabras que lo forman.

  ENTRADA: Cadena de texto con el html, texto con el tag a reemplazar
  SALIDA: Cadena de texto con las palabras que lo forman

- GetLinks: Obtiene el conjunto de Urls que aparecen en los enlaces de la página dada

  ENTRADA: Cadena de texto con el documento html
  SALIDA: Lista de cadenas de texto con las Urls de los enlaces que lo forman

- GetHead: Obtiene el fragmento de html definido entre <head></head>

  Hace uso de la expresión regular de obtención de la cabecera:

  ```
  <head[^>]*>((.|\r|\n|\t)*?)(</head>|<body)
  ```

  ENTRADA: Cadena de texto con el documento html

SALIDA: Lista de cadenas de texto con las cabeceras encontradas (generalmente 1)

- GetNWordsBody: Obtiene el número de palabras que conforman el cuerpo del html

  ENTRADA: Cadena de texto con el html
  SALIDA: Entero con el número de palabras que la forman

- GetNWordsUrl: Obtiene el número de palabras que conforman la url, previa tokenización

  ENTRADA: Cadena de texto con la url
  SALIDA: Entero con el número de palabras que la forman

- GetNWordsLinks: Obtiene el número de palabras de los enlaces de la página

  ENTRADA: Cadena de texto con el documento html
  SALIDA: Entero con el número de palabras que forman el texto de los enlaces

- GetNWords: Dada una lista de palabras o un texto obtiene el número de palabras individuales que lo componen

  Hace uso de la expresión regular de obtención de palabras:

  ```
  \b(?<word>[a-zñáàéèíóòúëï]+)\b
  ```

  ENTRADA: Cadena de texto
  SALIDA: Entero con el número de palabras que forman el texto

- GetNumOccurs: Devuelve el número de ocurrencias de un texto en otro texto

  ENTRADA: Cadena de texto con el texto y con el subtexto a buscar en el texto
  SALIDA: Número de veces que aparece el subtexto en el texto

- GetHtmlTag: Obtiene el fragmento de html contenido dentro de los tags dados como parámetro

  Hace uso de la expresión regular de obtención de tags, reemplazando %%1%% por el tag dado:

  ```
  <%%1%%[^>]*>((.|\r|\n|\t)*?)</%%1%%>
  ```

  ENTRADA: Cadena de texto con el documento html. Cadena con el tag a buscar
  SALIDA: Texto contenido entre los tags indicados

- GetNHtmlTag: Obtiene el número de elementos definidos por un tag dado que existen en el documento html

  Hace uso de la expresión regular de los tags:

  ```
  <%%1%%[^>]*>((.|\r|\n|\t)*?)</%%1%%>
  ```

ENTRADA: Cadena de texto con el documento html. Cadena con el tag a buscar
SALIDA: Entero con el número de elementos definidos con el tag.

- GetNComments: Obtiene el número de elementos comentario que aparecen en el documento html

  Hace uso de la expresión regular de obtención de los comentarios:

```
(?<comment>((\[?[0-9]*\]?
*(comments|comentarios))|((comment|comentario)          *\[?[0-
9]*\]?)))
```

  ENTRADA: Cadena de texto con el html
  SALIDA: Número de comentarios

- GetNCommentsInLinks: Obtiene el número de elementos comentario que aparecen en los enlaces del documento html

  Hace uso de la expresión regular de obtención de los comentarios:

```
(?<comment>((\[?[0-9]*\]?
*(comments|comentarios))|((comment|comentario)          *\[?[0-
9]*\]?)))
```

  ENTRADA: Cadena de texto con el html
  SALIDA: Número de comentarios que están dentro de enlaces

- GetNLinks: Obtiene el número total de enlaces que hay en un documento html, así como los que son al propio dominio y los que son a dominios externos

  ENTRADA: Cadena de texto con el html
  SALIDA: Número de enlaces

- GetNLinksIslands: Obtiene las islas de enlaces del documento html, devolviendo un booleano indicando la existencia de alguna, el número de enlaces totales que contienen las mismas, el número de enlaces a páginas internas del dominio y el número de enlaces externas al dominio

  ENTRADA: Cadena de texto con el html
  SALIDA: Booleano indicando que es una isla
  Entero con el número de enlaces hacia el propio dominio
  Entero con el número de enlaces hacia fuera del dominio
  Entero con el número de enlaces total

- GetNDates: Obtiene el número de fechas contenidas en el documento, teniendo en cuenta diferentes formatos de las mismas

  Hace uso de la expresión regular de obtención de las fechas:

```
// week day, month day, year
(?<date>(monday|tuesday|wednesday|thursday|friday|saturday|sunday)?,?
*(january|february|march|april|may|june|july|august|september|october|
november|december) *(0?[1-9]|[12][0-9]|3[01]),? *(19|20)?\\d\\d)

?<date>(monday|tuesday|wednesday|thursday|friday|saturday|sunday)?,?
*(jan|feb|mar|apr|may|jun|jul|aug|sep|oct|nov|dec) *(0?[1-9]|[12][0-
9]|3[01]),? *(19|20)?\\d\\d)

// día de mes de año
(?<date>(0?[1-9]|[12][0-9]|3[01]) *(-|de)?
*(enero|febrero|marzo|abril|mayo|junio|julio|agosto|septiembre|octubre
|noviembre|diciembre) *(-|de)? *(19|20)?\\d\\d)

(?<date>(0?[1-9]|[12][0-9]|3[01]) *(-|de)?
*(ene|feb|mar|abr|may|jun|jul|ago|sep|oct|nov|dic) *(-|de)?
*(19|20)?\\d\\d)

// DD/MM/AA
(?<date>(0?[1-9]|[12][0-9]|3[01])(/|-)(0[1-9]|1[1|2])(/|-
)(19|20)?\\d\\d)

// MM/DD/YY
(?<date>(0[1-9]|1[1|2])(/|-)(0?[1-9]|[12][0-9]|3[01])(/|-
)(19|20)?\\d\\d)
```

ENTRADA: Cadena de texto con el html
SALIDA: Número de fechas

- GetBodyBow: Obtiene la representación BoW en una tabla hash para el contenido del cuerpo del documento html con la ponderación dada

  ENTRADA: Cadena de texto con el html. Tabla hash con BoW previamente extraído. Peso para ponderar.
  SALIDA: Tabla hash con el BoW obtenido a partir de las palabras del cuerpo

- GetTitleBoW: Obtiene la representación BoW en una tabla hash para el contenido del título del documento html con la ponderación dada

  ENTRADA: Cadena de texto con el html. Tabla hash con BoW previamente extraído. Peso para ponderar
  SALIDA: Tabla hash con el BoW obtenido a partir de las palabras del título

- GetHeadBoW: Obtiene la representación BoW en una tabla hash para el contenido del encabezado del documento html con la ponderación dada

  ENTRADA: Cadena de texto con el html. Tabla hash con un BoW previamente obtenido. Pero para ponderar
  SALIDA: Tabla hash con el BoW obtenido a partir de las palabras de la cabecera

- GetUrlBoW: Obtiene la representación BoW en una tabla hash para el texto contenido en la url del documento html con la ponderación dada

ENTRADA: Cadena de texto con la url. Tabla hash con un BoW previamente obtenido. Peso para ponderar.
SALIDA: Tabla hash con el BoW obtenido a partir de la url

- GetLinksUrlsBoW: Obtiene la representación BoW en una tabla hash para el texto de la url de los enlaces que contiene el documento html con la ponderación dada

  ENTRADA: Cadena de texto con el documento html. Tabla hash con un BoW previamente obtenido. Peso para ponderar.
  SALIDA: Tabla hash con el BoW obtenido a partir del texto de las Urls de los enlaces

- GetLinksBoW: Obtiene la representación BoW en una tabla hash para el texto de los enlaces que contiene el documento html con la ponderación dada

  ENTRADA: Cadena de texto con el documento html. Tabla hash con un BoW previamente obtenido. Peso para ponderar.
  SALIDA: Tabla hash con el BoW obtenido a partir del texto de los enlaces

- GetBISHBoW: Obtiene la representación BoW en una tabla hash para el texto contenido en la etiqueta de encabezado indicada con la ponderación dada

  ENTRADA: Cadena de texto con el documento html. Tabla hash con un BoW previamente obtenido. Cadena con el tag para el cuál obtener el BoW. Peso para ponderar.
  SALIDA: Tabla hash con el BoW obtenido a partir del texto de las palabras extraídas del tag dado.

- GetBoW: Obtiene la representación BoW en una tabla hash para una combinación ponderada de todas las representaciones anteriores para el documento html dado.

  ENTRADA: Cadena de texto con el documento html. Cadena de texto con la url. Tabla hash con un BoW previamente obtenido. Conjunto de pesos para las diferentes etiquetas.
  SALIDA: Tabla hash con el BoW obtenido a partir del texto de los siguientes elementos:

  - Cuerpo
  - Título
  - Url
  - H1
  - H2
  - H3
  - B
  - I
  - Enlaces
  - Url enlaces

Con las ponderaciones dadas para cada uno. Hace uso de todos los GetXXXBoW anteriores.

II.5.3. Diseño del ArffMngr

El gestor de ficheros Arff es de implementación sencilla. Su única función es permitir, a partir de un conjunto de características dadas generar un encabezado arff, y a partir de los valores añadir las líneas de las instancias al mismo.

Por ello proporciona los cuatro métodos siguientes:

- CreateArffHeader: Crea la cabecera de un fichero Arff con el conjunto de características determinada por el corpus facilitado ordenándolo alfabéticamente. Existen dos modalidades, la creación simple, con una característica por palabra, y la modalidad h&l&u con tres características por palabra, tal y como se definió en el diseño de las representaciones.

ENTRADA: Tabla hash con el corpus o conjunto de características. Modalidad.
SALIDA: Tablas hash ordenada alfabéticamente

- CreateArffHeaderBlogs: Crea la cabecera de un fichero Arff para las características específicas de la representación de los Blogs.

No tiene entradas ni salidas

- AddArffData: Añade una línea de datos a partir de los valores dados para el corpus. Dependiendo de la modalidad añadirá los valores de la tabla hash oportuna.

ENTRADA: Tabla hash con los valores del cuerpo, tabla hash con los de la cabecera, tabla hash con los de las Urls.
SALIDA: Nada

- AddArffDataBlogs: Añade una línea de datos a partir de los valores concretos para un blog dado.

ENTRADA: Valores para el blog
SALIDA: Nada

II.5.4. Diseño del BoWMngr

La clase BoWMngr servirá de apoyo para la obtención del vector de palabras a partir del texto facilitado. En ella se implementará el gestor de palabras vacías así como el stemmer lingüístico.

El método que publica GetBoW se encarga de obtener un vector de palabras a partir de un texto dado. Tiene varias alternativas, así pues se puede obtener un vector de palabras sumándolo a otro vector previamente obtenido, se puede obtener con valores

ponderados mediante una asignación de pesos y se puede obtener eliminando o manteniendo aquellas palabras que no aparezcan en el stemmer.

Hace uso de la expresión regular para obtener palabras descrita en otros puntos:

```
\b(?<word>[a-zñáàéèíóòúëï]+)\b
```

Por lo tanto el conjunto de sus entradas y salidas es el siguiente:

ENTRADA: Cadena de texto con el documento html
Tabla hash con un vector previamente obtenido
Peso asignado a las palabras actuales
Booleano indicando si se mantendrá o eliminará la palabra no encontrada
SALIDA: Tabla hash con el vector de palabras

Su algoritmo es el siguiente:

```
GetBoW
        if not exists hash
                create hash
        matches <- regex.MatchWords (html)
        ival <- weight / matchex.count
        foreach match in matches
                word <- match.word
                word <- normalize(word)
                if (word isn't stop word)
                        if hash contains word
                                hash(word) <- hash(word) + ival
                        else
                                hash (word) <- ival
                endif
        endforeach
        return hash
End GetBoW
```

Los métodos normalize y isstopword son los encargados respectivamente de aplicar el proceso de stem y comprobar si la palabra obtenida es una palabra vacía.

El proceso de stem consiste en algorítmicamente obtener la raíz común de las palabras. En la implementación del resultado se ha utilizado una tabla hash que contiene las correspondencias, previamente obtenidas por un proceso de stem, entre la palabra completa y su raíz. Esta lista de correspondencias se ha obtenido de [Snowball] y se ha ampliado y modificado manualmente a las necesidades concretas del proyecto.

II.5.5. Diseño del ObtainHtml

La interfaz para obtener los documentos html debe proporcionar la posibilidad de elegir el origen de las Urls anotadas así como el destino del respositorio extraído, y realizar un proceso iterativo de crawl de las diferentes Urls, reportando un estado de progreso al usuario.

Todo ello se define mediante el siguiente algoritmo, utilizado para la implementación del método Obtain de la clase.

```
Obtain
        foreach Annotation in UrlsAnnotarions
                url <- Annotation.Url
                foreach class in Annotation
                        if url.EndsWith(“*”)
                                Repos <- Repos + Crawl(url, url, class, 0, maxDepth)
                        else
                                Repos <- Repos + Crawl(url, url, class, 0, 0)
                        endif
                        ShowProgress
                        ShowStatistics
                endForeach
        endForeach
        ShowEndMessage
End Obtain
```

II.5.6. Diseño del UrlsViewer

El UrlsViewer es una clase de interfaz que únicamente visualiza el contenido del repositorio y permite su manipulación. Con la UI actual del sistema .Net toda la interacción y manejo de los datos, así como su lectura y almacenamiento quedan totalmente manejados y libres de programación específica, por lo que no merece la pena ni su descripción algorítmica por tratase de cuatro llamadas básicas.

Sí merece en cambio la pena nombrar la necesidad de eliminar duplicados, sobre todo tras un mezclado de datos, por lo que se provee de un método que será invocado al almacenar el respositorio que lo recorre y si encuentra duplicados los elimina. Para ello ordena el repositorio por url y clase y para cada línea comprueba si previamente existió, y en caso afirmativo la elimina y pasa a la siguiente. Es un proceso secuencial de rápida ejecución, por lo que no se incide en más en su mejora.

II.5.7. Diseño del GenerateCorpus

La interfaz de generación de corpus es una de las primeras a desarrollar, tras la obtención del html, pues es la que servirá de base para todo el estudio y para obtener las representaciones enumeradas.

Su cometido es obtener, a partir de un repositorio de Webs dadas, el conjunto de palabras que la forman, realizando el proceso de stem y stop necesario, y generando de este modo el corpus base sobre el que generar los vectores de palabras.

El algoritmo se describe a continuación:

```
GenerateCorpus
        foreach web in Repos
                url <- web.url
                html <- web.url
                class <- web.class

                if method = body
                        corpus <- corpus + HtmlMngr.GetBodyBoW(html)
                else if method = l&h&u
                        corpus <- corpus + HtmlMngr.GetLinksBoW(html)
                        corpus <- corpus + HtmlMngr.GetLinksUrlsBoW(html)
                        corpus <- corpus + HtmlMngr.GetURLBoW(url)
```

```
                    corpus <- corpus + HtmlMngr.GetHeadBoW(html)
                end if
                ShowProgress
        endforeach
        ShowEndMessage
End GenerateCorpus
```

II.5.8. Diseño de EstadisticaClases

Esta clase se limita a leer el repositorio de Webs y obtener el número de Webs de cada clase, de manera que el usuario se pueda hacer una idea de la distribución de las mismas.

El resultado final para el repositorio utilizado en la investigación se adjunta en el ANEXO I

II.5.9. Diseño del ArffCreator_BoW

Esta es la clase encargada de obtener las diferentes representaciones BoW enumeradas en las alternativas. Así pues servirá para crear el BoW estándar, utilizando para ello la clase del HtmlMngr para obtener el BoW a partir del cuerpo del documento, el BoW mejorado, utilizando entonces la clase del HtmlMngr para obtener el BoW a partir de diferentes etiquetas del documento, con una ponderación previamente definida (en este caso la definida en el punto 4.4.1.3, el BoW Url mediante el método de HtmlMngr para obtener el BoW a partir de la Url, y el BoW H&L&U mediante los métodos para obtener el BoW a partir de la cabecera, los enlaces (texto y url) y la url del documento.

Con todo ello recorrerá el repositorio de Webs, obtendrá la representación indicada, mostrará el progreso al usuario y generará el fichero Arff adecuado.

El algoritmo es el siguiente:

```
ArffCreator_BoW
        ArffMngr.CreateArffHeader
        foreach web in repos
                url <- web.url
                html <- web.html
                class <- web.class
                switch (BoWOption)
                        case Standar:
                                body <- HtmlMngr.GetBodyBoW(url, html)
                        case Improved:
                                body <- HtmlMngr.GetBoW(url, html, BoWOptions)
                        case Urls:
                                Urls <- HtmlMngr.GetUrlBoW(url)
                        case H&L&U:
                                head <- HtmlMngr.GetHeadBoW(html)
                                links <- HtmlMngr.GetBoW(url, html, BoWOptions)
                                Urls <- HtmlMngr.GetUrlBoW(url)

                end switch
                ArffMngr.AddArffData(body, head, links, Urls)
                ShowProgress
        endforeach
        ShowEndMessage
```

```
End ArffCreator_BoW
```

II.5.10. Diseño del ArffCreator_Blogs

El caso de la creación de los Blogs es similar, pero en lugar de obtener representaciones BoW obtendrá las representaciones específicas para los mismos, para lo cuál utilizará la clase ClsBlog que se define al final del apartado y que será capaz de obtener los valores para cada una de las características diseñadas en el apartado 4.4.4

El algoritmo es el siguiente:

```
ArffCreator_Blogs
        ArffMngr.CreateArffHeader
        foreach web in repos
                url <- web.url
                html <- web.html
                class <- web.class
                blogAtt <- ClsBlogs.GetBlogAtt(url, html, class)
                ArffMngr.AddArffData(blogAtt)
                ShowProgress()
        endforeach
        ShowEndMessage
End ArffCreator_Blogs
```

II.5.11. Diseño del Binarizador

La binarización es el proceso de obtener a partir de un fichero Arff general para N categorías, N ficheros Arff binarios con el valor de clase a 1 para la categoría actual y 0 para el resto.

Como ya se indicó, este proceso viene históricamente de las primeras investigaciones en un único clasificador global.

Su algoritmia se describe a continuación:

```
Binarize
        file <- File to Binarize (filename selected by user)
        foreach class in classes
                newFile <- file.replace(class, “1”)
                foreach class2 in classes
                        if  class2 <> class
                                newFile <- file.replace(class2, “0”)
                        endif
                endforeach
                SaveFile(newFile, “filename.class.arff”)
        endforeach
        ShowEndMessage
End Binarize
```

II.5.12. Diseño del Navigator

El navegador Web no tiene mayor misterio que además, en paralelo, realiza una clasificación binaria de la página actual en las diferentes categorías previamente modelizadas.

Para ello cuando navega crea un hilo por categoría y lanza la clasificación por medio del mismo, obteniendo de manera asíncrona el resultado y mostrando al usuario el mismo a través de unos checks en la pantalla.

La interfaz que se adjunta en el programa sirve de simple comprobación. Una interfaz más evolucionada podría resultar en una propia página web que navegue a la página indicada y además, mediante algún sistema de menús o iframes, mostrar la clasificación de la misma.

El algoritmo básico es el siguiente:

```
url <- from user
Navigate(url)
foreach class in classificators
        fork clsclass.classificate(url)
endforeach
```

La clase clsclass y su método classificate sería uno de los descritos a continuación, bien el específico para Blogs, bien el general para el resto de clases a partir de las características H&L&U

II.5.13. Diseño del ClsBlogs

La clase de manejo de los Blogs es la encargada de obtener los atributos y valores para una página dada en la manera diseñada en puntos anteriores, así como la de obtener un modelo previamente entrenado y determinar, a partir de una dirección, si pertenece o no a la clase.

Para ello se proveen los siguientes métodos:

- LoadClassifier: Carga un modelo previamente entrenado con Weka

ENTRADA: La ruta al modelo
SALIDA: Un clasificador entrenado

- IsBlog: Indica si una determinada página web es blog a partir de un modelo previamente entrenado y cargado. Para ello lo puede realizar a partir de su url o de su url más su html. Si no se facilita el html lo obtiene de Internet.

ENTRADA: Url a la página y /o html de la misma
SALIDA: Booleano indicando si pertenece a la categoría Blogs

- GetInstance: Obtiene una instancia weka a partir de la url y/o el html de la página web que se desee. Para ello obtiene los atributos necesarios para el blog y los genera con el formato concreto de Weka.

ENTRADA: Url y/o html
SALIDA: Una instancia de weka para la representación Blogs con el interfaz weka.core.Instances

- GetBlogAttributes: Obtiene el conjutno de atributos de una página web para la representación concreta de los Blogs.

ENTRADA: Url y/o html
SALIDA: Structura con el conjunto de características de un Blog.

Su algoritmo es el siguiente:

```
StrBlog GetBlogAttributtes(url, html)

        iNWords <- HtmlMngr.GetNWords(html)
        iNWordsUrl <- HtmlMngr.GetNWordsUrl(url)

        strBlog.NBLOGINURL <- HtmlMngr.GetNumOccurs(url, "blog") / iNWordsUrl
        strBlog.NBLOG <- HtmlMngr.GetNumOccurs(html, "blog") / iNWords
        strBlog.NPOST <- HtmlMngr.GetNumOccurs(html, "post") / iNWords
        strBlog.NRSS <- HtmlMngr.GetNumOccurs(html, "rss") +
                                HtmMngr.GetNumOccurs(html, "atom") / iNWords

        iNDates <- HtmlMngr.GetNDates(html)
        iNComments <- HtmlMngr.GetNComments(html)
        iNCommentsInLinks <- HtmlMngr.GetNCommentsInLinks(html)

        strBlog.CommentsVsDates <- iNComments / iNDates
        strBlog.LinkCommentsVsDates <- iNCommentsInLinks / iNDates
        strBlog.LinkCommentsVsComments <- iNCommentsInLinks / iNComments

        strBlog.HasIsland <- HtmlMngr.GetNLinksInIslands(url, html, out strBlog.NILinksRatio,
                                                                    out strBlog.NOLinksRatio,
                                                                    out strBlog.NIOLinksRatio)

        iH1 <- HtmlMngr.GetHtmlTag(html, "h1")
        iH2 <- HtmlMngr.GetHtmlTag(html, "h2")
        iH3 <- HtmlMngr.GetHtmlTag(html, "h3")

        if (iNDates > iNComments)
                idif <- iNDates – iH1
                idif2 <- iNDates – iH2
                idif3 <- iNDates – iH3

        else
                idif <- iNComments – iH1
                idif2 <- iNComments – iH2
                idif3 <- iNComments –iH3
        endif

        idiv <- iH1
        if idif2 < idif
                idif <- idif2
                idiv <- iH2
        endif
        if idif3 < idif
                idif <- idif3
                idiv <- iH3
        endif

        strBlog.DatesRatio <- iNDates / iDiv
        strBlog.CommentsRatio <- iNComments / iDiv
        strBlog.CommentsInLinksRatio <- iNCommentsInLinks / iDiv
```

```
        strBlog.Class <- class

        return strBlog
End GetBlogAttributtes
```

Como se puede observar se hace uso de las clases de HtmlMngr descritas en puntos anteriores, y se obtienen los ratios tal y como se definió en su apartado correspondiente.

II.5.14. Diseño del ClsClass

El ClsClass se diseña de manera análoga al ClsBlogs pero en lugar de obtener las características definidas para los Blogs se obtienen para el corpus facilitado.

II.6. Conclusiones al diseño

Con esto queda concluido el diseño del sistema procurando la máxima adecuación a las necesidades descritas en el análisis y a las características de sencillez y orientación a pruebas que define la XP.

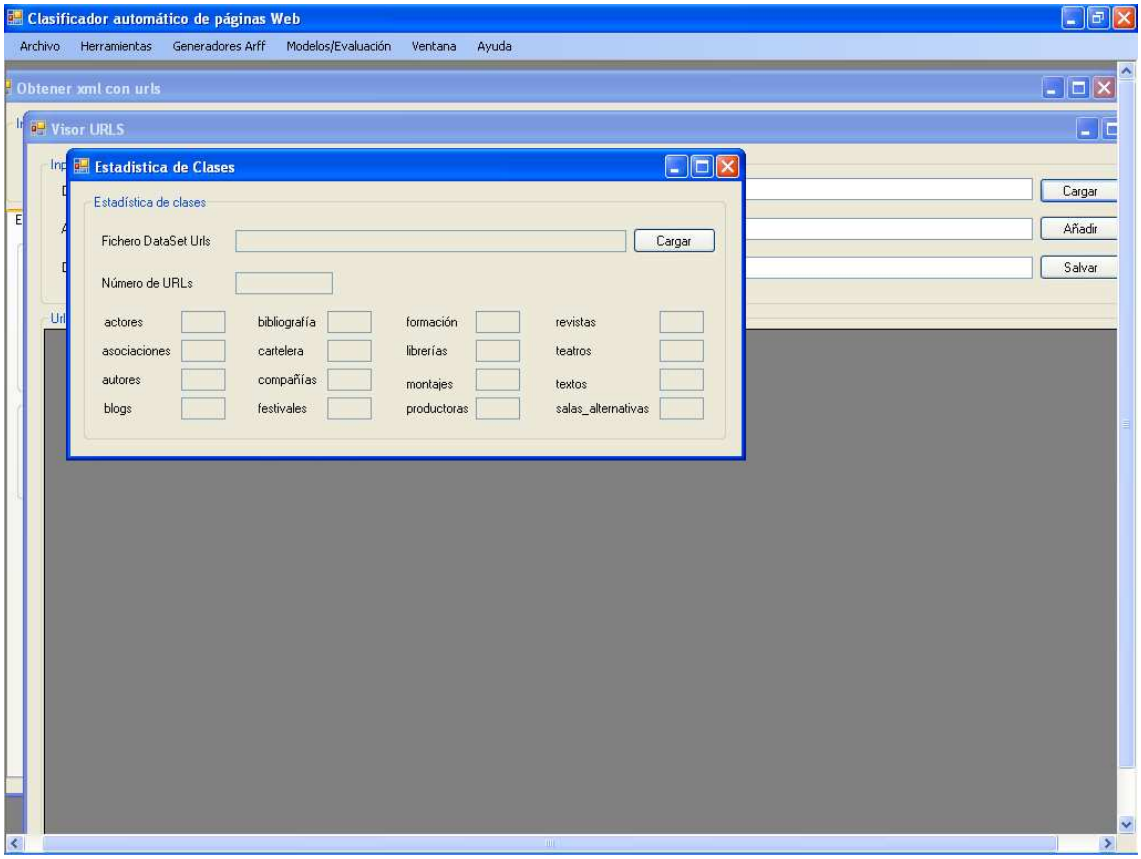
En el apartado siguiente se determinará la evaluación del proyecto, por un lado la evaluación del sistema, es decir, el conjunto de pruebas realizadas, y por otro y más importante, el conjunto de pruebas aplicadas a los modelos para determinar su validez.

ANEXO III: EJEMPLOS DE EJECUCIÓN

Entorno MDI

La aplicación se presenta en un entorno MDI (Multi-Document Interface) donde se pueden abrir diferentes vistas de los documentos para realizar trabajos en paralelo, manejando mediante ventanas apilables los mismos.

Esto dota al usuario de la posibilidad de realizar tareas de supervisión y control mientras se están realizando tareas costosas como la extracción de urls o la creación de las representaciones.

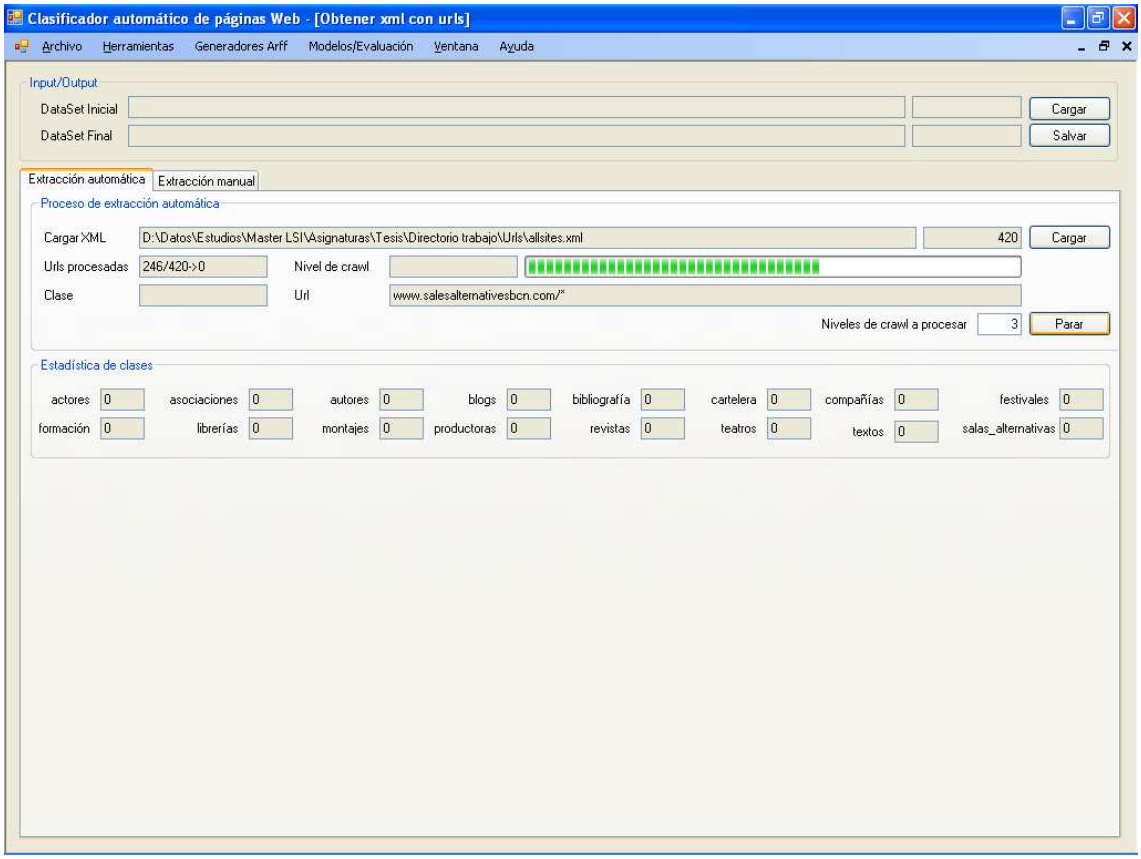


Obtener Html

La interfaz para obtener las páginas Web nos permite, a partir de un fichero de anotaciones, obtener el fichero con el repositorio de Webs hasta un nivel de crawl dado.

Se permite la incorporación de nuevas webs a un repositorio previamente extraído de manera que se pueda realizar la creación de la colección de manera incremental.

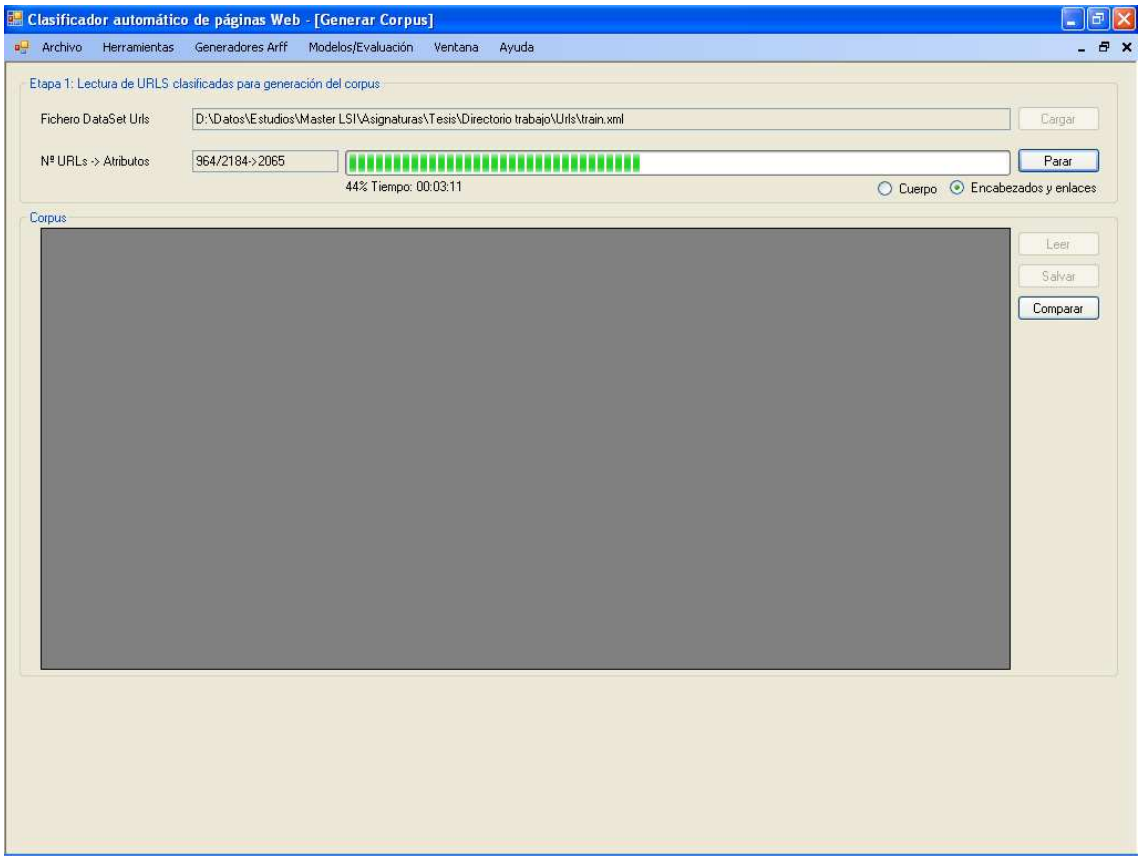
Durante el proceso de extracción se mostrará tanto información de progreso como de estadísticas de distribución en clases de las urls previamente extraídas.



Generar Corpus

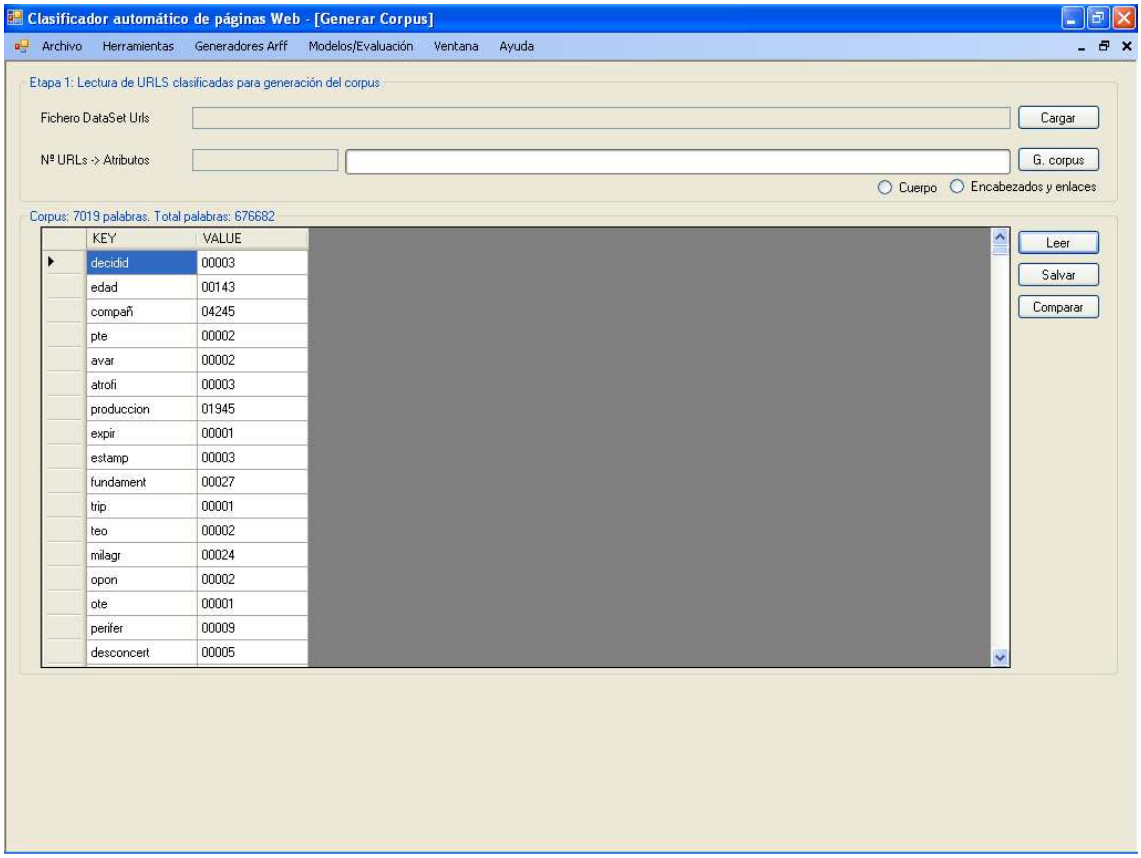
Mediante esta interfaz se nos permite la generación de un corpus a partir de un conjunto de documentos previamente extraídos de internet y almacenados en un repositorio local.

La generación del corpus permite realizarla desde el cuerpo de los documentos o desde los encabezados y los enlaces, de manera que se obtengan los dos tipos de corpus utilizados en la investigación.



Manipular Corpus

La interfaz permite la visualización, comparación y modificación de los corpus mediante un control de rejilla editable.

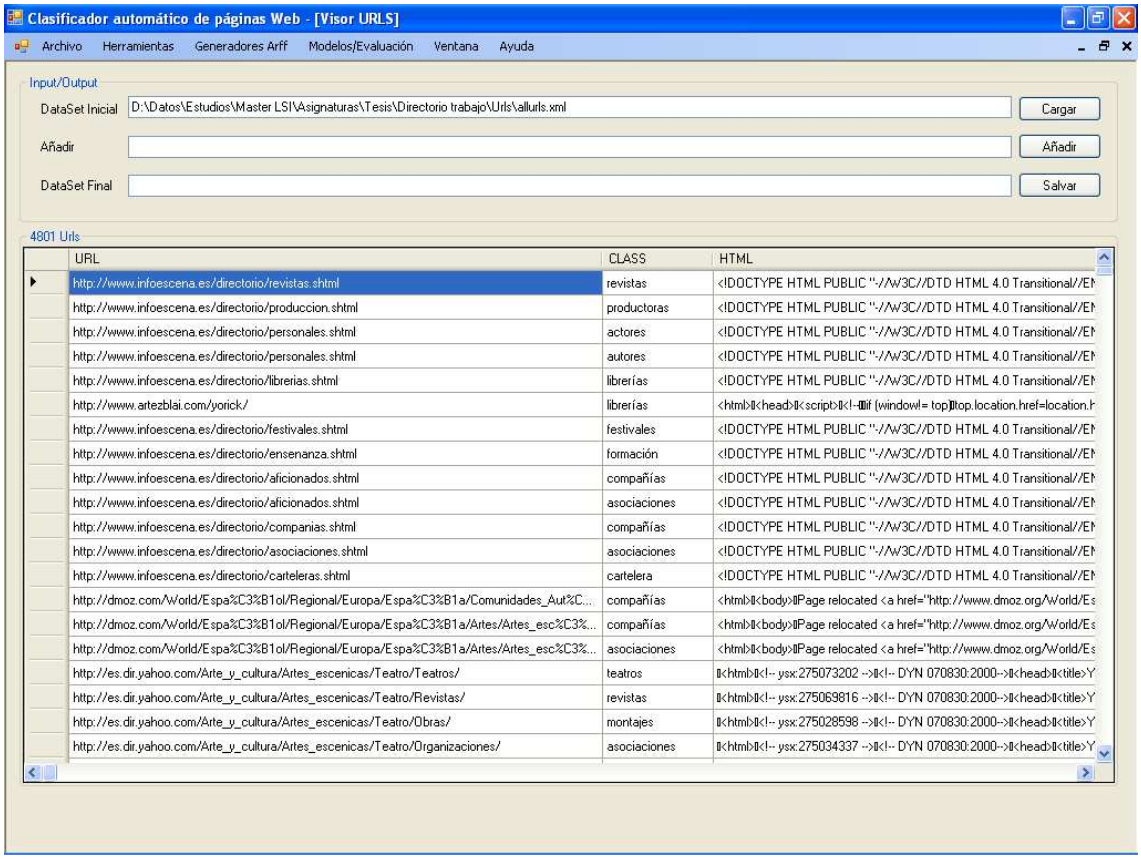


Visualizar Webs

Los repositorios extraídos se pueden combinar, visualizar y modificar a partir de esta interfaz.

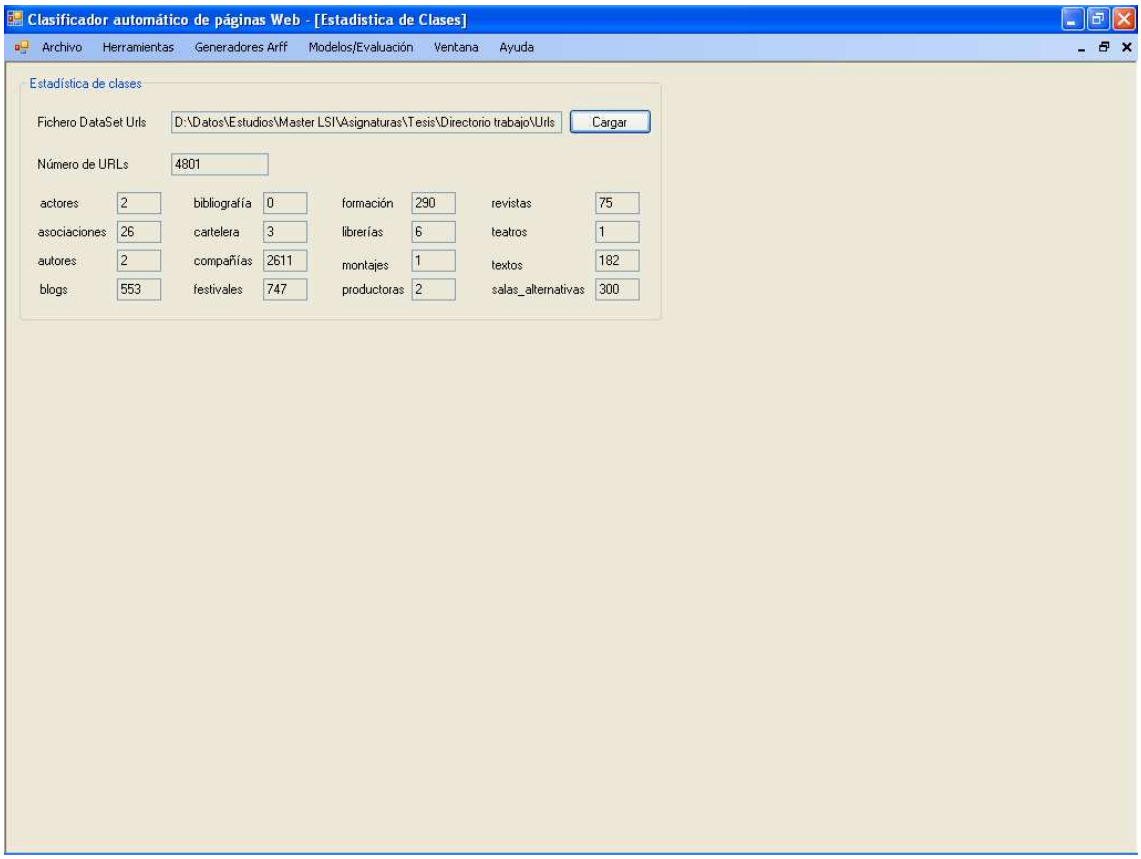
Mediante un control de rejilla se editan las urls y la clase a la que pertenecen, de manera que se puede modificar la misma o eliminar.

Mediante doble click sobre una url se abre un navegador integrado para la visualización de la misma.



Estadística de clases

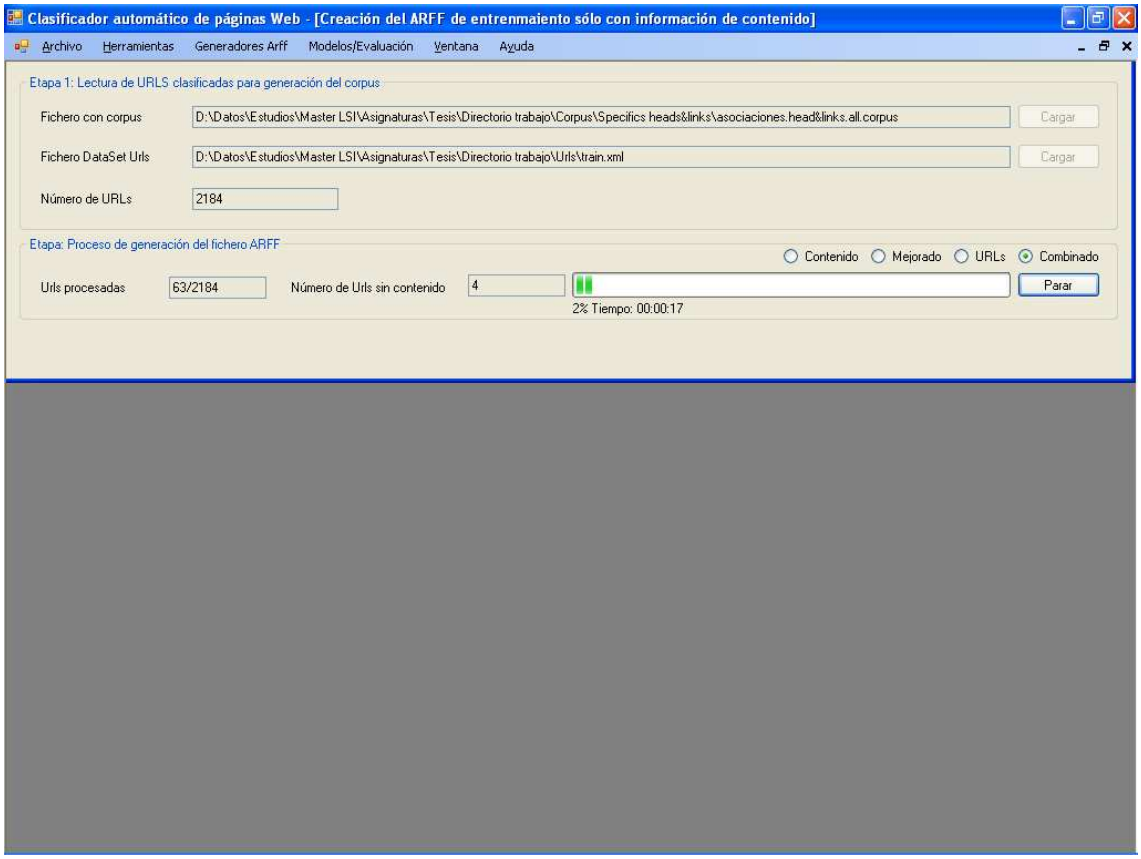
Sencilla interfaz para comprobar el número de urls que componen un repositorio y su distribución por clases.



Generar Arff BoW

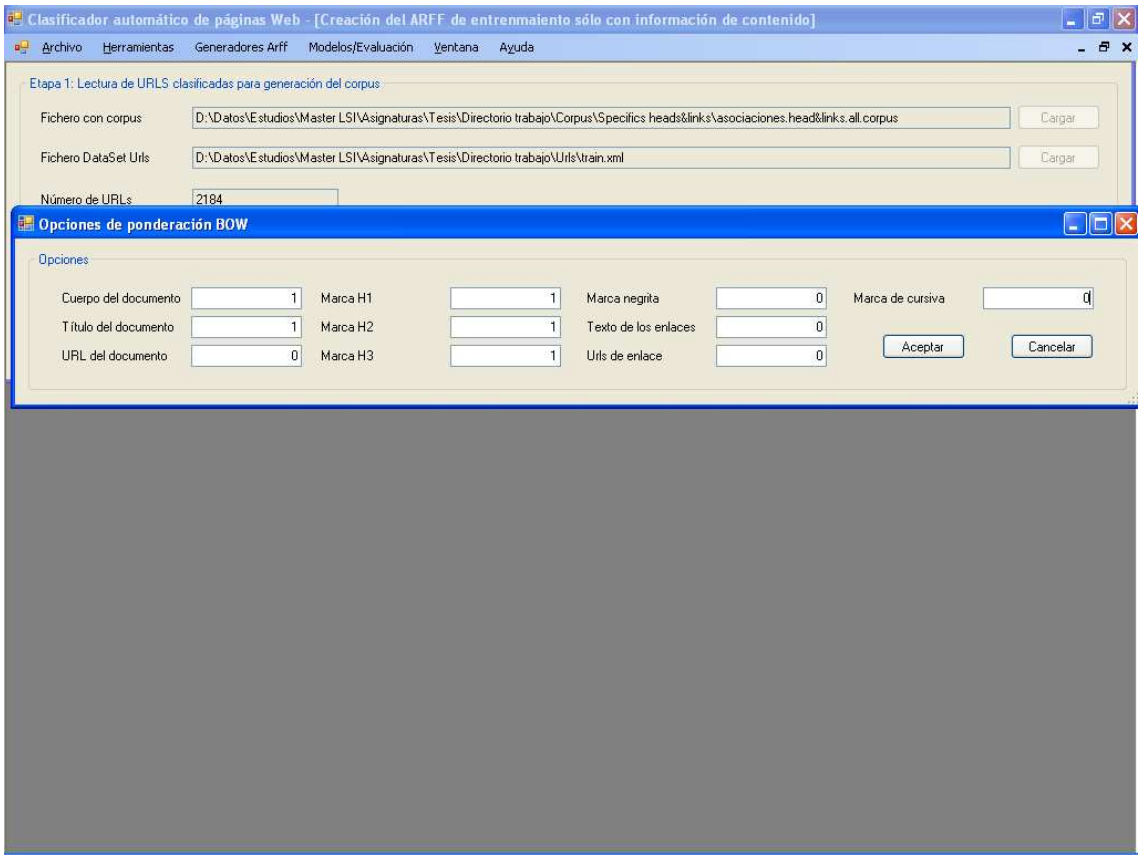
Mediante esta interfaz se generan los ficheros de representación basados en BoW a partir de un repositorio de webs y un corpus determinado.

Existen cuatro maneras de obtener los ficheros, por contenido, por contenido mejorado, que permite la selección de la ponderación para diferentes elementos html, a partir de las urls y a partir del método propuesto de cabecera + enlaces + url.



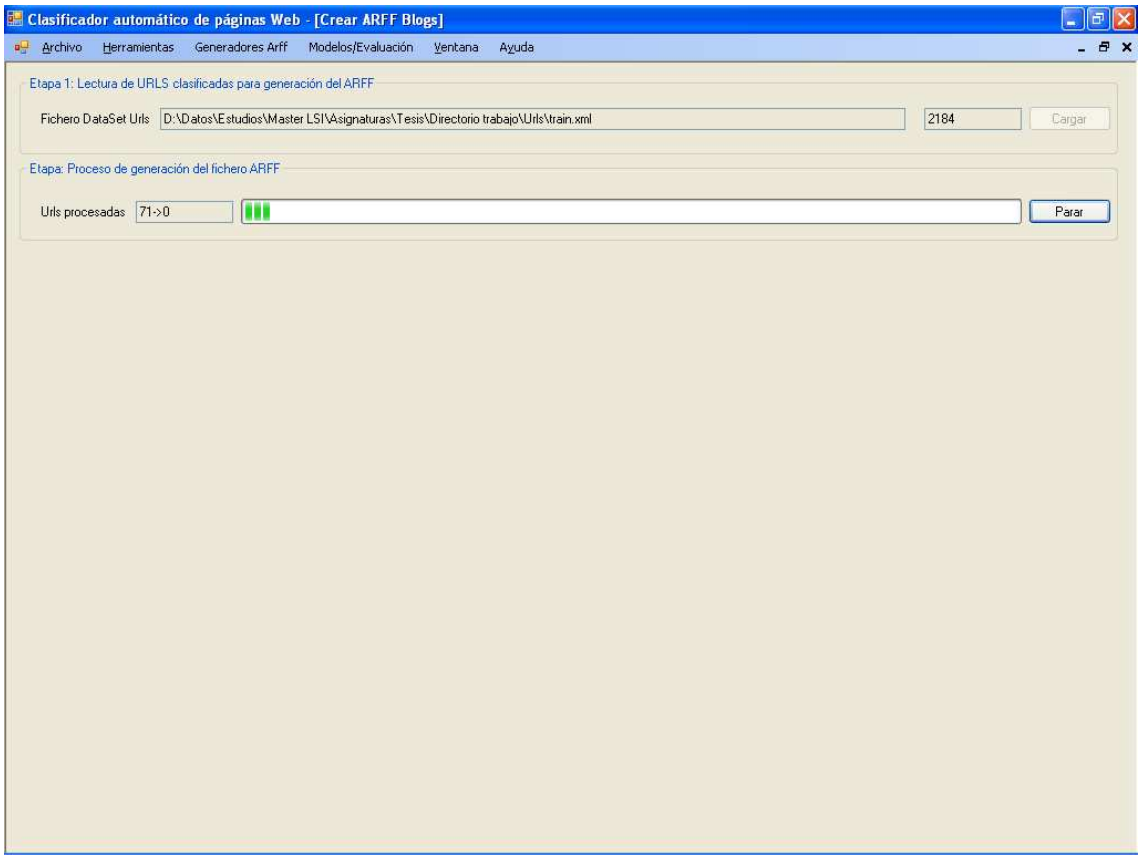
Opciones ponderación

En esta interfaz se permite definir la ponderación para las diferentes etiquetas html en una representación BoW mejorada.



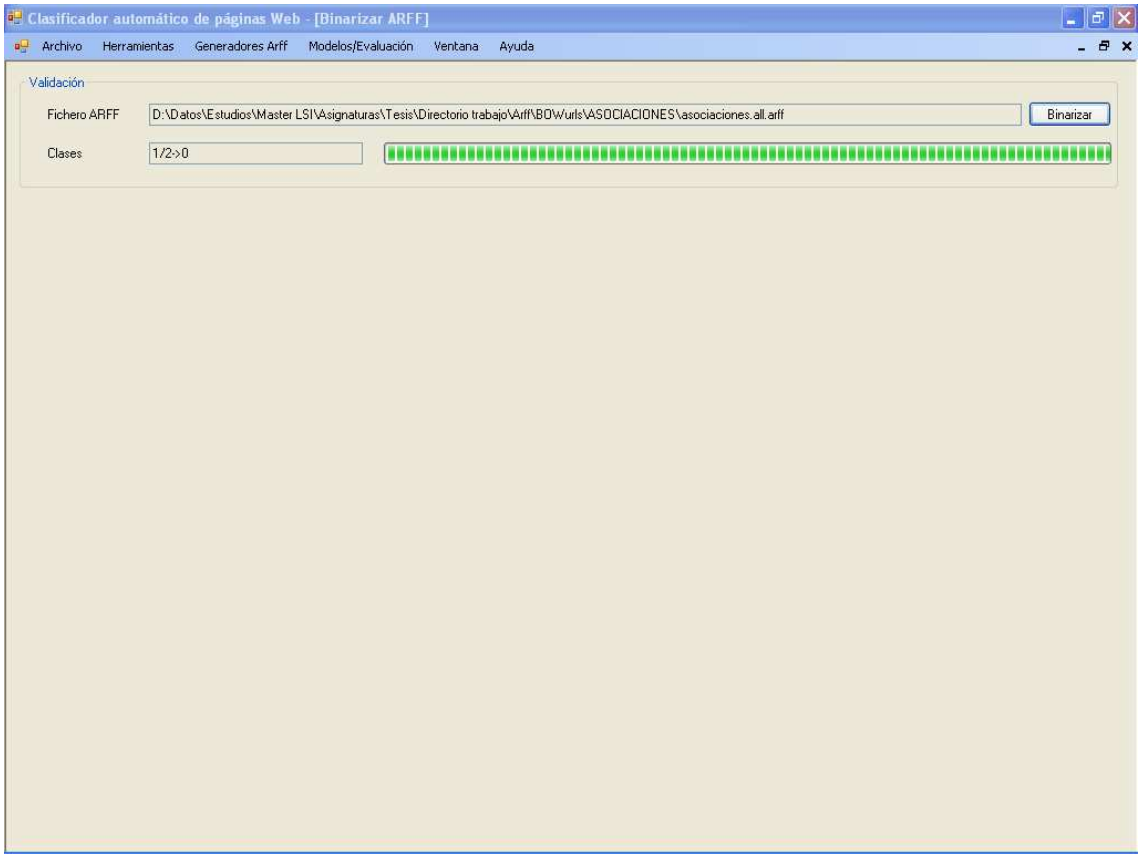
Generar Arff Blogs

Es la interfaz para la creación de los ficheros arff específicos para los Blogs a partir de un repositorio dado.



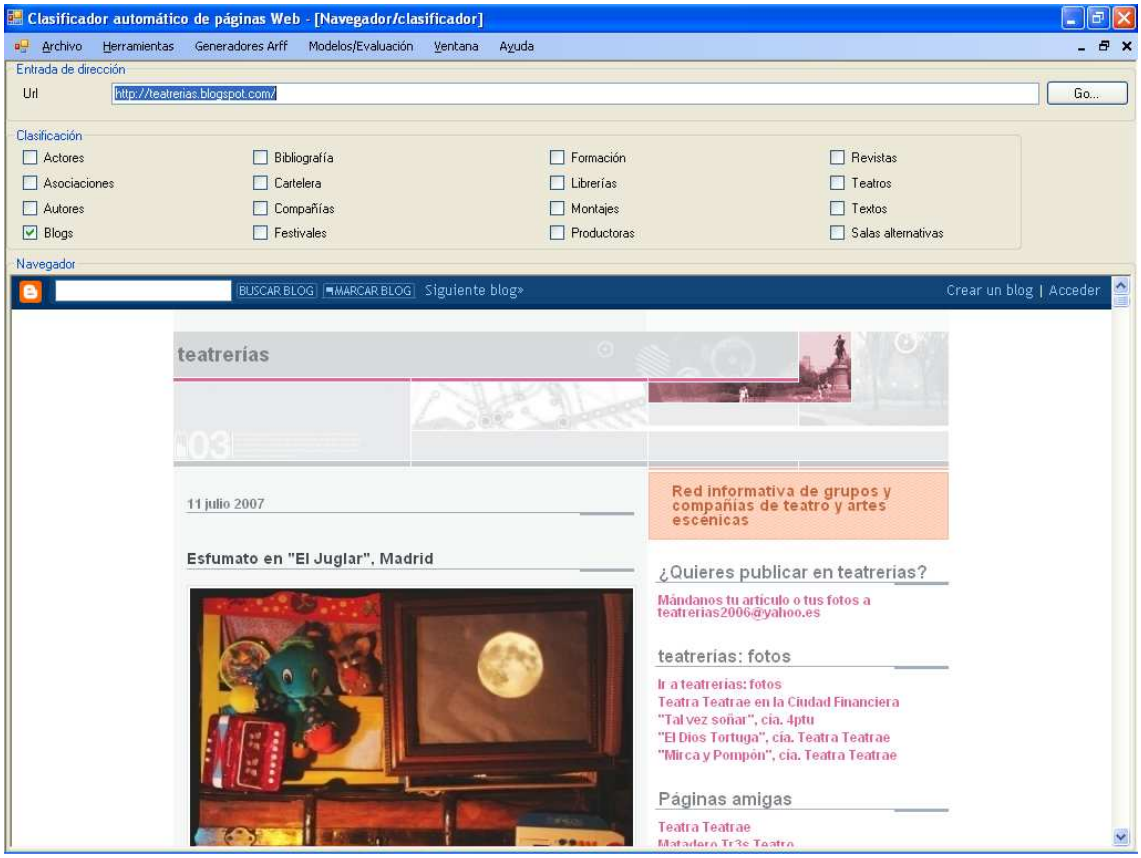
Binarizar

A partir de un fichero arff obtenido para las 16 categorías, obtiene 16 ficheros arff para cada una de las categorías, dónde las páginas marcadas para dicha categoría se clasifican como 1 y el resto como 0.



Navegar y clasificar

Es un navegador integrado que clasifica a la vez que navega las páginas en sus diferentes categorías. No sólo clasifica las introducidas en la url sino también aquellas a las que se navega desde los enlaces.



ANEXO IV: RESULTADOS COMPLETOS DE LAS EVALUACIONES

IV.1 EVALUACIÓN URLs ANOTADAS

MÉTODO: BoW Cross Validation sobre Urls anotadas					
Nº ATRIBUTOS: 2353					
Nº INSTANCIAS: 97					
ACIERTOS: 9 / 9,278%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
Asociaciones	0,000	0,044	0,000	0,000	0,000
Blogs	0,400	0,130	0,143	0,400	0,211
Compañías	0,107	0,188	0,188	0,107	0,136
Festivales	0,222	0,057	0,286	0,222	0,250
Formación	0,000	0,096	0,000	0,000	0,000
Revistas	0,000	0,022	0,000	0,000	0,000
Salas Alternativas	0,000	0,063	0,000	0,000	0,000
Textos	0,0250	0,022	0,333	0,250	0,286

ErrorR(S) = 0,907 +- 0,058

FIGURA IV.1: Evaluación Cross Validation BoW std DSA

IV.2 EVALUACIÓN URLs EXPANDIDAS

MÉTODO: BoW Cross Validation sobre Urls expandidas					
Nº ATRIBUTOS: 7019					
Nº INSTANCIAS: 4663					
ACIERTOS: 3604 / 77,3%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
Asociaciones	0,313	0,011	0,086	0,313	0,135
Blogs	0,753	0,025	0,800	0,753	0,776
Compañías	0,823	0,008	0,992	0,823	0,900
Festivales	0,683	0,029	0,819	0,683	0,745
Formación	0,708	0,013	0,767	0,708	0,736
Revistas	0,548	0,029	0,202	0,548	0,296
Salas Alternativas	0,637	0,019	0,683	0,637	0,659
Textos	0,928	0,002	0,944	0,928	0,936

ErrorR(S) = 0,227 +- 0,012

FIGURA IV.2: Evaluación Cross Validation BoW std DS

IV.3 EVALUACIÓN DS1/DS2, DS2/DS1 Y 2X2

MÉTODO: BoW DS1/DS2 sobre Urls expandidas					
Nº ATRIBUTOS: 7019					
Nº INSTANCIAS: 2091 / 2572					
ACIERTOS: 8 / 3,2%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
Asociaciones	0	0,004	0	0	0
Blogs	0,112	0,131	0,039	0,112	0,058
Compañías	0,019	0,026	0,863	0,019	0,037
Festivales	0	0,575	0	0	0
Formación	0,089	0,076	0,046	0,089	0,061
Revistas	0,333	0,068	0,006	0,333	0,011
Salas Alternativas	0,341	0,028	0,176	0,341	0,233

ErrorR(S) = 0,997 +- 0,002

ASOCIAC.		BLOGS		COMPAÑ.		FESTIVAL.		FORMAC.		REVISTAS		SALAS	
0	10	250	185	20	231	0	740	50	123	3	56	58	182
29	2052	602	1054	54	1786	1	1350	579	1339	14	2018	212	1639

FIGURA IV.3: Evaluación DS1/DS2 BoW std DS

MÉTODO: BoW DS2/DS1 sobre Urls expandidas					
Nº ATRIBUTOS: 7019					
Nº INSTANCIAS: 2572 / 2091					
ACIERTOS: 381 / 18,2%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
Asociaciones	0	0,014	0	0	0
Blogs	0,575	0,364	0,293	0,575	0,389
Compañías	0,080	0,029	0,270	0,08	0,123
Festivales	0	0,001	0	0	0
Formación	0,289	0,305	0,079	0,289	0,124
Revistas	0,051	0,007	0,176	0,051	0,079
Salas Alternativas	0,242	0,115	0,215	0,242	0,227

ErrorR(S) = 0,818 +- 0,017

ASOCIAC.		BLOGS		COMPAÑ.		FESTIVAL.		FORMAC.		REVISTAS		SALAS	
0	6	13	103	44	2256	0	2	9	92	1	2	15	29
9	2557	321	2135	7	265	1478	1092	187	2284	174	2395	70	2458

FIGURA IV.4: Evaluación DS2/DS1 BoW std DS

MÉTODO: BoW 2X2 sobre Urls expandidas					
Nº ATRIBUTOS: 7019					
Nº INSTANCIAS: 4663					
ACIERTOS: 389 / 8,3%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
Asociaciones	0	0,008	0	0	0
Blogs	0,477	0,224	0,222	0,477	0,303
Compañías	0,025	0,029	0,512	0,025	0,048
Festivales	0	0.377	0	0	0
Formación	0,215	0,175	0,072	0,215	0,107
Revistas	0,065	0,041	0,021	0,065	0,031
Salas Alternativas	0,257	0,064	0,206	0,257	0,228

ErrorR(S) = 0,917 +- 0,008

ASOCIAC.		**BLOGS**		**COMPAÑ.**		**FESTIVAL.**		**FORMAC.**		**REVISTAS**		**SALAS**	
0	16	263	288	64	2487	0	742	59	215	4	58	73	211
38	4609	923	3189	61	2051	1479	2442	766	3623	188	4413	282	4097

FIGURA IV.5: Evaluación 2x2 BoW std DS

IV.4 EVALUACIÓN SIN STEM

MÉTODO: BoW Cross Validation sobre corpus sin stem					
Nº ATRIBUTOS: 4169					
Nº INSTANCIAS: 4719					
ACIERTOS: 3953 / 83,8%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
Asociaciones	0,211	0,010	0,078	0,211	0,114
Blogs	0,788	0,016	0,870	0,788	0,827
Compañías	0,892	0,003	0,997	0,892	0,942
Festivales	0,771	0,023	0,862	0,771	0,814
Formación	0,760	0,009	0,839	0,760	0,798
Revistas	0,747	0,033	0,271	0,747	0,397
Salas Alternativas	0,744	0,012	0,806	0,744	0,774
Textos	0,917	0,002	0,943	0,917	0,93

ErrorR(S) = 0,162 +- 0,011

FIGURA IV.6: Evaluación Cross Validation BoW std DS sin stem

IV.5 EVALUACIÓN BoW ESTÁNDAR CON CORPUS COMÚN

IV.5.1 Asociaciones

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 7019					
Nº INSTANCIAS: 2091 / 2572					
ACIERTOS: 2133 / 82,9%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO ASOC.	0,829	0,167	1	0,829	0,906
ASOCIAC.	0,833	0,171	0,011	0,833	0,022
MATRIZ DE CONTINGENCIA					
2128	438				
1	5				

ErrorR($S \notin clase$ **) = 0,171 +- 0,015**

ErrorR($S \in clase$ **) = 0,167 +- 0,298**

ErrorR(S) = 0,171 +- 0,015

FIGURA IV.7: Evaluación DS1/DS2 Asociaciones BoW std Corpus común

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 7019					
Nº INSTANCIAS: 2572 / 2091					
ACIERTOS: 1762 / 84,3%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO ASO	0,847	1	0,994	0,847	0,915
ASOCIA.	0	0,153	0	0	0
MATRIZ DE CONTINGENCIA					
1762	319				
10	0				

ErrorR($S \notin clase$ **) = 0,153 +- 0,015**

ErrorR($S \in clase$ **) = 1 +- 0**

ErrorR(S) = 0,157 +- 0,016

FIGURA IV.8: Evaluación DS2/DS1 Asociaciones BoW std Corpus común

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 7019					
Nº INSTANCIAS: 4663					
ACIERTOS: 3895 / 83,5%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO ASOC.	0,837	0,163	0,997	0,837	0,910
ASOCIA	0,313	0,688	0,007	0,313	0,013
MATRIZ DE CONTINGENCIA					
3890	757				
11	5				

ErrorR($S \notin clase$ **) = 0,163 +- 0,011**

ErrorR($S \in clase$ **) = 0,688 +- 0,227**

ErrorR(S) = 0,165 +- 0,011

FIGURA IV.9: Evaluación 2x2 Asociaciones BoW std Corpus común

IV.5.2 Blogs

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 7019					
Nº INSTANCIAS: 2091 / 2572					
ACIERTOS: 1745 / 67,8%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO BLOG	0,703	0,845	0,946	0,703	0,807
BLOG	0,155	0,297	0,024	0,155	0,042
MATRIZ DE CONTINGENCIA					
1727	729				
98	18				

ErrorR($S \notin clase$ **) = 0,297 +- 0,018**

ErrorR($S \in clase$ **) = 0,845 +- 0,066**

ErrorR(S) = 0,322 +- 0,018

FIGURA IV.10: Evaluación DS1/DS2 Blogs BoW std Corpus común

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 7019					
Nº INSTANCIAS: 2572 / 2091					
ACIERTOS: 988 / 47,2%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO BLOG	0,357	0,090	0,938	0,357	0,518
BLOG	0,910	0,643	0,271	0,910	0,418
MATRIZ DE CONTINGENCIA					
592	1064				
39	396				

ErrorR($S \notin clase$ **) = 0,643 +- 0,023**

ErrorR($S \in clase$ **) = 0,090 +- 0,027**

ErrorR(S) = 0,527 +- 0,021

FIGURA IV.11: Evaluación DS2/DS1 Blogs BoW std Corpus común

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 7019					
Nº INSTANCIAS: 4663					
ACIERTOS: 2733 / 58,6%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO BLOG	0,564	0,436	0,944	0,564	0,706
BLOG	0,751	0,249	0,188	0,751	0,300
MATRIZ DE CONTINGENCIA					
2319	1793				
137	414				

ErrorR($S \notin clase$ **) = 0,436 +- 0,015**

ErrorR($S \in clase$ **) = 0,249 +- 0,036**

ErrorR(S) = 0,414 +- 0,014

FIGURA IV.12: Evaluación 2x2 Blogs BoW std Corpus común

IV.5.3 Compañías

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 7019					
Nº INSTANCIAS: 2091 / 2572					
ACIERTOS: 1628 / 63,3%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO COMP.	0,754	0,381	0,189	0,754	0,303
COMPAÑÍA	0,619	0,246	0,955	0,619	0,751
MATRIZ DE CONTINGENCIA					
205	67				
877	1423				

ErrorR($S \notin clase$ **) = 0,246 +- 0,051**

ErrorR($S \in clase$ **) = 0,381 +- 0,020**

ErrorR(S) = 0,367 +- 0,019

FIGURA IV.13: Evaluación DS1/DS2 Compañías BoW std Corpus común

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 7019					
Nº INSTANCIAS: 2572 / 2091					
ACIERTOS: 1718 / 82,16%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO COMP	0,840	0,315	0,951	0,840	0,892
COMPAÑÍA	0,685	0,160	0,369	0,685	0,480
MATRIZ DE CONTINGENCIA					
1546	294				
79	172				

ErrorR($S \notin clase$ **) = 0,160 +- 0,017**

ErrorR($S \in clase$ **) = 0,315 +- 0,057**

ErrorR(S) = 0,178 +- 0,016

FIGURA IV.14: Evaluación DS2/DS1 Compañías BoW std Corpus común

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 7019					
Nº INSTANCIAS: 4663					
ACIERTOS: 3346 / 71,8%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO COMP	0,829	0,171	0,647	0,829	0,727
COMPAÑÍA	0,625	0,375	0,815	0,625	0,708
MATRIZ DE CONTINGENCIA					
1751	361				
956	1595				

ErrorR($S \notin clase$ **) = 0,171 +- 0,016**

ErrorR($S \in clase$ **) = 0,375 +- 0,019**

ErrorR(S) = 0,282 +- 0,013

FIGURA IV.15: Evaluación 2x2 Compañías BoW std Corpus común

IV.5.4 Festivales

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 7019					
Nº INSTANCIAS: 2091 / 2572					
ACIERTOS: 1642 / 63,8%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FEST	0,638	0	1	0,638	0,779
FESTIVAL	1	0,362	0,002	1	0,004
MATRIZ DE CONTINGENCIA					
1640	930				
0	2				

ErrorR($S \notin clase$ **) = 0,362 +- 0,019**

ErrorR($S \in clase$ **) = 0 +- 0**

ErrorR(S) = 0,362 +- 0,019

FIGURA IV.16: Evaluación DS1/DS2 Festivales BoW std Corpus común

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 7019					
Nº INSTANCIAS: 2572 / 2091					
ACIERTOS: 1249 / 59,7%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FEST	0,866	0,893	0,639	0,866	0,735
FESTIVAL	0,107	0,134	0,304	0,107	0,158
MATRIZ DE CONTINGENCIA					
1170	181				
661	79				

ErrorR($S \notin clase$ **) = 0,134 +- 0,018**

ErrorR($S \in clase$ **) = 0,893 +- 0,022**

ErrorR(S) = 0,403 +- 0,021

FIGURA IV.17: Evaluación DS2/DS1 Festivales BoW std Corpus común

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 7019					
Nº INSTANCIAS: 4663					
ACIERTOS: 2891 / 62,0%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FEST	0,717	0,283	0,810	0,717	0,760
FESTIVAL	0,109	0,891	0,068	0,109	0,084
MATRIZ DE CONTINGENCIA					
2810	1111				
661	81				

ErrorR($S \notin clase$ **) = 0,283 +- 0,014**

ErrorR($S \in clase$ **) = 0,891 +- 0,022**

ErrorR(S) = 0,380 +- 0,014

FIGURA IV.18: Evaluación 2x2 Festivales BoW std Corpus común

IV.5.5 Formación

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 7019					
Nº INSTANCIAS: 2091 / 2572					
ACIERTOS: 2062 / 80,2%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FORM	0,821	0,663	0,968	0,821	0,888
FORMAC.	0,337	0,179	0,071	0,337	0,118
MATRIZ DE CONTINGENCIA					
2028	443				
67	34				

ErrorR($S \notin clase$ **) = 0,179 +- 0,015**

ErrorR($S \in clase$ **) = 0,663 +- 0,092**

ErrorR(S) = 0,198 +- 0,015

FIGURA IV.19: Evaluación DS1/DS2 Formación BoW std Corpus común

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 7019					
Nº INSTANCIAS: 2572 / 2091					
ACIERTOS: 835 / 39,9%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FORM	0,369	0,260	0,940	0,369	0,530
FORMACIÓN	0,740	0,631	0,096	0,740	0,169
MATRIZ DE CONTINGENCIA					
707	1211				
45	128				

ErrorR($S \notin clase$ **) = 0,631 +- 0,022**

ErrorR($S \in clase$ **) = 0,260 +- 0,065**

ErrorR(S) = 0,601 +- 0,021

FIGURA IV.20: Evaluación DS2/DS1 Formación BoW std Corpus común

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 7019					
Nº INSTANCIAS: 4663					
ACIERTOS: 2897 / 62,1%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FORM.	0,623	0,377	0,961	0,623	0,756
FORMACIÓN	0,591	0,409	0,089	0,591	0,155
MATRIZ DE CONTINGENCIA					
2735	1654				
112	162				

ErrorR($S \notin clase$ **) = 0,377 +- 0,014**

ErrorR($S \in clase$ **) = 0,409 +- 0,058**

ErrorR(S) = 0,379 +- 0,014

FIGURA IV.21: Evaluación 2x2 Formación BoW std Corpus común

IV.5.6 Revistas

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 7019					
Nº INSTANCIAS: 2091 / 2572					
ACIERTOS: 2452 / 95,3%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO REV	0,954	0,333	1	0,954	0,976
REVISTA	0,667	0,046	0,017	0,667	0,032
MATRIZ DE CONTINGENCIA					
2450	119				
1	2				

ErrorR($S \notin clase$) = 0,046 +- 0,008
ErrorR($S \in clase$) = 0,333 +- 0,533
ErrorR(S) = 0,047 +- 0,008
FIGURA IV.22: Evaluación DS1/DS2 Revistas BoW std Corpus común

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 7019					
Nº INSTANCIAS: 2572 / 2091					
ACIERTOS: 1931 / 92,3%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO REV.	0,948	0,915	0,973	0,948	0,960
REVISTA	0,085	0,052	0,045	0,085	0,059
MATRIZ DE CONTINGENCIA					
1926	106				
54	5				

ErrorR($S \notin clase$) = 0,052 +- 0,010
ErrorR($S \in clase$) = 0,915 +- 0,071
ErrorR(S) = 0,077 +- 0,011
FIGURA IV.23: Evaluación DS2/DS1 Revistas BoW std Corpus común

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 7019					
Nº INSTANCIAS: 4663					
ACIERTOS: 4383 / 94%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO REV.	0,951	0,049	0,988	0,951	0,969
REVISTA	0,113	0,887	0,030	0,113	0,048
MATRIZ DE CONTINGENCIA					
4376	225				
55	7				

ErrorR($S \notin clase$) = 0,049 +- 0,006
ErrorR($S \in clase$) = 0,887 +- 0,079
ErrorR(S) = 0,060 +- 0,007
FIGURA IV.24: Evaluación 2x2 Revistas BoW std Corpus común

IV.5.7 Salas Alternativas

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 7019					
Nº INSTANCIAS: 2091 / 2572					
ACIERTOS: 1749 / 68,0%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO SALA	0,678	0,182	0,995	0,678	0,806
SALA	0,818	0,322	0,042	0,818	0,080
MATRIZ DE CONTINGENCIA					
1713	815				
8	36				

ErrorR($S \notin clase$ **) = 0,322 +- 0,018**

ErrorR($S \in clase$ **) = 0,182 +- 0,114**

ErrorR(S) = 0,320 +- 0,018

FIGURA IV.25: Evaluación DS1/DS2 Salas Alternativas BoW std Corpus común

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 7019					
Nº INSTANCIAS: 2572 / 2091					
ACIERTOS: 1382 / 66,1%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO SALA	0,672	0,425	0,924	0,672	0,778
SALA	0,575	0,328	0,185	0,575	0,28
MATRIZ DE CONTINGENCIA					
1244	607				
102	138				

ErrorR($S \notin clase$ **) = 0,328 +- 0,021**

ErrorR($S \in clase$ **) = 0,425 +- 0,063**

ErrorR(S) = 0,339 +- 0,020

FIGURA IV.26: Evaluación DS2/DS1 Salas Alternativas BoW std Corpus común

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 7019					
Nº INSTANCIAS: 4663					
ACIERTOS: 3131 / 67,1%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO SALA	0,675	0,325	0,964	0,675	0,794
SALA	0,613	0,387	0,109	0,613	0,185
MATRIZ DE CONTINGENCIA					
2957	1422				
110	174				

ErrorR($S \notin clase$ **) = 0,325 +- 0,014**

ErrorR($S \in clase$ **) = 0,387 +- 0,057**

ErrorR(S) = 0,329 +- 0,013

FIGURA IV.27: Evaluación 2x2 Salas Alternativas BoW std Corpus común

IV.5.8 Textos

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 7019					
Nº INSTANCIAS: 4663					
ACIERTOS: 4619 / 99,1%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO TEXT	0,991	0,022	0,999	0,991	0,995
TEXT	0,978	0,009	0,816	0,978	0,889
MATRIZ DE CONTINGENCIA					

4442	40
4	177

ErrorR($S \notin clase$ **) = 0,009 +- 0,003**

ErrorR($S \in clase$ **) = 0,022 +- 0,021**

ErrorR(S) = 0,009 +- 0,003

FIGURA IV.28: Evaluación Cross Validation Textos BoW std Corpus común

IV.6 EVALUACIÓN BoW ESTÁNDAR

IV.6.1 Asociaciones

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 522					
Nº INSTANCIAS (train/test): 2091 / 2572					
ACIERTOS: 2530 / 98,37%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO ASOC.	0,986	1	0,998	0,986	0,992
ASOCIAC.	0,014	0	0	0	0
MATRIZ DE CONTINGENCIA					
2530	36				
6	0				

ErrorR($S \notin clase$ **) = 0,014 +- 0,005**
ErrorR($S \in clase$ **) = 1 +- 0**
ErrorR(S) = 0,016 +- 0,005
FIGURA IV.29: Evaluación DS1/DS2 Asociaciones BoW std

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 522					
Nº INSTANCIAS (train/test): 2572 / 2091					
ACIERTOS: 1756 / 83,979%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO ASO	0,844	1	0,994	0,844	0,913
ASOCIA.	0	0,156	0	0	0
MATRIZ DE CONTINGENCIA					
1756	325				
10	0				

ErrorR($S \notin clase$ **) = 0,156 +- 0,016**
ErrorR($S \in clase$ **) = 1 +- 0**
ErrorR(S) = 0,160 +- 0,016
FIGURA IV.30: Evaluación DS2/DS1 Asociaciones BoW std

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 522					
Nº INSTANCIAS: 4663					
ACIERTOS: 4286 / 91,9%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO ASOC.	0,922	0,078	0,996	0,922	0,958
ASOCIA	0,000	1,000	0,000	0,000	0,000
MATRIZ DE CONTINGENCIA					
4286	361				
16	0				

ErrorR($S \notin clase$ **) = 0,078 +- 0,008**
ErrorR($S \in clase$ **) = 1 +- 0**
ErrorR(S) = 0,081 +- 0,008
FIGURA IV.31: Evaluación 2x2 Asociaciones BoW std

IV.6.2 Blogs

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 4516					
Nº INSTANCIAS (train/test): 2091 / 2572					
ACIERTOS: 1748 / 68,0%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO BLOG	0,705	0,862	0,945	0,705	0,808
BLOG	0,138	0,295	0,022	0,138	0,037
MATRIZ DE CONTINGENCIA					
1732	724				
100	16				

ErrorR($S \notin clase$ **) = 0,295 +- 0,018**
ErrorR($S \in clase$ **) = 0,862 +- 0,063**
ErrorR(S) = 0,320 +- 0,018
FIGURA IV.32: Evaluación DS1/DS2 Blogs BoW std

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 4516					
Nº INSTANCIAS (train/test): 2572 / 2091					
ACIERTOS: 988 / 47,25%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO BLOG	0,359	0,094	0,935	0,359	0,519
BLOG	0,906	0,641	0,271	0,906	0,417
MATRIZ DE CONTINGENCIA					
594	1062				
41	394				

ErrorR($S \notin clase$ **) = 0,641 +- 0,023**
ErrorR($S \in clase$ **) = 0,094 +- 0,027**
ErrorR(S) = 0,527 +- 0,021
FIGURA IV.33: Evaluación DS2/DS1 Blogs BoW std

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 4516					
Nº INSTANCIAS: 4663					
ACIERTOS: 2736 / 58,7%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO BLOG	0,566	0,434	0,943	0,566	0,707
BLOG	0,744	0,256	0,187	0,744	0,299
MATRIZ DE CONTINGENCIA					
2326	1786				
141	410				

ErrorR($S \notin clase$ **) = 0,434 +- 0,015**
ErrorR($S \in clase$ **) = 0,256 +- 0,036**
ErrorR(S) = 0,413 +- 0,014
FIGURA IV.34: Evaluación 2x2 Blogs BoW std

IV.6.3 Compañías

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 3091					
Nº INSTANCIAS (train/test): 2091 / 2572					
ACIERTOS: 1458 / 56,59%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO COMP.	0,757	0,456	0,164	0,757	0,27
COMPAÑÍA	0,544	0,243	0,95	0,544	0,692
MATRIZ DE CONTINGENCIA					
206	66				
1048	1252				

ErrorR($S \notin clase$ **) = 0,243 +- 0,051**
ErrorR($S \in clase$ **) = 0,456 +- 0,020**
ErrorR(S) = 0,433 +- 0,019
FIGURA IV.35: Evaluación DS1/DS2 Compañías BoW std

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 3091					
Nº INSTANCIAS (train/test): 2572 / 2091					
ACIERTOS: 1735 / 82,97%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO COMP	0,849	0,315	0,952	0,849	0,898
COMPAÑÍA	0,685	0,151	0,383	0,685	0,491
MATRIZ DE CONTINGENCIA					
1563	277				
79	172				

ErrorR($S \notin clase$ **) = 0,151 +- 0,016**
ErrorR($S \in clase$ **) = 0,315 +- 0,057**
ErrorR(S) = 0,170 +- 0,016
FIGURA IV.36: Evaluación DS2/DS1 Compañías BoW std

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 3091					
Nº INSTANCIAS: 4663					
ACIERTOS: 3193 / 68,5%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO COMP	0,838	0,162	0,611	0,838	0,706
COMPAÑÍA	0,558	0,442	0,806	0,558	0,660
MATRIZ DE CONTINGENCIA					
1769	343				
1127	1424				

ErrorR($S \notin clase$ **) = 0,162 +- 0,016**
ErrorR($S \in clase$ **) = 0,442 +- 0,019**
ErrorR(S) = 0,315 +- 0,013
FIGURA IV.37: Evaluación 2x2 Compañías BoW std

IV.6.4 Festivales

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 5375					
Nº INSTANCIAS (train/test): 2091 / 2572					
ACIERTOS: 1641 / 63,80%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FEST	0,638	0	1	0,638	0,779
FESTIVAL	1	0,362	0,002	1	0,004
MATRIZ DE CONTINGENCIA					
1639	931				
0	2				

ErrorR($S \notin clase$ **) = 0,362 +- 0,019**

ErrorR($S \in clase$ **) = 1 +- 0**

ErrorR(S) = 0,362 +- 0,019

FIGURA IV.38: Evaluación DS1/DS2 Festivales BoW std

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 5375					
Nº INSTANCIAS (train/test): 2572 / 2091					
ACIERTOS: 1249 / 59,73%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FEST	0,866	0,893	0,639	0,866	0,735
FESTIVAL	0,107	0,134	0,304	0,107	0,158
MATRIZ DE CONTINGENCIA					
1170	181				
661	79				

ErrorR($S \notin clase$ **) = 0,134 +- 0,018**

ErrorR($S \in clase$ **) = 0,893 +- 0,022**

ErrorR(S) = 0,403 +- 0,021

FIGURA IV.39: Evaluación DS2/DS1 Festivales BoW std

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 5375					
Nº INSTANCIAS: 4663					
ACIERTOS: 2890 / 62,0%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FEST	0,716	0,284	0,810	0,716	0,760
FESTIVAL	0,109	0,891	0,068	0,109	0,084
MATRIZ DE CONTINGENCIA					
2809	1112				
661	81				

ErrorR($S \notin clase$ **) = 0,284 +- 0,014**

ErrorR($S \in clase$ **) = 0,891 +- 0,022**

ErrorR(S) = 0,380 +- 0,014

FIGURA IV.40: Evaluación 2x2 Festivales BoW std

IV.6.5 Formación

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 3274					
Nº INSTANCIAS (train/test): 2091 / 2572					
ACIERTOS: 2113 / 85,15%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FORM	0,841	0,663	0,969	0,841	0,901
FORMAC.	0,337	0,159	0,08	0,337	0,129
MATRIZ DE CONTINGENCIA					
2079	392				
67	34				

ErrorR($S \notin clase$) = 0,156 +- 0,014
ErrorR($S \in clase$) = 0,663 +- 0,092
ErrorR(S) = 0,178 +- 0,015
FIGURA IV.41: Evaluación DS1/DS2 Formación BoW std

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 3274					
Nº INSTANCIAS (train/test): 2572 / 2091					
ACIERTOS: 816 / 39,02%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FORM	0,359	0,26	0,939	0,359	0,519
FORMACIÓN	0,74	0,641	0,094	0,74	0,167
MATRIZ DE CONTINGENCIA					
688	1230				
45	128				

ErrorR($S \notin clase$) = 0,641 +- 0,021
ErrorR($S \in clase$) = 0,260 +- 0,065
ErrorR(S) = 0,610 +- 0,021
FIGURA IV.42: Evaluación DS2/DS1 Formación BoW std

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 3274					
Nº INSTANCIAS: 4663					
ACIERTOS: 2929 / 62,8%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FORM.	0,630	0,370	0,961	0,630	0,761
FORMACIÓN	0,591	0,409	0,091	0,591	0,157
MATRIZ DE CONTINGENCIA					
2767	1622				
112	162				

ErrorR($S \notin clase$) = 0,370 +- 0,014
ErrorR($S \in clase$) = 0,409 +- 0,058
ErrorR(S) = 0,372 +- 0,014
FIGURA IV.43: Evaluación 2x2 Formación BoW std

IV.6.6 Revistas

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 2557					
Nº INSTANCIAS (train/test): 2091 / 2572					
ACIERTOS: 2446 / 95,10%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO REV	0,951	0,333	1	0,951	0,975
REVISTA	0,667	0,049	0,016	0,667	0,031
MATRIZ DE CONTINGENCIA					
2444	125				
1	2				

ErrorR($S \notin clase$ **) = 0,049 +- 0,008**
ErrorR($S \in clase$ **) = 0,333 +- 0,533**
ErrorR(S) = 0,049 +- 0,008
FIGURA IV.44: Evaluación DS1/DS2 Revistas BoW std

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 2557					
Nº INSTANCIAS (train/test): 2572 / 2091					
ACIERTOS: 1936 / 92,59%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO REV.	0,95	0,915	0,973	0,95	0,961
REVISTA	0,085	0,05	0,047	0,085	0,061
MATRIZ DE CONTINGENCIA					
1931	191				
54	5				

ErrorR($S \notin clase$ **) = 0,090 +- 0,012**
ErrorR($S \in clase$ **) = 0,915 +- 0,071**
ErrorR(S) = 0,074 +- 0,011
FIGURA IV.45: Evaluación DS2/DS1 Revistas BoW std

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 2557					
Nº INSTANCIAS: 4753					
ACIERTOS: 4382 / 92,2%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO REV.	0,933	0,067	0,988	0,933	0,959
REVISTA	0,113	0,887	0,022	0,113	0,036
MATRIZ DE CONTINGENCIA					
4375	316				
55	7				

ErrorR($S \notin clase$ **) = 0,067 +- 0,007**
ErrorR($S \in clase$ **) = 0,887 +- 0,079**
ErrorR(S) = 0,078 +- 0,008
FIGURA IV.46: Evaluación 2x2 Revistas BoW std

IV.6.7 Salas Alternativas

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 2989					
Nº INSTANCIAS (train/test): 2091 / 2572					
ACIERTOS: 1745 / 67,85%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO SALA	0,676	0,182	0,995	0,676	0,805
SALA	0,818	0,324	0,042	0,818	0,08
MATRIZ DE CONTINGENCIA					
1709	819				
8	36				

ErrorR($S \notin clase$ **) = 0,324 +- 0,018**

ErrorR($S \in clase$ **) = 0,182 +- 0,114**

ErrorR(S) = 0,322 +- 0,018

FIGURA IV.47: Evaluación DS1/DS2 Salas Alternativas BoW std

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 2989					
Nº INSTANCIAS (train/test): 2572 / 2091					
ACIERTOS: 1391 / 66,52%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO SALA	0,677	0,429	0,924	0,677	0,728
SALA	0,571	0,323	0,187	0,571	0,281
MATRIZ DE CONTINGENCIA					
1254	597				
103	137				

ErrorR($S \notin clase$ **) = 0,323 +- 0,021**

ErrorR($S \in clase$ **) = 0,429 +- 0,063**

ErrorR(S) = 0,335 +- 0,020

FIGURA IV.48: Evaluación DS2/DS1 Salas Alternativas BoW std

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 2989					
Nº INSTANCIAS: 4663					
ACIERTOS: 3136 / 67,3%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO SALA	0,677	0,323	0,964	0,677	0,795
SALA	0,609	0,391	0,109	0,609	0,185
MATRIZ DE CONTINGENCIA					
2963	1416				
111	173				

ErrorR($S \notin clase$ **) = 0,323 +- 0,014**

ErrorR($S \in clase$ **) = 0,391 +- 0,057**

ErrorR(S) = 0,327 +- 0,013

FIGURA IV.49: Evaluación 2x2 Salas Alternativas BoW std

IV.6.8 Textos

MÉTODO: BoW estándar					
Nº ATRIBUTOS: 2316					
Nº INSTANCIAS: 4686					
ACIERTOS: 4607 / 98,31					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO TEXT	0,984	0,044	0,998	0,984	0,991
TEXT	0,956	0,016	0,709	0,956	0,814
MATRIZ DE CONTINGENCIA					

4434	71
8	173

ErrorR($S \notin clase$ **) = 0,016 +- 0,004**

ErrorR($S \in clase$ **) = 0,044 +- 0,030**

ErrorR(S) = 0,017 +- 0,004

FIGURA IV.50: Evaluación Cross Validation Textos BoW std

IV.7 EVALUACIÓN BoW MEJORADO

IV.7.1 Asociaciones

MÉTODO: BoW mejorado					
Nº ATRIBUTOS: 522					
Nº INSTANCIAS (train/test): 2151 / 2582					
ACIERTOS: 1419 / 54,96%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO ASOC.	0,549	0,143	0,999	0,549	0,708
ASOCIAC.	0,857	0,451	0,005	0,857	0,01
MATRIZ DE CONTINGENCIA					

1413	1162
1	6

ErrorR($S \notin clase$ **) = 0,451 +- 0,019**

ErrorR($S \in clase$ **) = 0,143 +- 0,259**

ErrorR(S) = 0,450 +- 0,019

FIGURA IV.51: Evaluación DS1/DS2 Asociaciones BoW Improv

MÉTODO: BoW mejorado					
Nº ATRIBUTOS: 522					
Nº INSTANCIAS (train/test): 2582 / 2151					
ACIERTOS: 1457 / 67,74%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO ASO	0,679	0,471	0,995	0,679	0,807
ASOCIA.	0,529	0,321	0,013	0,529	0,025
MATRIZ DE CONTINGENCIA					

1448	686
8	9

ErrorR($S \notin clase$ **) = 0,321 +- 0,020**

ErrorR($S \in clase$ **) = 0,471 +- 0,237**

ErrorR(S) = 0,323 +- 0,020

FIGURA IV.52: Evaluación DS2/DS1 Asociaciones BoW Improv

MÉTODO: BoW mejorado					
Nº ATRIBUTOS: 522					
Nº INSTANCIAS: 4733					
ACIERTOS: 2876 / 60,8%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO ASO	0,608	0,392	0,997	0,608	0,755
ASOCIA	0,625	0,375	0,008	0,625	0,016
MATRIZ DE CONTINGENCIA					

2861	1848
9	15

ErrorR($S \notin clase$ **) = 0,392 +- 0,014**

ErrorR($S \in clase$ **) = 0,375 +- 0,194**

ErrorR(S) = 0,392 +- 0,014

FIGURA IV.53: Evaluación 2x2 Asociaciones BoW Improv

IV.7.2 Blogs

MÉTODO: BoW mejorado					
Nº ATRIBUTOS: 4516					
Nº INSTANCIAS (train/test): 2151 / 2582					
ACIERTOS: 2079 / 80,5%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO BLOG	0,839	0,922	0,951	0,839	0,892
BLOG	0,078	0,161	0,022	0,078	0,035
MATRIZ DE CONTINGENCIA					
2070	396				
107	9				

ErrorR($S \notin clase$ **) = 0,161 +- 0,014**

ErrorR($S \in clase$ **) = 0,922 +- 0,049**

ErrorR(S) = 0,195 +- 0,015

FIGURA IV.54: Evaluación DS1/DS2 Blogs BoW Improv

MÉTODO: BoW mejorado					
Nº ATRIBUTOS: 4516					
Nº INSTANCIAS (train/test): 2582 / 2151					
ACIERTOS: 1773 / 82,43%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO BLOG	0,846	0,26	0,928	0,846	0,885
BLOG	0,74	0,154	0,549	0,74	0,63
MATRIZ DE CONTINGENCIA					
1451	265				
113	322				

ErrorR($S \notin clase$ **) = 0,154 +- 0,017**

ErrorR($S \in clase$ **) = 0,260 +- 0,041**

ErrorR(S) = 0,176 +- 0,016

FIGURA IV.55: Evaluación DS2/DS1 Blogs BoW Improv

MÉTODO: BoW mejorado					
Nº ATRIBUTOS: 4516					
Nº INSTANCIAS: 4733					
ACIERTOS: 3852 / 81,4%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO BLOG	0,842	0,158	0,941	0,842	0,889
BLOG	0,601	0,399	0,334	0,601	0,429
MATRIZ DE CONTINGENCIA					
3521	661				
220	331				

ErrorR($S \notin clase$ **) = 0,158 +- 0,011**

ErrorR($S \in clase$ **) = 0,399 +- 0,041**

ErrorR(S) = 0,186 +- 0,011

FIGURA IV.56: Evaluación 2x2 Blogs BoW Improv

IV.7.3 Compañías

MÉTODO: BoW mejorado					
Nº ATRIBUTOS: 3091					
Nº INSTANCIAS (train/test): 2151 / 2582					
ACIERTOS: 2457 / 95,16%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO COMP	0,589	0,004	0,949	0,589	0,726
COMPAÑÍA	0,996	0,411	0,952	0,996	0,973
MATRIZ DE CONTINGENCIA					
166	116				
9	2291				

ErrorR($S \notin clase$ **) = 0,411 +- 0,057**

ErrorR($S \in clase$ **) = 0,004 +- 0,003**

ErrorR(S) = 0,048 +- 0,008

FIGURA IV.57: Evaluación DS1/DS2 Compañías BoW Improv

MÉTODO: BoW mejorado					
Nº ATRIBUTOS: 3091					
Nº INSTANCIAS (train/test): 2582 / 2151					
ACIERTOS: 1392 / 64,71%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO COMP	0,653	0,394	0,917	0,653	0,763
COMPAÑÍA	0,606	0,347	0,209	0,606	0,311
MATRIZ DE CONTINGENCIA					
1221	648				
111	171				

ErrorR($S \notin clase$ **) = 0,347 +- 0,022**

ErrorR($S \in clase$ **) = 0,394 +- 0,057**

ErrorR(S) = 0,353 +- 0,020

FIGURA IV.58: Evaluación DS2/DS1 Compañías BoW Improv

MÉTODO: BoW mejorado					
Nº ATRIBUTOS: 3091					
Nº INSTANCIAS: 4733					
ACIERTOS: 3849 / 81,3%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO COMP.	0,645	0,355	0,920	0,645	0,758
COMPAÑÍA	0,954	0,046	0,763	0,954	0,848
MATRIZ DE CONTINGENCIA					
1387	764				
120	2462				

ErrorR($S \notin clase$ **) = 0,355 +- 0,020**

ErrorR($S \in clase$ **) = 0,046 +- 0,008**

ErrorR(S) = 0,187 +- 0,011

FIGURA IV.59: Evaluación 2x2 Compañías BoW Improv

IV.7.4 Festivales

MÉTODO: BoW mejorado					
Nº ATRIBUTOS: 5375					
Nº INSTANCIAS (train/test): 2151 / 2582					
ACIERTOS: 1173 / 45,43%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FEST	0,453	0	1	0,453	0,624
FESTIVAL	1	0,547	0,004	1	0,008
MATRIZ DE CONTINGENCIA					
1167	1409				
0	6				

ErrorR($S \notin clase$ **) = 0,547 +- 0,019**

ErrorR($S \in clase$ **) = 1 + 0**

ErrorR(S) = 0,546 +- 0,019

FIGURA IV.60: Evaluación DS1/DS2 Festivales BoW Improv

MÉTODO: BoW mejorado					
Nº ATRIBUTOS: 5375					
Nº INSTANCIAS (train/test): 2582 / 2151					
ACIERTOS: 1548 / 71,97%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FEST	0,843	0,514	0,757	0,843	0,798
FESTIVAL	0,486	0,157	0,619	0,486	0,544
MATRIZ DE CONTINGENCIA					
1188	222				
381	360				

ErrorR($S \notin clase$ **) = 0,157 +- 0,019**

ErrorR($S \in clase$ **) = 0,514 +- 0,036**

ErrorR(S) = 0,280 +- 0,019

FIGURA IV.61: Evaluación DS2/DS1 Festivales BoW Improv

MÉTODO: BoW mejorado					
Nº ATRIBUTOS: 5375					
Nº INSTANCIAS: 4733					
ACIERTOS: 2721 / 57,5%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FEST.	0,591	0,409	0,861	0,591	0,701
FESTIVAL	0,490	0,510	0,183	0,490	0,267
MATRIZ DE CONTINGENCIA					
2355	1631				
381	366				

ErrorR($S \notin clase$ **) = 0,409 +- 0,015**

ErrorR($S \in clase$ **) = 0,510 +- 0,036**

ErrorR(S) = 0,425 +- 0,014

FIGURA IV.62: Evaluación 2x2 Festival BoW Improv

IV.7.5 Formación

MÉTODO: BoW mejorado					
Nº ATRIBUTOS: 3274					
Nº INSTANCIAS (train/test): 2151 / 2582					
ACIERTOS: 1781 / 68,98%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FORM.	0,696	0,448	0,973	0,696	0,811
FORMAC.	0,552	0,304	0,071	0,552	0,126
MATRIZ DE CONTINGENCIA					
1723	754				
47	58				

ErrorR($S \notin clase$) = 0,304 +- 0,018
ErrorR($S \in clase$) = 0,448 +- 0,095
ErrorR(S) = 0,310 +- 0,018
FIGURA IV.63: Evaluación DS1/DS2 Formación BoW Improv

MÉTODO: BoW mejorado					
Nº ATRIBUTOS: 3274					
Nº INSTANCIAS (train/test): 2582 / 2151					
ACIERTOS: 989 / 45,98%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FORM	0,418	0,078	0,983	0,418	0,586
FORMACIÓN	0,922	0,582	0,126	0,922	0,221
MATRIZ DE CONTINGENCIA					
824	1148				
14	165				

ErrorR($S \notin clase$) = 0,582 +- 0,022
ErrorR($S \in clase$) = 0,078 +- 0,039
ErrorR(S) = 0,540 +- 0,021
FIGURA IV.64: Evaluación DS2/DS1 Formación BoW Improv

MÉTODO: BoW mejorado					
Nº ATRIBUTOS: 3274					
Nº INSTANCIAS: 4733					
ACIERTOS: 2770 / 58,5%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FORM.	0,572	0,428	0,977	0,572	0,722
FORMACIÓN	0,785	0,215	0,105	0,785	0,185
MATRIZ DE CONTINGENCIA					
2547	1902				
61	223				

ErrorR($S \notin clase$) = 0,428 +- 0,015
ErrorR($S \in clase$) = 0,215 +- 0,048
ErrorR(S) = 0,415 +- 0,014
FIGURA IV.65: Evaluación 2x2 Formación BoW Improv

IV.7.6 Revistas

MÉTODO: BoW mejorado					
Nº ATRIBUTOS: 2557					
Nº INSTANCIAS (train/test): 2151 / 2582					
ACIERTOS: 508 / 19,67%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO REV	0,197	0,667	0,996	0,197	0,328
REVISTA	0,333	0,803	0	0,333	0,001
MATRIZ DE CONTINGENCIA					
507	2072				
2	1				

ErrorR($S \notin clase$ **) = 0,803 +- 0,015**

ErrorR($S \in clase$ **) = 0,667 +- 0,533**

ErrorR(S) = 0,803 +- 0,015

FIGURA IV.66: Evaluación DS1/DS2 Revistas BoW Improv

MÉTODO: BoW mejorado					
Nº ATRIBUTOS: 2557					
Nº INSTANCIAS (train/test): 2582 / 2151					
ACIERTOS: 1373 / 63,83%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO REV	0,652	0,847	0,965	0,652	0,778
REVISTA	0,153	0,348	0,012	0,153	0,023
MATRIZ DE CONTINGENCIA					
1364	728				
50	9				

ErrorR($S \notin clase$ **) = 0,348 +- 0,020**

ErrorR($S \in clase$ **) = 0,847 +- 0,092**

ErrorR(S) = 0,362 +- 0,020

FIGURA IV.67: Evaluación DS2/DS1 Revistas BoW Improv

MÉTODO: BoW mejorado					
Nº ATRIBUTOS: 2557					
Nº INSTANCIAS: 4733					
ACIERTOS: 1881 / 39,7%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO REV.	0,401	0,599	0,973	0,401	0,567
REVISTA	0,161	0,839	0,004	0,161	0,007
MATRIZ DE CONTINGENCIA					
1871	2800				
52	10				

ErrorR($S \notin clase$ **) = 0,599 +- 0,014**

ErrorR($S \in clase$ **) = 0,839 +- 0,092**

ErrorR(S) = 0,603 +- 0,014

FIGURA IV.68: Evaluación 2x2 Revistas BoW Improv

IV.7.7 Salas Alternativas

MÉTODO: BoW mejorado					
Nº ATRIBUTOS: 2989					
Nº INSTANCIAS (train/test): 2151 / 2582					
ACIERTOS: 407 / 15,75%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO SALA	0,143	0,044	0,995	0,143	0,251
SALA	0,956	0,857	0,019	0,956	0,038
MATRIZ DE CONTINGENCIA					
364	2173				
2	43				

ErrorR($S \notin clase$ **) = 0,857 +- 0,014**

ErrorR($S \in clase$ **) = 0,044 +- 0,060**

ErrorR(S) = 0,842 +- 0,014

FIGURA IV.69: Evaluación DS1/DS2 Salas Alternativas BoW Improv

MÉTODO: BoW mejorado					
Nº ATRIBUTOS: 2989					
Nº INSTANCIAS (train/test): 2582 / 2151					
ACIERTOS: 1550 / 72,06%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO SALA	0,737	0,404	0,933	0,737	0,823
SALA	0,596	0,263	0,23	0,596	0,331
MATRIZ DE CONTINGENCIA					
1401	500				
101	149				

ErrorR($S \notin clase$ **) = 0,263 +- 0,020**

ErrorR($S \in clase$ **) = 0,404 +- 0,061**

ErrorR(S) = 0,279 +- 0,019

FIGURA IV.70: Evaluación DS2/DS1 Salas Alternativas BoW Improv

MÉTODO: BoW mejorado					
Nº ATRIBUTOS: 2989					
Nº INSTANCIAS: 4733					
ACIERTOS: 1957 / 41,3%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO SALA	0,398	0,602	0,945	0,398	0,560
SALA	0,651	0,349	0,067	0,651	0,122
MATRIZ DE CONTINGENCIA					
1765	2673				
103	192				

ErrorR($S \notin clase$ **) = 0,602 +- 0,014**

ErrorR($S \in clase$ **) = 0,349 +- 0,054**

ErrorR(S) = 0,587 +- 0,014

FIGURA IV.71: Evaluación 2x2 Salas Alternativas BoW Improv

IV.7.8 Textos

MÉTODO: BoW mejorado					
Nº ATRIBUTOS: 2316					
Nº INSTANCIAS: 4753					
ACIERTOS: 4359 / 91,71%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO TEXTO	0,918	0,099	0,996	0,918	0,955
TEXTO	0,901	0,082	0,304	0,901	0,454
MATRIZ DE CONTINGENCIA					
4195	376				
18	164				

ErrorR($S \notin clase$ **) = 0,082 +- 0,008**

ErrorR($S \in clase$ **) = 0,099 +- 0,043**

ErrorR(S) = 0,083 +- 0.008

FIGURA IV.72: Evaluación Cross Textos BoW Improv

IV.8 EVALUACIÓN BoW URL

IV.8.1 Asociaciones

MÉTODO: Urls					
Nº ATRIBUTOS: 522					
Nº INSTANCIAS (train/test): 2184 / 2590					
ACIERTOS: 146 / 5,64%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO ASOC.	0,055	0,375	0,979	0,055	0,103
ASOCIAC.	0,625	0,945	0,002	0,625	0,004
MATRIZ DE CONTINGENCIA					
141	2441				
3	5				

ErrorR($S \notin clase$ **) = 0,945 +- 0,009**

ErrorR($S \in clase$ **) = 0,375 +- 0,335**

ErrorR(S) = 0,944 +- 0,009

FIGURA IV.73: Evaluación DS1/DS2 Asociaciones BoW Url

MÉTODO: Urls					
Nº ATRIBUTOS: 522					
Nº INSTANCIAS (train/test): 2590 / 2184					
ACIERTOS: 99 / 4,53%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO ASO.	0,039	0,167	0,966	0,039	0,075
ASOCIA.	0,833	0,961	0,007	0,833	0,014
MATRIZ DE CONTINGENCIA					
84	2082				
3	15				

ErrorR($S \notin clase$ **) = 0,961 +- 0,008**

ErrorR($S \in clase$ **) = 0,167 +- 0,172**

ErrorR(S) = 0,955 +- 0,009

FIGURA IV.74: Evaluación DS2/DS1 Asociaciones BoW Url

MÉTODO: Urls					
Nº ATRIBUTOS: 522					
Nº INSTANCIAS: 4774					
ACIERTOS: 245 / 5,1%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO ASOC	0,047	0,953	0,974	0,047	0,090
ASOCIA.	0,769	0,231	0,004	0,769	0,009
MATRIZ DE CONTINGENCIA					
225	4523				
6	20				

ErrorR($S \notin clase$ **) = 0,953 +- 0,006**

ErrorR($S \in clase$ **) = 0,231 +- 0,162**

ErrorR(S) = 0,949 +- 0,006

FIGURA IV.75: Evaluación 2x2 Asociaciones BoW Url

IV.8.2 Blogs

MÉTODO: Urls					
Nº ATRIBUTOS: 4516					
Nº INSTANCIAS (train/test): 2184 / 2590					
ACIERTOS: 2499 / 96,49%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO BLOG	0,966	0,052	0,997	0,966	0,981
BLOG	0,948	0,035	0,564	0,948	0,707
MATRIZ DE CONTINGENCIA					
2389	85				
6	110				

ErrorR($S \notin clase$ **) = 0,034 +- 0,007**
ErrorR($S \in clase$ **) = 0,052 +- 0,040**
ErrorR(S) = 0,035 +- 0,007
FIGURA IV.76: Evaluación DS1/DS2 Blogs BoW Url

MÉTODO: Urls					
Nº ATRIBUTOS: 4516					
Nº INSTANCIAS (train/test): 2590 / 2184					
ACIERTOS: 1614 / 73,90%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO BLOG	0,747	0,292	0,911	0,747	0,821
BLOG	0,708	0,253	0,41	0,708	0,519
MATRIZ DE CONTINGENCIA					
1306	443				
127	308				

ErrorR($S \notin clase$ **) = 0,253 +- 0,020**
ErrorR($S \in clase$ **) = 0,292 +- 0,043**
ErrorR(S) = 0,261 +- 0,018
FIGURA IV.77: Evaluación DS2/DS1 Blogs BoW Url

MÉTODO: Urls					
Nº ATRIBUTOS: 4516					
Nº INSTANCIAS: 4774					
ACIERTOS: 4113 / 86,2%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO BLOG	0,875	0,125	0,965	0,875	0,918
BLOG	0,759	0,241	0,442	0,759	0,558
MATRIZ DE CONTINGENCIA					
3695	528				
133	418				

ErrorR($S \notin clase$ **) = 0,125 +- 0,010**
ErrorR($S \in clase$ **) = 0,241 +- 0,036**
ErrorR(S) = 0,138 +- 0,010
FIGURA IV.78: Evaluación 2x2 Blogs BoW Url

IV.8.3 Compañías

MÉTODO: Urls					
Nº ATRIBUTOS: 3091					
Nº INSTANCIAS (train/test): 2184 / 2590					
ACIERTOS: 291 / 11,24%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO COMP	0,254	0,905	0,034	0,254	0,06
COMPAÑÍA	0,095	0,746	0,505	0,095	0,159
MATRIZ DE CONTINGENCIA					
73	214				
2085	218				

ErrorR($S \notin clase$ **) = 0,746 +- 0,050**
ErrorR($S \in clase$ **) = 0,905 +- 0,012**
ErrorR(S) = 0,888 +- 0,012
FIGURA IV.79: Evaluación DS1/DS2 Compañías BoW Url

MÉTODO: Urls					
Nº ATRIBUTOS: 3091					
Nº INSTANCIAS (train/test): 2590 / 2184					
ACIERTOS: 1855 / 84,94%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO COMP	0,988	0,997	0,858	0,988	0,919
COMPAÑÍA	0,003	0,012	0,042	0,003	0,006
MATRIZ DE CONTINGENCIA					
1854	21				
306	1				

ErrorR($S \notin clase$ **) = 0,011 +- 0,005**
ErrorR($S \in clase$ **) = 0,997 +- 0,006**
ErrorR(S) = 0,151 +- 0,015
FIGURA IV.80: Evaluación DS2/DS1 Compañías BoW Url

MÉTODO: Urls					
Nº ATRIBUTOS: 3091					
Nº INSTANCIAS: 4772					
ACIERTOS: 2146 / 45,0%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO COMP.	0,891	0,109	0,446	0,891	0,595
COMPAÑÍA	0,084	0,916	0,482	0,084	0,143
MATRIZ DE CONTINGENCIA					
1927	235				
2391	219				

ErrorR($S \notin clase$ **) = 0,109 +- 0,013**
ErrorR($S \in clase$ **) = 0,916 +- 0,011**
ErrorR(S) = 0,550 +- 0,014
FIGURA IV.81: Evaluación 2x2 Compañías BoW Url

IV.8.4 Festivales

MÉTODO: Urls					
Nº ATRIBUTOS: 5375					
Nº INSTANCIAS (train/test): 2184 / 2590					
ACIERTOS: 2578 / 99,54%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FEST	0,998	1	0,998	0,998	0,998
FESTIVAL	0	0.002	0	0	0
MATRIZ DE CONTINGENCIA					
2578	6				
6	0				

ErrorR($S \notin clase$ **) = 0,002 +- 0,002**

ErrorR($S \in clase$ **) = 1 +- 0**

ErrorR(S) = 0,005 +- 0,003

FIGURA IV.82: Evaluación DS1/DS2 Festivales BoW Url

MÉTODO: Urls					
Nº ATRIBUTOS: 5375					
Nº INSTANCIAS (train/test): 2590 / 2184					
ACIERTOS: 1479 / 67,72%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FEST	0,533	0,042	0,961	0,533	0,686
FESTIVAL	0,958	0,467	0,513	0,958	0,668
MATRIZ DE CONTINGENCIA					
769	674				
31	710				

ErrorR($S \notin clase$ **) = 0,467 +- 0,026**

ErrorR($S \in clase$ **) = 0,042 +- 0,014**

ErrorR(S) = 0,323 +- 0,020

FIGURA IV.83: Evaluación DS2/DS1 Festivales BoW Url

MÉTODO: Urls					
Nº ATRIBUTOS: 5375					
Nº INSTANCIAS: 4774					
ACIERTOS: 4057 / 85,0%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FEST	0,831	0,169	0,989	0,831	0,903
FESTIVAL	0,950	0,050	0,511	0,950	0,664
MATRIZ DE CONTINGENCIA					
3347	680				
37	710				

ErrorR($S \notin clase$ **) = 0,169 +- 0,012**

ErrorR($S \in clase$ **) = 0,050 +- 0,016**

ErrorR(S) = 0,150 +- 0,010

FIGURA IV.84: Evaluación 2x2 Festivales BoW Url

IV.8.5 Formación

MÉTODO: Urls					
Nº ATRIBUTOS: 3274					
Nº INSTANCIAS (train/test): 2184 / 2590					
ACIERTOS: 2350 / 90,73%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FORM.	0,917	0,321	0,985	0,917	0,95
FORMAC.	0,679	0,083	0,265	0,679	0,381
MATRIZ DE CONTINGENCIA					
2276	205				
35	74				

ErrorR($S \notin clase$ **) = 0,083 +- 0,011**
ErrorR($S \in clase$ **) = 0,321 +- 0,088**
ErrorR(S) = 0,093 +- 0,011
FIGURA IV.85: Evaluación DS1/DS2 Formación BoW Url

MÉTODO: Urls					
Nº ATRIBUTOS: 3274					
Nº INSTANCIAS (train/test): 2590 / 2184					
ACIERTOS: 248 / 11,35%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FORM	0,037	0,039	0,914	0,037	0,071
FORMACIÓN	0,961	0,963	0,083	0,961	0,152
MATRIZ DE CONTINGENCIA					
74	1929				
7	174				

ErrorR($S \notin clase$ **) = 0,963 +- 0,008**
ErrorR($S \in clase$ **) = 0,039 +- 0,028**
ErrorR(S) = 0,886 +- 0,013
FIGURA IV.86: Evaluación DS2/DS1 Formación BoW Url

MÉTODO: Urls					
Nº ATRIBUTOS: 3274					
Nº INSTANCIAS: 4774					
ACIERTOS: 2598 / 54,4%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FORM.	0,524	0,476	0,982	0,524	0,684
FORMACIÓN	0,855	0,145	0,104	0,855	0,186
MATRIZ DE CONTINGENCIA					
2350	2134				
42	248				

ErrorR($S \notin clase$ **) = 0,476 +- 0,015**
ErrorR($S \in clase$ **) = 0,145 +- 0,041**
ErrorR(S) = 0,456 +- 0,014
FIGURA IV.87: Evaluación 2x2 Formación BoW Url

IV.8.6 Revistas

MÉTODO: Urls					
Nº ATRIBUTOS: 2557					
Nº INSTANCIAS (train/test): 2184 / 2590					
ACIERTOS: 2434 / 93,98%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO REV	0,94	0,667	0,999	0,94	0,969
REVISTA	0,333	0,06	0,006	0,333	0,013
MATRIZ DE CONTINGENCIA					
2433	154				
2	1				

ErrorR($S \notin clase$ **) = 0,060 +- 0,009**
ErrorR($S \in clase$ **) = 0,667 +- 0,533**
ErrorR(S) = 0,060 +- 0,009
FIGURA IV.88: Evaluación DS1/DS2 Revistas BoW Url

MÉTODO: Urls					
Nº ATRIBUTOS: 2557					
Nº INSTANCIAS (train/test): 2590 / 2184					
ACIERTOS: 554 / 25,37 %					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO REV	0,244	0,39	0,957	0,244	0,389
REVISTA	0,61	0,756	0,022	0,61	0,042
MATRIZ DE CONTINGENCIA					
518	1607				
23	36				

ErrorR($S \notin clase$ **) = 0,756 +- 0,018**
ErrorR($S \in clase$ **) = 0,390 +- 0,124**
ErrorR(S) = 0,746 +- 0,018
FIGURA IV.89: Evaluación DS2/DS1 Revistas BoW Url

MÉTODO: Urls					
Nº ATRIBUTOS: 2557					
Nº INSTANCIAS: 4774					
ACIERTOS: 2988 / 62,6%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO REV.	0,626	0,374	0,992	0,626	0,768
REVISTA	0,597	0,403	0,021	0,597	0,040
MATRIZ DE CONTINGENCIA					
2951	1761				
25	37				

ErrorR($S \notin clase$ **) = 0,374 +- 0,014**
ErrorR($S \in clase$ **) = 0,403 +- 0,122**
ErrorR(S) = 0,374 +- 0,014
FIGURA IV.90: Evaluación 2x2 Revistas BoW Url

IV.8.7 Salas Alternativas

MÉTODO: Urls					
Nº ATRIBUTOS: 2989					
Nº INSTANCIAS (train/test): 2184 / 2590					
ACIERTOS: 935 / 36,10%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO SALA	0,356	0,378	0,982	0,356	0,523
SALA	0,622	0,644	0,017	0,622	0,033
MATRIZ DE CONTINGENCIA					
907	1638				
17	28				

ErrorR($S \notin clase$ **) = 0,644 +- 0,019**
ErrorR($S \in clase$ **) = 0,378 +- 0,142**
ErrorR(S) = 0,639 +- 0,018
FIGURA IV.91: Evaluación DS1/DS2 Salas Alternativas BoW Url

MÉTODO: Urls					
Nº ATRIBUTOS: 2989					
Nº INSTANCIAS (train/test): 2590 / 2184					
ACIERTOS: 788 / 36,08%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO SALA	0,294	0,133	0,943	0,294	0,448
SALA	0,867	0,706	0,14	0,867	0,24
MATRIZ DE CONTINGENCIA					
567	1362				
34	221				

ErrorR($S \notin clase$ **) = 0,706 +- 0,020**
ErrorR($S \in clase$ **) = 0,133 +- 0,042**
ErrorR(S) = 0,639 +- 0,020
FIGURA IV.92: Evaluación DS2/DS1 Salas Alternativas BoW Url

MÉTODO: Urls					
Nº ATRIBUTOS: 2989					
Nº INSTANCIAS: 4774					
ACIERTOS: 1723 / 36,1%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO SALA	0,329	0,671	0,967	0,329	0,491
SALA	0,830	0,170	0,077	0,830	0,140
MATRIZ DE CONTINGENCIA					
1474	3000				
51	249				

ErrorR($S \notin clase$ **) = 0,671 +- 0,014**
ErrorR($S \in clase$ **) = 0,170 +- 0,043**
ErrorR(S) = 0,639 +- 0,014
FIGURA IV.93: Evaluación 2x2 Salas Alternativas BoW Url

IV.8.8 Textos

MÉTODO: Urls					
Nº ATRIBUTOS: 2316					
Nº INSTANCIAS: 4720					
ACIERTOS: 4605 / 97,56%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO TEXTO	0,998	0,582	0,977	0,998	0,987
TEXTO	0,418	0,002	0,894	0,418	0,569
MATRIZ DE CONTINGENCIA					
4529	9				
106	76				

ErrorR($S \notin clase$ **) = 0,002 +- 0,001**

ErrorR($S \in clase$ **) = 0,582 +- 0,072**

ErrorR(S) = 0,024 +- 0,004

FIGURA IV.94: Evaluación Cross Textos BoW Url

IV.9 EVALUACIÓN BoW L&H&U

IV.9.1 Asociaciones

MÉTODO: H&L&U					
Nº ATRIBUTOS: 655					
Nº INSTANCIAS (train/test): 2049 / 2568					
ACIERTOS: 2096 / 81,62%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO ASOC.	0,816	0,25	0,999	9,816	0,899
ASOCIAC.	0,75	0,184	0,013	0,75	0,025
MATRIZ DE CONTINGENCIA					
2090	470				
2	6				

ErrorR($S \notin clase$ **) = 0,184+-0,015**

ErrorR($S \in clase$ **) = 0,250 +- 0,300**

ErrorR(S) = 0,184 +- 0,015

FIGURA IV.95: Evaluación DS1/DS2 Asociaciones H&L&U

MÉTODO: H&L&U					
Nº ATRIBUTOS: 655					
Nº INSTANCIAS (train/test): 2568 / 2049					
ACIERTOS: 913 / 44,56%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO ASO.	0,445	0,5	0,992	0,445	0,615
ASOCIA.	0,5	0,555	0,006	0,5	0,012
MATRIZ DE CONTINGENCIA					
906	1129				
7	7				

ErrorR($S \notin clase$ **) = 0,555 +- 0,022**

ErrorR($S \in clase$ **) = 0,500 +- 0,262**

ErrorR(S) = 0,554 +- 0,022

FIGURA IV.96: Evaluación DS2/DS1 Asociaciones H&L&U

MÉTODO: H&L&U					
Nº ATRIBUTOS: 655					
Nº INSTANCIAS: 4617					
ACIERTOS: 3009 / 65,2%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO ASOC.	0,652	0,348	0,997	0,652	0,788
ASOCIA	0,591	0,409	0,008	0,591	0,016
MATRIZ DE CONTINGENCIA					
2996	1599				
9	13				

ErrorR($S \notin clase$ **) = 0,348 +- 0,014**

ErrorR($S \in clase$ **) = 0,409 +- 0,205**

ErrorR(S) = 0,348 +- 0,014

FIGURA IV.97: Evaluación 2x2 Asociaciones H&L&U

IV.9.2 Blogs

MÉTODO: H&L&U					
Nº ATRIBUTOS: 4165					
Nº INSTANCIAS (train/test): 1989 / 2540					
ACIERTOS: 2432 / 95,75%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO BLOG	1	0,922	0,958	1	0,978
BLOG	0,078	0	0,9	0,078	0,143
MATRIZ DE CONTINGENCIA					
2423	1				
107	9				

ErrorR($S \notin clase$ **) = 1 +- 0,001**
ErrorR($S \in clase$ **) = 0,922 +- 0,049**
ErrorR(S) = 0,043 +- 0,008
FIGURA IV.98: Evaluación DS1/DS2 Blogs H&L&U

MÉTODO: H&L&U					
Nº ATRIBUTOS: 4165					
Nº INSTANCIAS (train/test): 2540 / 1989					
ACIERTOS: 1683 / 84,61%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO BLOG	0,827	0,085	0,972	0,827	0,894
BLOG	0,915	0,173	0,597	0,915	0,722
MATRIZ DE CONTINGENCIA					
1285	269				
37	398				

ErrorR($S \notin clase$ **) = 0,173 +- 0,019**
ErrorR($S \in clase$ **) = 0,085 +- 0,026**
ErrorR(S) = 0,154 +- 0,016
FIGURA IV.99: Evaluación DS2/DS1 Blogs H&L&U

MÉTODO: H&L&U					
Nº ATRIBUTOS: 4165					
Nº INSTANCIAS: 4529					
ACIERTOS: 4115 / 90,9%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO BLOG	0,932	0,068	0,963	0,932	0,947
BLOG	0,739	0,261	0,601	0,739	0,663
MATRIZ DE CONTINGENCIA					
3708	270				
144	407				

ErrorR($S \notin clase$ **) = 0,068 +- 0,008**
ErrorR($S \in clase$ **) = 0,261 +- 0,037**
ErrorR(S) = 0,091 +- 0,008
FIGURA IV.100: Evaluación 2x2 Blogs H&L&U

IV.9.3 Compañías

MÉTODO: H&L&U					
Nº ATRIBUTOS: 2119					
Nº INSTANCIAS (train/test): 2026 / 2572					
ACIERTOS: 2489 / 96,77%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO COMP	0,745	0,006	0,94	0,745	0,832
COMPAÑÍA	0,994	0,255	0,97	0,994	0,982
MATRIZ DE CONTINGENCIA					
205	70				
13	2284				

ErrorR($S \notin clase$ **) = 0,255 +- 0,051**
ErrorR($S \in clase$ **) = 0,006 +- 0,003**
ErrorR(S) = 0,032 +- 0,007
FIGURA IV.101: Evaluación DS1/DS2 Compañías H&L&U

MÉTODO: H&L&U					
Nº ATRIBUTOS: 2119					
Nº INSTANCIAS (train/test): 2572 / 2026					
ACIERTOS: 1093 / 53,95%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO COMP.	0,547	0,514	0,885	0,547	0,676
COMPAÑÍA	0,486	0,453	0,129	0,486	0,203
MATRIZ DE CONTINGENCIA					
974	807				
126	119				

ErrorR($S \notin clase$ **) = 0,453 +- 0,023**
ErrorR($S \in clase$ **) = 0,514 +- 0,063**
ErrorR(S) = 0,461 +- 0,022
FIGURA IV.102: Evaluación DS2/DS1 Compañías H&L&U

MÉTODO: H&L&U					
Nº ATRIBUTOS: 2119					
Nº INSTANCIAS: 4598					
ACIERTOS: 3582 / 77,9%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO COMP.	0,573	0,427	0,895	0,573	0,699
COMPAÑÍA	0,945	0,055	0,733	0,945	0,825
MATRIZ DE CONTINGENCIA					
1179	877				
139	2403				

ErrorR($S \notin clase$ **) = 0,427 +- 0.021**
ErrorR($S \in clase$ **) = 0,055 +- 0,009**
ErrorR(S) = 0,221 +- 0,012
FIGURA IV.103: Evaluación 2x2 Compañías H&L&U

IV.9.4 Festivales

MÉTODO: H&L&U					
Nº ATRIBUTOS: 1951					
Nº INSTANCIAS (train/test): 2026 / 2572					
ACIERTOS: 2386 / 92,77%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FEST	0,928	0,333	1	0,928	0,962
FESTIVAL	0,667	0,072	0,011	0,667	0,021
MATRIZ DE CONTINGENCIA					
2384	185				
1	2				

ErrorR($S \notin clase$ **) = 0,072 +- 0,010**

ErrorR($S \in clase$ **) = 0,333 +- 0,533**

ErrorR(S) = 0,072 +- 0,010

FIGURA IV.104: Evaluación DS1/DS2 Festivales H&L&U

MÉTODO: H&L&U					
Nº ATRIBUTOS: 1951					
Nº INSTANCIAS (train/test): 2572 / 2026					
ACIERTOS: 1760 / 86,87%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FEST	0,836	0,067	0,96	0,836	0,894
FESTIVAL	0,933	0,164	0,745	0,933	0,828
MATRIZ DE CONTINGENCIA					
1118	220				
46	642				

ErrorR($S \notin clase$ **) = 0,164 +- 0,020**

ErrorR($S \in clase$ **) = 0,067 +- 0,019**

ErrorR(S) = 0,131 +- 0,015

FIGURA IV.105: Evaluación DS2/DS1 Festivales H&L&U

MÉTODO: H&L&U					
Nº ATRIBUTOS: 1951					
Nº INSTANCIAS: 4598					
ACIERTOS: 4146 / 90,2%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FEST	0,896	0,104	0,987	0,896	0,939
FESTIVAL	0,932	0,068	0,614	0,932	0,740
MATRIZ DE CONTINGENCIA					
3502	405				
47	644				

ErrorR($S \notin clase$ **) = 0,104 +- 0,010**

ErrorR($S \in clase$ **) = 0,068 +- 0,019**

ErrorR(S) = 0,098 +- 0,009

FIGURA IV.106: Evaluación 2x2 Festivales H&L&U

IV.9.5 Formación

MÉTODO: H&L&U					
Nº ATRIBUTOS: 2089					
Nº INSTANCIAS (train/test): 1989 / 2540					
ACIERTOS: 2471 / 97,28%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FORM.	0,99	0,433	0,982	0,99	0,986
FORMAC.	0,567	0,01	0,711	0,567	0,631
MATRIZ DE CONTINGENCIA					
2412	24				
45	59				

ErrorR($S \notin clase$ **) = 0,010 +- 0,004**

ErrorR($S \in clase$ **) = 0,433 +- 0,095**

ErrorR(S) = 0,027 +- 0,006

FIGURA IV.107: Evaluación DS1/DS2 Formación H&L&U

MÉTODO: H&L&U					
Nº ATRIBUTOS: 2098					
Nº INSTANCIAS (train/test): 2540 / 1989					
ACIERTOS: 1339 / 67,32%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FORM	0,652	0,06	0,993	0,652	0,787
FORMACIÓN	0,94	0,348	0,179	0,94	0,301
MATRIZ DE CONTINGENCIA					
1199	641				
9	140				

ErrorR($S \notin clase$ **) = 0,348 +- 0,022**

ErrorR($S \in clase$ **) = 0,060 +- 0,038**

ErrorR(S) = 0,327 +- 0,021

FIGURA IV.108: Evaluación DS2/DS1 Formación H&L&U

MÉTODO: H&L&U					
Nº ATRIBUTOS: 2098					
Nº INSTANCIAS: 4529					
ACIERTOS: 3810 / 84,1%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO FORM	0,844	0,156	0,985	0,844	0,909
FORMACIÓN	0,787	0,213	0,230	0,787	0,356
MATRIZ DE CONTINGENCIA					
3611	665				
54	199				

ErrorR($S \notin clase$ **) = 0,156 +- 0,011**

ErrorR($S \in clase$ **) = 0,213 +- 0,050**

ErrorR(S) = 0,159 +- 0,011

FIGURA IV.109: Evaluación 2x2 Formación H&L&U

IV.9.6 Revistas

MÉTODO: H&L&U					
Nº ATRIBUTOS: 1414					
Nº INSTANCIAS (train/test): 2026 / 2572					
ACIERTOS: 621 / 24,14 %					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO REV	0,241	0,667	0,997	0,241	0,389
REVISTA	0,333	0,759	0,001	0,333	0,001
MATRIZ DE CONTINGENCIA					
620	1949				
2	1				

ErrorR($S \notin clase$ **) = 0,759 +- 0,017**

ErrorR($S \in clase$ **) = 0,667 +- 0,533**

ErrorR(S) = 0,759 +- 0,017

FIGURA IV.110: Evaluación DS1/DS2 Revistas H&L&U

MÉTODO: H&L&U					
Nº ATRIBUTOS: 1414					
Nº INSTANCIAS (train/test): 2572 / 2026					
ACIERTOS: 945 / 46,64%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NOREV	0,473	0,759	0,955	0,473	0,633
REVISTA	0,241	0,527	0,013	0,241	0,025
MATRIZ DE CONTINGENCIA					
931	1037				
44	14				

ErrorR($S \notin clase$ **) = 0,527 +- 0,022**

ErrorR($S \in clase$ **) = 0,759 +- 0,110**

ErrorR(S) = 0,534 +- 0,022

FIGURA IV.111: Evaluación DS2/DS1 Revistas H&L&U

MÉTODO: H&L&U					
Nº ATRIBUTOS: 1414					
Nº INSTANCIAS: 4598					
ACIERTOS: 1566 / 34,1%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO REV.	0,342	0,658	0,971	0,342	0,506
REVISTA	0,246	0,754	0,005	0,246	0,010
MATRIZ DE CONTINGENCIA					
1551	2986				
46	15				

ErrorR($S \notin clase$ **) = 0,658 +- 0,014**

ErrorR($S \in clase$ **) = 0,754 +- 0,108**

ErrorR(S) = 0,659 +- 0,014

FIGURA IV.112: Evaluación 2x2 Revistas H&L&U

IV.9.7 Salas Alternativas

MÉTODO: H&L&U					
Nº ATRIBUTOS: 1399					
Nº INSTANCIAS (train/test): 2026 / 2572					
ACIERTOS: 2374 / 92,30%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO SALA	0,93	0,476	0,992	0,93	0,96
SALA	0,524	0,07	0,11	0,524	0,182
MATRIZ DE CONTINGENCIA					
2352	178				
20	22				

ErrorR($S \notin clase$ **) = 0,070 +- 0,010**

ErrorR($S \in clase$ **) = 0,476 +- 0,151**

ErrorR(S) = 0,077 +- 0,010

FIGURA IV.113: Evaluación DS1/DS2 Salas Alternativas H&L&U

MÉTODO: H&L&U					
Nº ATRIBUTOS: 1399					
Nº INSTANCIAS (train/test): 2572 / 2026					
ACIERTOS: 1730 / 85,39%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO SALA	0,883	0,372	0,948	0,883	0,915
SALA	0,628	0,117	0,413	0,628	0,498
MATRIZ DE CONTINGENCIA					
1583	209				
87	147				

ErrorR($S \notin clase$ **) = 0,117 +- 0,015**

ErrorR($S \in clase$ **) = 0,372 +- 0,062**

ErrorR(S) = 0,146 +- 0,015

FIGURA IV.114: Evaluación DS2/DS1 Salas Alternativas H&L&U

MÉTODO: H&L&U					
Nº ATRIBUTOS: 1399					
Nº INSTANCIAS: 4598					
ACIERTOS: 4104 / 89,3%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO SALA	0,910	0,090	0,974	0,910	0,941
SALA	0,612	0,388	0,304	0,612	0,406
MATRIZ DE CONTINGENCIA					
3935	387				
107	169				

ErrorR($S \notin clase$ **) = 0,090 +- 0,009**

ErrorR($S \in clase$ **) = 0,612 +- 0,057**

ErrorR(S) = 0,107 +- 0,009

FIGURA IV.115: Evaluación 2x2 Salas Alternativas H&L&U

IV.9.8 Textos

MÉTODO: H&L&U					
Nº ATRIBUTOS: 3391					
Nº INSTANCIAS: 4617					
ACIERTOS: 4564 / 98,85%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO TEXTO	0,989	0,033	0,999	0,989	0,994
TEXTO	0,967	0,011	0,788	0,967	0,868
MATRIZ DE CONTINGENCIA					

4389	47
6	175

ErrorR($S \notin clase$ **) = 0,011 +- 0,003**

ErrorR($S \in clase$ **) = 0,033 +- 0,026**

ErrorR(S) = 0,011 +- 0,003

FIGURA IV.116: Evaluación Cross Textos H&L&U

IV.10 EVALUACIÓN BLOG SPECIFIC

MÉTODO: Blog Específico					
Nº ATRIBUTOS: 15					
Nº INSTANCIAS (train/test): 2138 / 2555					
ACIERTOS: 2541 / 99,45%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO BLOG	0,897	0,001	0,081	0,897	0,937
BLOG	0,999	0,103	0,995	0,999	0,997
MATRIZ DE CONTINGENCIA					
2437	2				
12	104				

ErrorR($S \notin clase$ **) = 0,001 +- 0,001**
ErrorR($S \in clase$ **) = 0,103 +- 0,055**
ErrorR(S) = 0,005 +- 0,003
FIGURA IV.117: Evaluación DS1/DS2 Blogs Specific

MÉTODO: Blog Específico					
Nº ATRIBUTOS: 15					
Nº INSTANCIAS (train/test): 2555 / 2138					
ACIERTOS: 2059 / 96,31%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO BLOG	0,954	0	1	0,954	0,976
BLOG	1	0,046	0,845	1	0,916
MATRIZ DE CONTINGENCIA					
1629	79				
0	430				

ErrorR($S \notin clase$ **) = 0,041 +- 0,009**
ErrorR($S \in clase$ **) = 0 + 0**
ErrorR(S) = 0,037 +- 0,008
FIGURA IV.118: Evaluación DS2/DS1 Blogs Specifics

MÉTODO: Blog Específico					
Nº ATRIBUTOS: 15					
Nº INSTANCIAS (train/test): 4693					
ACIERTOS: 4600 / 98%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO BLOG	0,980	0,020	0,997	0,980	0,989
BLOG	0,978	0,022	0,868	0,978	0,920
MATRIZ DE CONTINGENCIA					
4066	81				
12	534				

ErrorR($S \notin clase$ **) = 0,020 +- 0,004**
ErrorR($S \in clase$ **) = 0,022 +- 0,012**
ErrorR(S) = 0,020 +- 0,004
FIGURA IV.119: Evaluación 2x2 Blogs Specifics

IV.11 EVALUACIÓN ESPECIAL BLOGS SPECIFICS

MÉTODO: Blog Específico					
Nº ATRIBUTOS: 15					
Nº INSTANCIAS (train/test): 4693 / 9158					
ACIERTOS: 8446 / 92,2%					
Clase	**TP**	**FP**	**Precission**	**Recall**	**F**
NO BLOG	0,874	0,009	0,993	0,874	0,930
BLOG	0,991	0,126	0,846	0,991	0,913
MATRIZ DE CONTINGENCIA					
4726	679				
33	3720				

ErrorR($S \notin clase$ **) = 0,126 +- 0,009**

ErrorR($S \in clase$ **) = 0,009 +- 0,003**

ErrorR(S) = 0,078 +- 0,005

FIGURA IV.120: Evaluación extra Blogs Specific

ANEXO V: Urls A LOS RECURSOS DEL PROYECTO

Descarga del programa y los datos utilizados para las pruebas:

- Programa clasificador (fichero .exe autodescomprimible):

http://www.corex.es/pwp/frandzi/downloads/tesis/webclass.exe

- Datos utilizados (fichero .exe autodescomprimible

http://www.corex.es/pwp/frandzi/downloads/tests/files.exe

- Framework 2.0 de .Net

http://www.corex.es/documentum/downloads/dotnet/dotnetfx.exe