Marco de Trabajo para Indexación, Clasificación y Recopilación Automática de Documentos Digitales

Javier Caicedo Espinoza, Gonzalo Parra Chico, Xavier Ochoa Chehab
Facultad de Ingeniería Eléctrica y Computación
Escuela Superior Politécnica del Litoral (ESPOL)
Campus Gustavo Galindo, Km 30.5 vía Perimetral
Apartado 09-01-5863. Guayaquil, Ecuador
jcaicedo@cti.espol.edu.ec, gparra@cti.espol.edu.ec, xavier@cti.espol.edu.ec

Resumen

La creciente cantidad de textos disponibles en el Web y la imperiosa necesidad de información relevante por parte de los usuarios, hacen de la búsqueda de información en el Web un punto crítico en cualquier actividad investigativa. Para ayudar al usuario en ésta tarea se han desarrollado poderosos motores de búsqueda que organizan la información disponible en base a categorías, títulos o contenido. Estas herramientas basan su funcionamiento en crear grandes índices de documentos, usados para resolver las consultas de los usuarios. La mayor parte de los documentos obtenidos como resultados a dichas consultas no son relevantes para el usuario a pesar de cumplir los criterios de búsqueda especificados. Este trabajo presenta la integración de varias herramientas desarrolladas en el área de inteligencia artificial dentro de un marco de trabajo, el cual permitirá clasificar automáticamente documentos digitales, de acuerdo a áreas temáticas definidas; y, así mismo, pueda explorar en la Web y localizar publicaciones relacionadas con estos documentos. Adicionalmente, tendrá la capacidad de indexar los resultados obtenidos para poder resolver búsquedas locales.

Palabras Claves: *indexación de documentos, clasificación de documentos, exploración web, extracción de información*

Abstract

The growing amounts of documents available online, along with the user's need of relevant information, make the online search of documents a critical part of any research activity. In order to help the user in this task, a number of powerful search engines have been developed. These engines organize all the information using categories, the titles or the content. This tools work by creating large document indexes, and then use this indexes to answer the user's queries. Most of the documents that result from those queries are not relevant for the user, despite adjusting to the input search parameters. In this document we present the integration of several artificial intelligence tools into a framework, that will allow to automatically classify digital documents, according to predefined topics; and also, can explore the Web looking for new publications related to those documents. It will also have the ability to index these results, and answer local queries.

1. Introducción

En la actualidad, la Web se convierte en el recurso más valioso para el desarrollo de investigaciones, debido a la gran cantidad de información disponible [1][2]. Sin embargo, al no tener al Web una entidad central que la administre, y debido al continuo incremento de información, se hace cada vez más difícil la búsqueda de documentos relevantes para un investigador [2].

Los motores de búsqueda actuales intentan ayudar al investigador en su continua necesidad de información actualizada, pero, dado que basan su funcionamiento en palabras claves, los resultados obtenidos no siempre serán representativos, al no tomar en cuenta las áreas de interés del usuario. Por ejemplo, al buscar el término "palma" obtendremos resultados con distintos significados de acuerdo al contexto en que se la considere: la palma de la mano, el árbol, lugares geográficos, etc. Pero si enfocamos la búsqueda dentro de un contexto definido, sea éste lugares geográficos, obtendremos rápidamente el resultado esperado.

El desarrollo actual de nuevas herramientas en el área de inteligencia artificial, permite dirigir el enfoque de una búsqueda no sólo a palabras claves, sino a las áreas de interés del investigador, proporcionando así resultados más precisos y de mayor relevancia.

El presente documento describe el diseño de un marco de trabajo que tenga la capacidad de clasificar automáticamente documentos digitales, de acuerdo a áreas temáticas definidas; y, así mismo, pueda explorar en la Web y localizar publicaciones relacionadas con estos documentos (Sección 2). Adicionalmente, este marco tendrá la capacidad de indexar los resultados obtenidos para poder resolver búsquedas locales. Además, se presenta aspectos de implementación del marco y se realizan pruebas usando tópicos tomados del OpenDirectory [3] (Sección 3 y 4). Finalmente, se presentan las conclusiones y recomendaciones (Sección 5 y 6) y se especifican las referencias (Sección 7).

2. Diseño del Marco de Trabajo

Un marco de trabajo es una estructura de soporte, en la cual se base otro proyecto de software. Los marcos de trabajo son diseñados para facilitar el desarrollo de software [4]; por lo tanto, la clave de un buen marco de trabajo es que sea escalable y fácil de usar.

Dado que el marco de trabajo está orientado a resolver problemas sobre la exploración, clasificación e indexación de documentos digitales, hemos decidido separarlo en tres componentes, uno para cada tarea. De esta forma, cada componente tiene asignada una funcionalidad muy concreta, facilitando así la reutilización según los requerimientos de la aplicación cliente, como se aprecia en la Figura 1. Además, se facilitará la tarea de agregar o desarrollar nuevas herramientas o enfoques de solución a uno o varios de los componentes principales, independiente uno del otro; haciendo de éste un proceso transparente para los demás componentes del marco de trabajo. A continuación explicaremos de manera general los tres componentes principales.

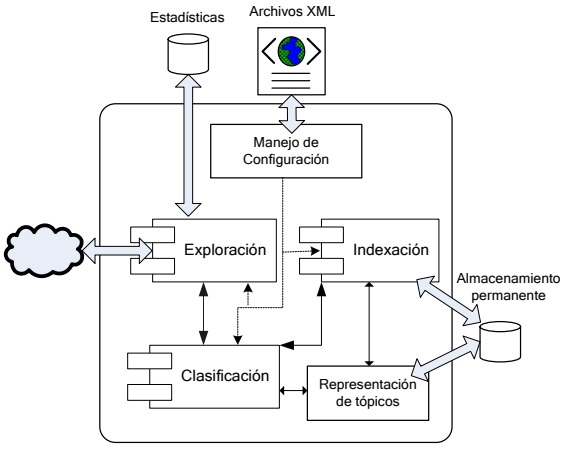

Figura 1. Arquitectura del Marco de Trabajo.

2.1. Exploración automática de páginas Web

Es el encargado de buscar páginas o documentos relacionados con un tópico específico a través del Web. El componente tendrá conexión a un buscador comercial el cual, a través de palabras claves, retornará las páginas Web iniciales referentes al tópico (semillas). Las páginas iniciales serán analizadas a fin de obtener vínculos a nuevos recursos similares; este proceso de análisis y extracción de nuevos vínculos se repite iterativamente hasta tener un máximo definido de recursos referentes a un mismo tópico. Los recursos obtenidos serán calificados de acuerdo a su relevancia para el tópico, por tal motivo, se deberá establecer una conexión con el componente de clasificación.

El componente de Exploración de páginas web estará formado por siete subcomponentes que procesarán los requerimientos de la aplicación cliente, como observamos en la Figura 2. A continuación describiremos cada uno de los subcomponentes.

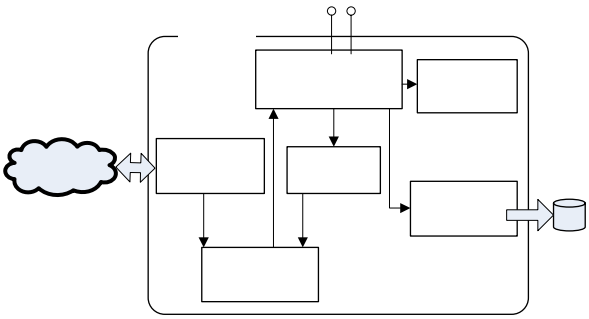

Figura 2. Diagrama del componente de exploración automática de páginas web.

- ***Obtención de semillas.*** Este subcomponente toma las palabras claves de un tópico que proporciona la aplicación cliente, y las utiliza para obtener los vínculos desde los cuales se iniciará el proceso de exploración. Para ello utiliza conexiones con buscadores comerciales (Google, por ejemplo).
- ***Extracción y evaluación de contenido.*** Es el encargado de extraer el contenido del recurso encontrado en Internet; de su conversión a texto plano y de la estimación de su importancia con respecto al tópico actual; esta evaluación es realizada por medio de una conexión con el componente de clasificación.
- ***Extracción de Vínculos.*** Una vez que se ha determinado la relevancia de un recurso, se procede a obtener vínculos (direcciones web) que apunten a recursos todavía no visitados por el agente explorador.
- ***Caché de recursos visitados.*** Utilizado para almacenar temporalmente los recursos que ya han sido analizados por el agente explorador, a fin de asegurarse de no volver a procesarlos. Los resultados que se retornan a la aplicación cliente al final del proceso son obtenidos de este almacenamiento temporal, para así evitar tener que descargarlos nuevamente del Internet. Los recursos en este caché se mantienen ordenados de acuerdo a su relevancia.

- ***Frontera de Vínculos.*** Es el encargado de almacenar los vínculos a los recursos aún no procesados por el componente. Estos vínculos se mantienen ordenados de acuerdo a su relevancia con respecto al tópico seleccionado por la aplicación cliente.
- ***Recolección de estadísticas.*** Es el encargado de almacenar estadísticas de la exploración, como páginas visitadas, documentos obtenidos, etc.

2.2. Clasificación de documentos.

Es el encargado de determinar si un documento es relevante para un tópico específico; para ello se utilizarán tópicos definidos con documentos positivos y negativos, los cuales servirán para el entrenamiento del algoritmo. El componente recibirá el contenido de un recurso a ser clasificado, el cual será preparado previo a su análisis. El algoritmo de clasificación puede ser adaptado al perfil del usuario, y así mejorar progresivamente su taza de aciertos.

El componente de Clasificación de documentos estará formado por cuatro subcomponentes que procesarán los requerimientos de la aplicación cliente, como observamos en la Figura 3. A continuación describiremos cada uno de los subcomponentes:

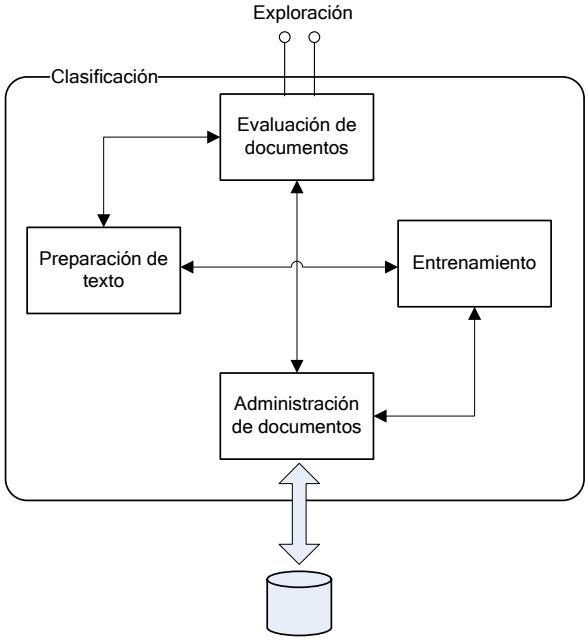

Figura 3. Diagrama del componente de clasificación de documentos.

- ***Preparación de texto.*** Es el subcomponente encargado de procesar el texto original de un documento previo a su análisis. Inicialmente, se eliminan las palabras más utilizadas en el idioma que no aportan contenido a una búsqueda (stopwords), y luego, se segmentan palabras reduciendo sus variantes a la forma léxica canónica (stemming).
- ***Entrenamiento.*** Es el encargado de instruir al algoritmo de clasificación para su correcto funcionamiento, basándose en ejemplos negativos y positivos con relación al tópico.
- ***Evaluación de documentos.*** Subcomponente que se encarga de la aplicación del algoritmo de clasificación sobre el documento; producto de este proceso se obtiene una calificación proporcional a la relevancia del recurso para el tópico escogido.
- ***Administración de documentos.*** Es el encargado de procesar el texto previamente preparado y representarlo de una manera entendible para una máquina, utilizando un modelo estructurado de representación.

2.3. Indexación de documentos

El componente de Indexación es el encargado de obtener metadatos descriptivos de un documento y luego agregarlo al índice creado para futuras búsquedas. El componente de Indexación de documentos estará formado por tres subcomponentes que procesarán los requerimientos de la aplicación cliente, como observamos en la Figura 4. A continuación describiremos cada uno de los subcomponentes:

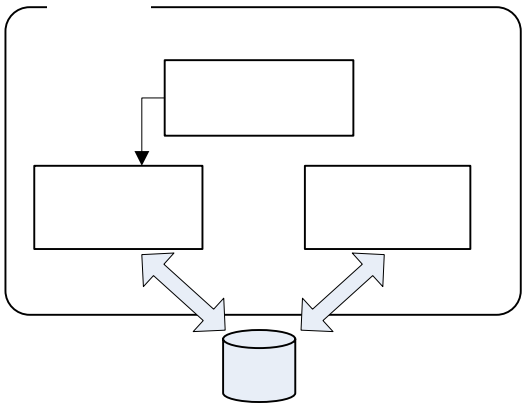

Figura 4. Diagrama del componente de indexación de documentos.

- ***Extracción de metadatos***. Es el subcomponente encargado de extraer datos descriptivos sobre un documento a ser almacenado. Los datos a ser extraídos son: palabras claves y resumen del texto.
- ***Administración de Repositorio Local.*** Es el encargado de manejar el repositorio local de documentos; esto es, la inserción y modificación de documentos dentro del índice generado.
- ***Búsqueda local.*** Encargado de manejar las búsquedas locales requeridas por una aplicación cliente, a través del índice creado para el repositorio local.

3. Implementación

La plataforma seleccionada como base para el desarrollo y ejecución de nuestro marco de trabajo es Java Platform. Esta conocida plataforma agrupa un conjunto de librerías estandarizadas que permiten el fácil desarrollo de aplicaciones utilizando el lenguaje Java; el usar esta plataforma para nuestro marco de trabajo implica que utilicen su funcionalidad también deberán ser implementadas usando este lenguaje de programación. Se ha escogido este lenguaje debido a su característica de ser portable, uno de los requerimientos principales del marco de trabajo; además por su gran favoritismo dentro de la comunidad de código abierto, de la cual obtenemos variadas librerías que permiten un fácil desarrollo del marco de trabajo.

En la programación de nuestro marco de trabajo, se decidió utilizar el paradigma de la Programación Orientada a Objetos, dado que este es el utilizado dentro de la plataforma Java. Esta plataforma provee una serie de librerías estándar, con las funciones más comunes a todos los proyectos de programación. Adicionalmente, y de acuerdo al concepto de reutilización de código, hemos utilizado una serie de librerías disponibles en el Lenguaje Java, bajo licencias de código abierto. Entre las principales podemos mencionar:

- ***Jakarta Commons Library***. Desarrollada como parte del proyecto Apache Jakarta , el conjunto de librerías Jakarta Commons se enfoca en proveer componentes reutilizables desarrollados bajo el lenguaje Java. Las clases utilizadas dentro del proyecto están organizadas en una serie de componentes, cada uno orientado a una tarea específica, por ejemplo: manejo de archivos de configuración o extensiones a las librerías nativas de Java.
- ***Log4j***. Librería que facilita el registro de sucesos (logging) y la depuración de errores dentro de programas. Esta librería permite habilitar o deshabilitar el registro de sucesos sin necesidad de alterar el código de la aplicación, a través de un archivo de configuración.
- ***HtmlParser***. Librería que implementa la interpretación y análisis de código HTML para el Lenguaje Java. En nuestro proyecto es utilizada por el agente explorador para extraer tanto el texto plano como los vínculos contenidos en un documento HTML.

4. Evaluación

Se ha decidido medir el rendimiento del marco de trabajo en base a su capacidad para obtener resultados relevantes para una búsqueda; para ello hemos definido un esquema de pruebas integrado por dos componentes: una evaluación por parte de usuarios reales y una evaluación automática en base a métricas (precisión y retentiva). Para propósitos de esta prueba se ha escogido el idioma inglés, ya que este idioma es el que presenta mayor cantidad de documentos en Internet.

4.1. Evaluación de usuarios reales

Este componente de pruebas se basa en la exploración de tres tópicos definidos y la obtención de resultados, los cuales serán calificados en base a su relevancia para cada tópico por tres usuarios escogidos al azar que tengan conocimiento en el área. Los tópicos escogidos están relacionados con el área de las Ciencias Informáticas y son: Computer Graphics, Machine Learning, Distributed Computing; estos tópicos serán construidos a partir del Open Directory [3]. El proceso de exploración será realizado por el algoritmo BestFirst, complementado con un clasificador Naive Bayes.

Se obtuvieron 30 resultados de la exploración por cada uno de los tópicos que se encuentran en el Anexo B. Tres usuarios con conocimientos en el área calificaron los 10 primeros resultados como relevantes o no relevantes. En las Figuras 5, 6 y 7, podemos apreciar la cantidad de resultados relevantes para los usuarios por cada uno de los tópicos explorados.

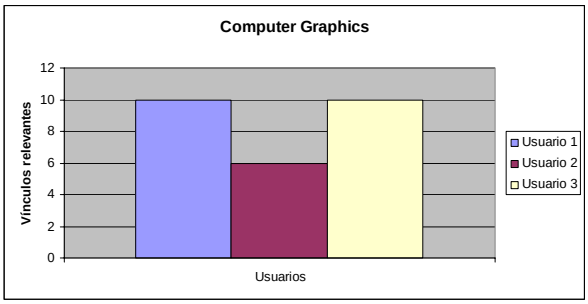

Figura 5. Comparación de la cantidad de vínculos relevantes para cada usuario encontrados durante el proceso de exploración del tópico Computer Graphics.

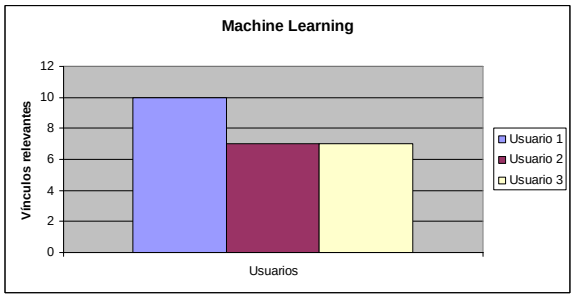

Figura 6. Comparación de la cantidad de vínculos relevantes para cada usuario encontrados durante el proceso de exploración del tópico Machine Learning.

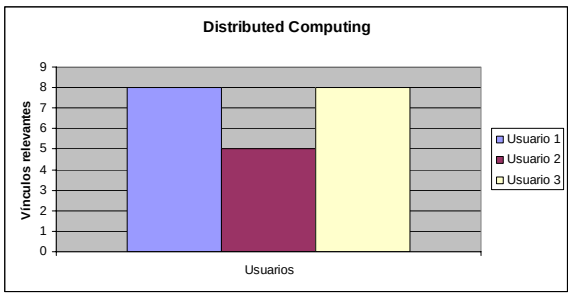

Figura 7. Comparación de la cantidad de vínculos relevantes para cada usuario encontrados durante el proceso de exploración del tópico Distributed Computing.

Como se puede apreciar en las figuras, el promedio de vínculos relevantes para los usuarios tópico es de 8, esto quiere decir que de los primeros 10 documentos, 8 son relevantes para el usuario y a medida que el usuario retroalimente al marco de trabajo con su elección de documentos relevantes, la efectividad puede incrementarse. En general, se obtuvieron mejores resultados con el tópico Computer Graphics, lo cual puede deberse a una mejor calidad de los documentos de ejemplo utilizados para entrenar al clasificador. Así mismo podemos observar que los resultados de la evaluación del Usuario 2 son inferiores a los de los demás; una razón de esta diferencia puede ser un menor dominio del lenguaje en el que se encuentran los resultados, en este caso el inglés.

4.2. Evaluación automática

Para este componente del esquema de pruebas, la exploración se la realizará en base a un tópico definido, y nos permitirá obtener la cuantificación de las métricas de precisión y retentiva de los resultados de la exploración. Al ser el Internet una gran fuente de recursos, es imposible conocer exactamente el conjunto de documentos relevantes para una búsqueda; por tal motivo definiremos un conjunto de documentos objetivo (T), el cual nos permitirá aproximar el conjunto desconocido de todos los documentos relevantes(R), como se observa en la Figura 8. De esta manera, se pueden aproximar las métricas de precisión y retentiva, al momento de ejecutar una exploración.

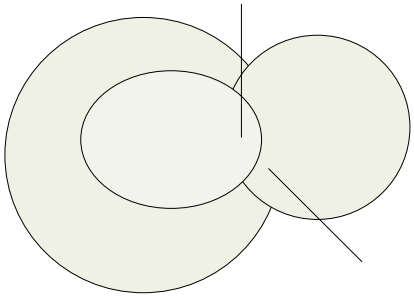

Figura 8. Diagrama del componente de indexación de documentos [5].

El proceso de exploración será realizado por dos algoritmos. El primero es el BestFirst, algoritmo elegido para la implementación dentro del marco de trabajo; el cual será complementado con un clasificador Naive Bayes entrenado con los ejemplos del tópico a ser explorado. Como segundo algoritmo, se utilizará el BreadthFirst, el cual servirá como punto de comparación de la eficiencia del algoritmo propuesto como solución.

A fin de que los resultados tengan validez, es necesario asegurarnos que el conjunto de documentos objetivos sean alcanzables, partiendo de las direcciones iniciales (semillas) escogidas para la exploración. Para ello, hemos definido un proceso de generación de semillas, que se muestra en la Figura 9. El proceso consiste en obtener las direcciones que apuntan a un documento objetivo y tomar una muestra aleatoria de este nuevo conjunto. Este proceso se repite un número N de veces definido previamente, el cual representa el número de saltos desde una semilla a un documento objetivo. A medida que aumenta N la dificultad del proceso se incrementa considerablemente.

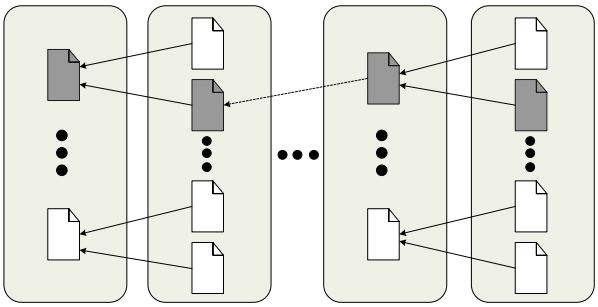

Figura 9. Proceso para extracción de semillas en el segundo esquema de pruebas. A mayor N, mayor dificultad en la tarea [5].

Para este proceso de pruebas hemos escogido un valor para N de 2. Además, las pruebas se correrán para cinco conjuntos de semillas diferentes obtenidos a partir de un conjunto de documentos objetivo del tópico Computer Graphics. Estos conjuntos de semillas serán obtenidos con ayuda de la interfaz para desarrolladores de Yahoo.

Para la evaluación, se realizó una exploración de un máximo de 1000 páginas a partir de cinco grupos de semillas del tópico Computer Graphics, y como se describió en la sección previa, se requiere encontrar la mayor cantidad de documentos objetivo durante la exploración. Para esta exploración se definieron 46 documentos objetivos y fue realizada con los algoritmos Best First y Breadth First. A medida que el explorador avanza, se calculan las métricas de precisión y retentiva, lo cual nos permite efectuar un análisis en el tiempo del proceso de exploración. En las Figuras 10 y 11 se observan las mediciones de precisión y retentiva obtenidas por cada uno de los algoritmos, promediadas para los 5 grupos de semillas.

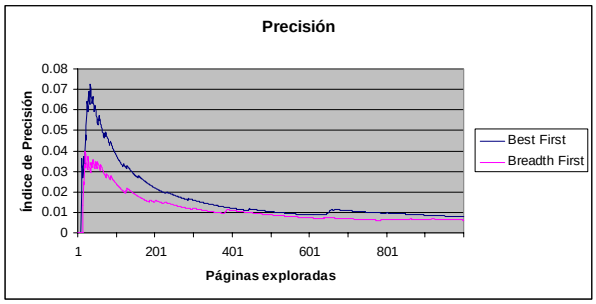

Figura 10. Comparación de la precisión promedio obtenida por los algoritmos Best First y Breadth First en cinco exploraciones.

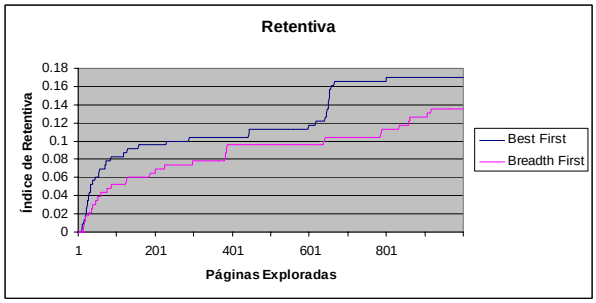

Figura 11. Comparación de la retentiva promedio obtenida por los algoritmos Best First y Breadth First en cinco exploraciones.

En la Figura 10 podemos apreciar que en ambos algoritmos la precisión muestra una tendencia a la baja, algo que podía esperarse, dada la dificultad de la tarea. Además, el algoritmo Best First guiado con un clasificador (solución propuesta por este trabajo) se desempeña de mejor manera que el algoritmo a ciegas Breadth First según el cálculo de las dos métricas; es decir, encuentra más documentos objetivos que su contraparte. Vemos, que al acercarse a las 1000 páginas exploradas, el algoritmo Best First presenta una precisión ligeramente inferior a 0.01, y una relevacia cercana a 0.18. Estos valores son cercanos a los obtenidos en investigaciones anteriores [5], donde se usó un esquema de pruebas similar.

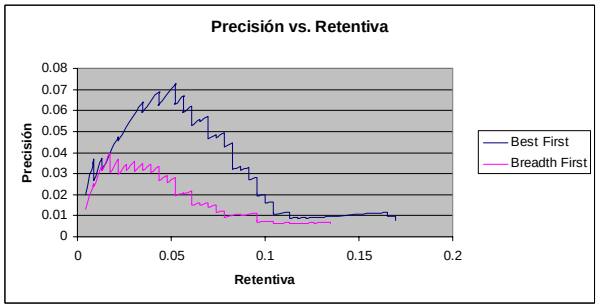

Figura 12. Comparación entre la retentiva y la precisión por el algoritmo Best First.

En la Figura 12 observamos la comparación entre los algoritmos Best First y Breadth First. Como se puede observar dado un valor de retentiva, se obtiene una mejor precisión con el algoritmo Best First. A medida que la retentiva es mayor a 0.05 la precisión va disminuyendo progresivamente, explicando así la relación inversa entre estas dos métricas..

5. Conclusiones

Luego de finalizada la implementación del marco de trabajo como solución al problema descrito, se concluyó que:

- La existencia de una considerable cantidad de trabajos previos en las áreas de la inteligencia artificial relacionadas con este trabajo, facilitó el análisis de las soluciones a incluir en el marco de trabajo.
- Luego de realizar un análisis de las técnicas disponibles para la búsqueda de documentos digitales en línea, se decidió escoger la estrategia de "primero el mejor", por ser la más general, y la que mejor se adapta a los requerimientos del marco de trabajo.
- Para la tarea de clasificación de documentos digitales, se escogió el algoritmo "Naive Bayes", ya que el tiempo requerido para su utilización es mucho menor comparado a las otras opciones existentes; y, adicionalmente, es el que mejor soporta el aprendizaje activo, permitiendo adaptar el modelo de clasificación a los requerimientos específicos del usuario.
- Se decidió incorporar la técnica de índices invertidos como solución a la indexación de documentos, ya que es la que más se ajusta a los requerimientos establecidos en el marco de trabajo, al tener un desempeño adecuado en un gran número de situaciones.
- La capacidad que tiene el marco de trabajo de poder añadir nuevas funcionalidades nos permitió poder hacer modificaciones e inclusive desarrollar diferentes enfoques de solución para realizar pruebas de rendimiento.
- Una herramienta orientada a la exploración del Internet, como este marco de trabajo, no tiene sentido si se lo aplica en búsquedas generales, donde un buscador comercial es mucho más eficiente. En cambio, el utilizarlo en un campo específico puede dar resultados mucho más alentadores, como los obtenidos al realizar las pruebas.
- Es importante verificar la calidad de los documentos de ejemplo utilizados para entrenar al clasificador bayesiano, ya que la eficiencia del clasificador afecta directamente al proceso de exploración.
- Los parámetros de la exploración pueden afectar el rendimiento del algoritmo a utilizar; éste fue el caso de Best First, donde si se define un tamaño de frontera muy pequeño podemos perder recursos valiosos que no serán explorados.

6. Recomendaciones

Consideramos las siguientes tareas para realizarse en un futuro próximo:

- El marco de trabajo es flexible, por tanto es recomendable estar atento a nuevos enfoques y proyectos en el área, a fin de implementar nuevas características que permitan actualizar y mejorar el desempeño del marco de trabajo.
- Dada la naturaleza del Internet, existe mucha redundancia de contenido, esto es, existen páginas web con contenidos iguales o muy similares, pero con diferente dirección. Esta característica es perjudicial para la exploración, porque tiende a encontrar resultados muy relevantes pero repetidos. Por ello es de considerar a futuro la búsqueda de otro mecanismo independiente del URL o en su defecto una variante para tener un identificador único por recurso.
- Soluciones implementadas utilizando este marco de trabajo, pueden ser integradas en sistemas de manejo de contenidos, donde un agente explorador puede obtener nuevos recursos automáticamente, para que sean revisados por usuarios reales y agregados a una biblioteca interna.

7. Referencias

[1] IBLNEWS, "Internet, primera fuente de información europea antes que el papel". Consultado Octubre 2006. Disponible en http://iblnews.com/story.php?id=18637.

[2]Lawrence S., Giles C. L., "Accessibility of information on the web", Nature, VOL400, Julio 1999, pp. 107-109.

[3]Open Directory Project. Consultado Octubre 2006. Disponible en http://www.dmoz.org/

[4] Clifton M., "What Is a Framework?", The Code Project. Consultado Octubre 2006. Disponible en http://www.codeproject.com/gen/design/WhatIsAFramework.asp.

[5] Menczer F., Pant G., Srinivasan P., "A General Evaluation Framework for Topical Crawlers", Information Retrieval 8 (3), 2005, pp. 417-447.