Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Computación

Un prototipo de buscador vertical sobre cine documental asistido por aprendizaje supervisado

Tesis presentada para optar al título de
Licenciado en Ciencias de la Computación

Iván Matías Badgen

Director: Dr. José Castaño

Buenos Aires, 2015

UN PROTOTIPO DE BUSCADOR VERTICAL SOBRE CINE DOCUMENTAL ASISTIDO POR APRENDIZAJE SUPERVISADO

En este trabajo se estudian y aplican distintas técnicas de *web mining* e *information retrieval* con el objetivo de explorar el espacio de sitios web y desarrollar un prototipo de buscador sobre cine, particularmente bajo la categoría de documentales. Se comenzó partiendo de algunas semillas consideradas de interés y luego se amplió a resultados de algunos buscadores tradicionales. La idea no fue sólo quedarse con ellos, sino intentar descubrir nuevos sitios que se pudieran clasificar también dentro del interés planteado. Por otra parte, utilizando *crawling* e indexando los resultados, se estudió el espacio obtenido en términos de grafos, para determinar qué sitios podrían ser más relevantes que otros dentro del dominio. En este caso, no necesariamente relevantes en cuanto a contenido, pero sí como potenciales semillas para encontrar otros sitios relacionados. El trabajo en buscadores verticales es usualmente complementado con técnicas de aprendizaje automático para mejorar tanto la búsqueda como la presentación de resultados. En el caso de este trabajo, se utilizaron algoritmos de clasificación para el descubrimiento de nuevas páginas relevantes y algoritmos de *clustering* para el análisis de los resultados obtenidos. Como resultado, se implementó un prototipo de buscador para el cine documental cuyo contenido esté restringido a documentales del cine hispano-americano.

Palabras claves: Web Mining, Information Retrieval, Classification, Clustering, Search Engines.

AGRADECIMIENTOS

A mi papá Javier, que lamentablemente no pudo verme llegar hasta acá, pero que desde chiquito me incentivó a estudiar y me apoyó siempre en todas mis decisiones.

A mi mamá Irene, que me inculcó el espíritu científico y emprendedor, y más allá de nuestros mutuos rayes, supo soportarme y aconsejarme durante todos estos años.

A mi hermano Zequi, que me convenció de estudiar esta carrera en esta facultad, y aún hoy a la distancia me sigue ayudando mucho y motivando para superar estas etapas.

A mi hermana Naty, que se interesó siempre en mi vida, personal y académica, y me supo aconsejar bien cuando lo necesitaba.

Al resto de mi familia, Tíos, Tías, Babis, Primos, etc. por todos los momentos compartidos, reuniones, fiestas, opiniones y consejos.

A mis amigos y compañeros de la facu: Bender, Tincher, Browar, Rinem, Vale, Leo, Manolo, Gabi, Kuja, Axel, Nacho, Manu, Juampi, Peter, Lucho, Javo, ¡espero no olvidarme de nadie! y especialmente a mis compañeros de grupos de TP, sin los cuales aprobar las materias habría sido definitivamente más tedioso y aburrido: Uri, Ale, Pablo y Soifer.

A mis compañeros ayudantes, JTPs y profesores de Algo 2 y Algo 3 que me formaron como docente y ayudaron mucho a superar la carrera a lo largo de los cuatrimestres.

A mis amigos y compañeros de trabajo a lo largo de todos estos años que tuvieron que bancarme en los días más locos entre entregas de TP, parciales, finales, etc: Nacho, Facu, Tincho, Mago, Tomer, Pablo, Ana, Pato, Uru, Fabi, Juampi, Tomi, Peter, Dani, René, Mauro, Lu S, Lu P, Emma, Julia, Tulio, Jony, Ari, Javi, Mauro, Richard, Joel, David y Gabi.

A mis amigos de la infancia, Mati y Nano, gracias por estar todos estos años!

A mis amigos del colegio y de la vida, mis compañeros en viajes, salidas, las buenas, las malas, las todas: Tomi, Choco, Coco, Mario, Frenk, Santi, Fede, Besio y Maxi.

A mi novia Lau, que me acompañó y bancó incondicionalmente este último año con todas las idas y vueltas que conlleva terminar una carrera y escribir esta Tesis.

Al Departamento de Computación y todos sus integrantes, por darme una excelente educación, ayudándome a crecer tanto personal, como profesionalmente, como alumno y como docente.

A José, por dirigir este trabajo y por sus consejos a lo largo de todo este trayecto.

A todo el que pude haberme olvidado de mencionar, pero que sabe que no es intencional y aún así fue parte de este camino.

A todos, ¡muchas gracias!

Índice general

1. INTRODUCCIÓN

Este proyecto surge como un pedido por parte de la *Maestría en periodismo documental*[1] de la Universidad de Tres de Febrero. La problemática con la que se encontraron fue la dificultad de encontrar resultados precisos y rápidos en internet, dentro del dominio del cine documental hispanoamericano.

Se presentó el desafío de explorar el espacio web con la perspectiva de desarrollar una herramienta que permitiera mejorar la experiencia de estas personas al buscar información sobre películas, festivales, revistas, etc. dentro de esta temática. En principio, podría plantearse el uso de buscadores web tradicionales, aunque surge el problema de que los buscadores genéricos suelen basar fuertemente la relevancia de un sitio en función de su conexión con otros sitios relevantes en la web en general. En este caso, las páginas que se buscan suelen ser blogs de aficionados, páginas amateur, sitios poco conocidos, que terminan apareciendo luego de navegar muchas páginas de resultados.

El objetivo final de este proyecto fue la exploración del espacio y una propuesta de construcción de un buscador web vertical o por tópicos, enfocado en sitios y páginas web donde se hable de **cine documental** en español, para lo cual se utilizaron y desarrollaron distintas herramientas desde la etapa de *crawling* e *indexing* hasta la búsqueda por parte de un usuario final.

Un buscador vertical, a diferencia de un *general search engine* puede enfocarse en un segmento específico de contenido online o abarcar un subconjunto o subdominio particular de la web. Algunos ejemplos de este tipo de buscador son:

- IceRocket - `http://www.icerocket.com/`, un buscador especializado en tres fuentes de contenido específicas: Twitter, Facebook y Blogs.
- Pipl - `https://pipl.com`, un buscador de personas basado en la indexación de las redes sociales.
- Indeed - `http://www.indeed.com`, un buscador de trabajo indexando diversos sitios de búsquedas laborales.
- TinEye - `http://tineye.com/`, un buscador inverso de imágenes, capaz de encontrar el sitio del cual provino una provista por el usuario.

En el subdominio del cine documental, surgen algunas preguntas que se estudiaron y se intentan responder en este trabajo:

- Sitios poco conocidos... ¿Cómo se los puede encontrar?
- ¿Cómo se los identifican? ¿Son de interés o no?
- ¿Qué significa que una página "hable sobre documentales"? ¿Existe un criterio objetivo para esta tarea?

[1] http://untref.edu.ar/posgrados/maestria-en-periodismo-documental/

1.1. Definición del problema y trabajo previo

Supongamos que se quiere buscar la ficha o un *trailer* de una película documental sobre el peronismo en la Argentina. Si se introduce la búsqueda "peronismo en la argentina" en un buscador tradicional, se encuentran muchísimos resultados hablando sobre historia, filosofía, economía y otros temas que pueden girar en torno al peronismo. Si bien serían relevantes para aprender sobre el tema, no lo son a la hora de encontrar rápidamente documentales o autores de ellos.

Los requerimientos para este trabajo fueron los de desarrollar un prototipo de herramienta que permitiera encontrar sitios web que trataran el tema de cine documental hispanoamericano. Para ello, fueron provistos por la UNTREF algunos sitios o portales de ejemplo como el de la revista electrónica de cine documental argentino[2] o el de la documentalista colombiana Marta Rodriguez[3].

Se desarrolló un prototipo de buscador web que muestra resultados de diferentes fuentes, como sitios provistos manualmente, sitios encontrados mediante Google o Bing, e incluso con la capacidad de descubrir páginas nuevas y detectar si tratan sobre cine documental o no. Esta detección se logra gracias a una base de datos de páginas ya clasificadas previamente como de documentales y una funcionalidad del buscador que permite decirle si el resultado que uno ve es o no de interés.

El tema abordado es complejo y entre las dificultades encontradas se encuentra la subjetividad que tiene para cada persona el interés que se tiene en los resultados de una búsqueda. Tal es el caso que quizás una página con la ficha de un documental puede no ser de interés para alguien que está buscando ver un *trailer*, mientras que sí puede ser de interés para quien busca información sobre la película. Sin embargo, esa página sí habla sobre cine documental, ¿cómo se las distingue?

El resultado principal fue la construcción de un buscador vertical en el tópico de documentales. Alrededor de ello, surgen otros subproblemas que se describen a lo largo de este trabajo, desde arquitectura, aprendizaje automático, recuperación de la información hasta algunos no tan técnicos como qué contenido es necesario almacenar para mostrar al usuario y con qué frecuencia actualizarlo.

El tema de buscadores verticales ha sido muy estudiado, dando lugar a distintos enfoques e investigaciones al respecto. Como se describe más adelante, el hecho de filtrar el dominio de búsqueda puede ser abordado a lo largo de distintas etapas. Un ejemplo puede encontrarse en [8] y [9] donde se utilizan herramientas de clasificación de texto para el problema particular de entender el dominio deseado, tanto desde el contenido como desde el análisis de la estructura de los sitios. En [10] y [11] se aplica un análisis semántico más avanzado utilizando técnicas de álgebra lineal para mejorar el foco del *crawling*.

En todos estos casos, el objetivo es reducir la colección de documentos en la cual se busca, de toda la web a un conjunto de documentos (o sitios) ya procesado y en el cual se tiene mayor certeza de encontrar un resultado acorde al dominio buscado.

[2] http://revista.cinedocumental.com.ar/
[3] http://www.martarodriguez.org/martarodriguez.org/Inicio.html

2. MARCO TEÓRICO Y TÉCNICAS UTILIZADAS

En este capítulo se introducen las técnicas utilizadas para resolver el problema, detallando la base teórica necesaria para comprender los experimentos realizados y la construcción del prototipo. El contenido presentado es una adaptación extraída de los capítulos 1, 6, 7 y 8 de [1] y los capítulos 1, 6, 13, 16, 20 y 21 de [2] y pueden ser consultados para ampliar sobre algún tema en particular.

2.1. Contexto

La *World Wide Web* es el más grande, accesible y conocido repositorio de hipertexto actual. Entendemos por hipertexto a documentos de texto que se muestran en una computadora, que contienen hipervínculos a otros documentos o sitios, imágenes, videos, metadatos, etc. Con su incremental crecimiento en tamaño y diversidad, la *Web* se convirtió en un repositorio de conocimiento de gran valor, al cual hoy en día cualquier persona con acceso a internet puede acceder y consultar.

2.2. Crawling and Indexing

Para poder procesar y buscar en esta gran cantidad de texto e hipertexto, necesitamos primero poder descargar cientos o miles de páginas por segundo para luego crear un índice con la información. Se conoce con el nombre de **Web crawlers**, **spiders** o **robots** a programas que automáticamente descargan páginas web, almacenan el contenido y son capaces de seguir hipervínculos para descubrir páginas nuevas.

En su forma más básica, un crawler comienza su tarea a partir de un conjunto inicial de páginas conocido como semillas o *seed URLs*. En el proceso, se mantiene un conjunto de páginas aún no visitas llamada *frontier* o frontera, inicializada con las semillas. Iterativamente se lee el contenido de cada página, se extraen los hipervínculos salientes de cada una y se agregan al conjunto actual de páginas a recorrer. De esta forma, en cada paso puede ir creciendo indeterminadamente el conjunto frontera, o eventualmente podría llegar a ser vacío. Se debe elegir un criterio de parada, que podría ser luego de una cierta cantidad de páginas recorridas, hasta cierto nivel de profundidad al seguir hipervínculos, o que el conjunto *frontier* resulte vacío.

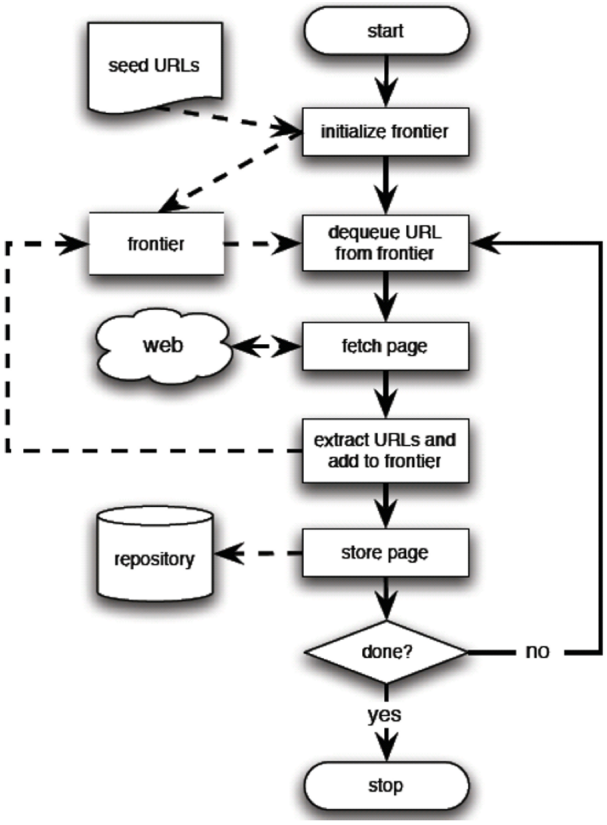

Fig. 2.1: Algoritmo general de crawling secuencial. Tomado de [1].

En la figura 2.1 se muestra un ejemplo ilustrativo de cómo se implementaría este proceso en forma secuencial. En la práctica, se implementan algunas técnicas diferentes para poder alcanzar un mejor objetivo de escalabilidad. En particular, cada URL desencolada, obtenida, procesada y almacenada puede ser procesada en paralelo (dentro de la misma computadora) o incluso en forma distribuida (utilizando varias computadoras en simultáneo).

Un *crawler* implementa, en esencia, un algoritmo de búsqueda o recorrido en un grafo. La Web es comúnmente modelada como un grafo dirigido, donde las páginas son los nodos o vértices y los hipervínculos sus aristas (que pueden ser entrantes o salientes). Si observamos el algoritmo presentado en la figura 2.1, puede verse una semejanza muy clara con algoritmos comúnmente aplicados a grafos como **BFS** (Breadth-first search), donde *frontier* representa la cola utilizada para su implementación. El algoritmo del crawler debe especificar en qué orden se visitan las páginas de la frontera, dando distinto comportamiento a la forma en que se explora el grafo. Entre los más comunes se encuentran Breadth-first, implementado con una cola FIFO y los preferenciales que utilizan una cola de prioridad, con algún criterio de comparación en la calidad de una página [5] [1].

Como se dijo anteriormente, para procesar y descargar contenido de sitios web es fundamental la utilización de una herramienta de *crawling*. En el caso de este trabajo, se aplicó al recorrido de sitios, variando algunos de los parámetros mencionados como el nivel de profundidad a seguir según se deseara obtener diferentes resultados.

2.3. Recuperación de la Información y Búsqueda en la Web

La búsqueda web tiene sus raíces en la recuperación de la información, *Information Retrieval* o **IR**, un campo de estudio que ayuda al usuario a encontrar la información necesitada entre una larga colección de documentos de texto. En general, se asume que la unidad básica de información es un documento, y una larga colección de documentos forma una base de datos de textos. En la Web, un documento está representado por una página web.

Este es un tema fundamental para este trabajo, ya que ayuda a comprender la representación computacional que se le da a la unidad elemental de trabajo, una página web. Esta representación es la que se utiliza, además de la recuperación o búsqueda, para la aplicación de otras técnicas como clasificación o agrupamiento.

Recuperar información significa encontrar un conjunto de documentos que sea relevante para la consulta del usuario, generalmente expresada como palabras clave o *keywords*. Esta tarea difiere de la recuperación en una base de datos, donde se cuenta con datos estructurados y almacenados en forma de tablas, mientras que la información en los textos es desestructurada.

Al extender este problema a páginas web aparece otro nivel de complejidad, ya que se agregan hipervínculos y sus textos asociados, los cuales juegan un rol importante a la hora de categorizar un sitio o de asignar la relevancia a un resultado. Además, se incorporan otros campos como los metadatos de una página, el título, el cuerpo, entre otros, lo cual hace pensar a estas páginas como datos semi-estructurados.

Dada una colección de documentos D, sea $V = \{t_1, t_2, ..., t_{|V|}\}$ el conjunto de términos distintos en la colección, donde t_i es un término. El conjunto V suele llamarse el vocabulario de la colección y $|V|$ su tamaño, la cantidad de términos existentes en V. A cada término t_i se le suele asociar un peso $w_{ij} > 0$ en un documento $d_j \in D$. Para un término que no aparece en el documento d_j, $w_{ij} = 0$. Cada documento d_j entonces es representado por un vector de términos, conocido como **term vector** en inglés.

$$d_j = (w_{1j}, w_{2j}, ..., w_{|V|j}),$$

donde cada peso w_{ij} corresponde al término $t_i \in V$ y describe el nivel de importancia de t_i en el documento d_j.

Con esta representación vectorial, una colección de documentos es simplemente representado como una matriz $W \in \mathbb{R}^{|V| \times |D|}$, donde una dimensión representa los documentos d_j y la otra los términos t_i. Según el modelo utilizado, w_{ij} será computado de diferente forma.

Modelo Booleano

Es el modelo más simple y usa nociones del álgebra de Boole. En este caso, el valor w_{ij} de la matriz toma los valores 0 ó 1, dependiendo de la ausencia o presencia respectivamente del término t_i en el documento d_j.

$$w_{ij} = \left\{ \begin{array}{ll} 1 & \text{Si } t_i \in d_j \\ 0 & \text{Si no} \end{array} \right\}.$$

Dado un conjunto de términos a buscar o *keywords*, la recuperación se basa en encontrar los documentos que los contengan o no, dependiendo de operadores de inclusión o exclusión

asociados a esos términos. Es decir, se aplica un filtro a cada documento, analizando los valores de las columnas de la matriz correspondientes a cada *keyword*. Si se quiere la presencia de un término, el valor buscado será 1, y 0 si se busca excluirlo.

Cabe destacar que según este modelo, un documento es relevante o irrelevante en base a la presencia o ausencia de un término, pero no tiene en cuenta si una palabra se encuentra repetidas veces, dando menores matices posibles a los resultados.

Vector Space Model

En este modelo, cada documento se representa con un vector de pesos asociados a los términos (ya no sólo 0 ó 1), donde cada peso se computa de distintas maneras, como la frecuencia absoluta de un término, o la frecuencia relativa de un término a todos los documentos.

Term Frequency (TF)

En este esquema, el peso w_{ij} de un término t_i en el documento d_j es la cantidad de veces que aparece t_i en d_j, denotado f_{ij}. Existen variantes, como normalizar esa frecuencia por ejemplo con el máximo f_{ij}. Una contra de este esquema es que no discrimina términos que aparecen muy seguido en todos los documentos, como podría ser un artículo.

Term Frequency - Inverse Document Frequency (TF-IDF)

Es el esquema de pesos más usado en la práctica, y si bien hay muchas variaciones, se introduce la versión más básica. La intuición detrás de este esquema es que si un término aparece en una gran cantidad de documentos, es probable que no sea importante o discriminativo para una posible búsqueda.

$$tf_{ij} = \begin{cases} \frac{f_{ij}}{max\{f_{1j},\ldots,f_{|V|j}\}} & \text{Si } t_i \in d_j \\ 0 & \text{Si no} \end{cases}$$

$$idf_i = log\frac{|D|}{|\{d \in D \mid t_i \in d\}|}$$

$$w_{ij} = tf_{ij} \times idf_i$$

A diferencia del modelo booleano, la decisión de si un documento es relevante en una búsqueda ya no va a ser dicotómica. En cambio, se suelen ordenar los resultados en base a su grado de relevancia. Para lograr esto, se suele considerar a las palabras clave como un nuevo documento **q** por query, representarlo en el espacio de la matriz de documentos W y calcular una medida de similitud entre q y cada d_j. La medida más usual es la determinada por el coseno entre el ángulo formado entre los vectores. A diferencia de la distancia euclídea, la distancia coseno no es tan sensible a diferencias en una sola de las dimensiones (donde por dimensiones entendemos el peso asignado a un término).

$$cosdist(d_j, q) = \frac{\langle d_j \cdot q \rangle}{||d_j|| \times ||q||}$$

2.3.1. Técnicas complementarias

Ignorando términos comunes: *Stop words*

En algunos casos, palabras extremadamente comunes ayudan muy poco a recuperar documentos relevantes para el usuario y necesitan ser excluídas enteramente del vocabulario. A estos términos se los llama ***stop words***. Ignorar este tipo de términos reduce significativamente el número de datos que un sistema tiene que almacenar.

Si bien una búsqueda con artículos o preposiciones como "el" o "por" no parece muy útil, hay algunos casos donde puede resultar provechoso. Por ejemplo, la búsqueda "presidente de La Argentina" es bastante más específica que la conjunción entre los términos "presidente" y "Argentina". Otro más conocido en el idioma inglés es el de canciones enteramente constituidos por palabras que comúnmente consideraríamos *stop words* como *To be or not to be, Let It Be, I don't want to be,....*

Normalización (clases de equivalencia de términos)

Como se mencionó, la forma más fácil de recuperar documentos que sean relevantes a la *query* es si tenemos términos en los documentos que se corresponden con las *keywords* buscadas. Sin embargo, hay muchos casos donde dos cadenas de caracteres no son exactamente la misma, pero quisiéramos que fueran equivalentes. Tomemos como ejemplo mayúsculas/minúsculas, tildes faltantes, siglas con o sin los puntos (RR.HH. ≡ RRHH, AFIP ≡ A.F.I.P.).

La normalización de *tokens* es el proceso de asignar formas canónicas a estas clases, a pesar de las diferencias superficiales en las cadenas de caracteres. Se utilizan diferentes formas de normalización en el almacenamiento/recuperación que podrían ser implícitas o explícitas. Podríamos buscar la forma canónica de todos los términos de cada documento y almacenarla, para luego aplicar la misma regla a la búsqueda. También podríamos almacenar los términos originales y buscar las equivalencias en forma dinámica. Es bastante claro que la primera forma es más eficiente, aunque estamos perdiendo información de los textos originales que podrían impactar semánticamente.

Lemmatization o lematización

Además de los ejemplos mencionados como remoción de tildes, puntos o unificación de mayúsculas/minúsculas, sería conveniente unificar formas verbales como **organizar, organiza, organizando** que se utilizan en los lenguajes por cuestiones gramaticales. Lo mismo ocurre con sustantivos, adjetivos y sustantivos abstractos como **democracia, democrático, democratización**. Esta técnica se conoce como *lemmatization* o lematización. Dependiendo del lenguaje utilizado, el algoritmo será muy diferente y usará diferente *corpus* o diccionario, por lo cual para aplicar estas técnicas es muy importante tener en cuenta el idioma en el que fueron escritos los documentos.

En el caso de este trabajo, todos los documentos con los que se trabajó fueron escritos en español, para el cual las herramientas existentes no son tan buenas como para el inglés.

2.4. Agrupamiento y Clasificación

En la web podemos encontrar directorios por tópicos construídos con esfuerzo humano (e.g. Yahoo! y Open Directory[1]). Lo que intentan hacer es agrupar sitios web según diversos temas como arte, juegos, negocios, recreación, etc. La pregunta que surge es si este tipo de sitios pueden ser construidos inmediatamente dado un corpus de páginas web, como las que se obtienen luego de un proceso de *crawling*.

En el caso de este trabajo, se buscó estudiar grupos que se formaran entre distintos sitios sobre documentales. Teniendo en cuenta la gran cantidad de documentos que pueden ser obtenidos como resultado de un proceso de *crawling*, una forma exploratoria de analizarlos es la de buscar *clusters* que se formen entre esas páginas, lo cual puede resultar en grupos por categorías de documentales, páginas que traten sobre películas similares o incluso distintos tipos de páginas (fichas, videos, biografías) dependiendo de qué información se utilice del hipertexto.

El problema de agrupamiento o *clustering* de textos consiste en encontrar grupos o *clusters* dentro de un conjunto de documentos que sean lo más similares posible dentro de un grupo, y lo menos similares posible entre grupos. Al igual que cuando se habló del *vector space model*, una pregunta importante es a qué se llama documentos similares.

En el dominio de hipertexto se suman algunas complicaciones más, ya que como se describió anteriormente, no es sólo texto aislado, sino que tiene otro conjunto de características como etiquetas de marcado, urls, dominios, títulos, entre muchos otros. Como complemento a descubrir grupos de páginas web, se suelen extraer y asignar etiquetas a cada *cluster* que lo representen lo mejor posible, y a su vez lo diferencien de los demás.

El **agrupamiento** es un área particular de estudio dentro del aprendizaje automático, siendo un problema muy estudiado y para el cual se han desarrollado diversos algoritmos. Una distinción importante es si la cantidad de grupos es sabida de antemano, o si es algo que se busca determinar. Los algoritmos que toman como parte de su entrada al número de grupos se conocen como **no jerárquicos**, mientras que a los que no buscan una cantidad particular se los llama **jerárquicos**.

El agrupamiento jerárquico busca construir una jerarquía de grupos y se divide en dos tipos: aglomerativo y divisivo. La diferencia es que en el caso aglomerativo se comienza con una cantidad de grupos equivalente a la totalidad de los documentos y se los va agrupando en forma ascendente tomando los de mayor similitud. El tipo divisivo es el caso opuesto, comenzando con un grupo con el total de los documentos y dividiéndolo en forma descendente. Es una técnica que suele aplicarse para visualizar la estructura de agrupamiento de los datos y darse una idea de qué cantidad de grupos podrían formarse mientras se mantenga un nivel deseado de similitud entre los documentos. El resultado es un árbol que se conoce como dendograma y puede verse en la figura 2.2.

[1] DMOZ - http://www.dmoz.org/

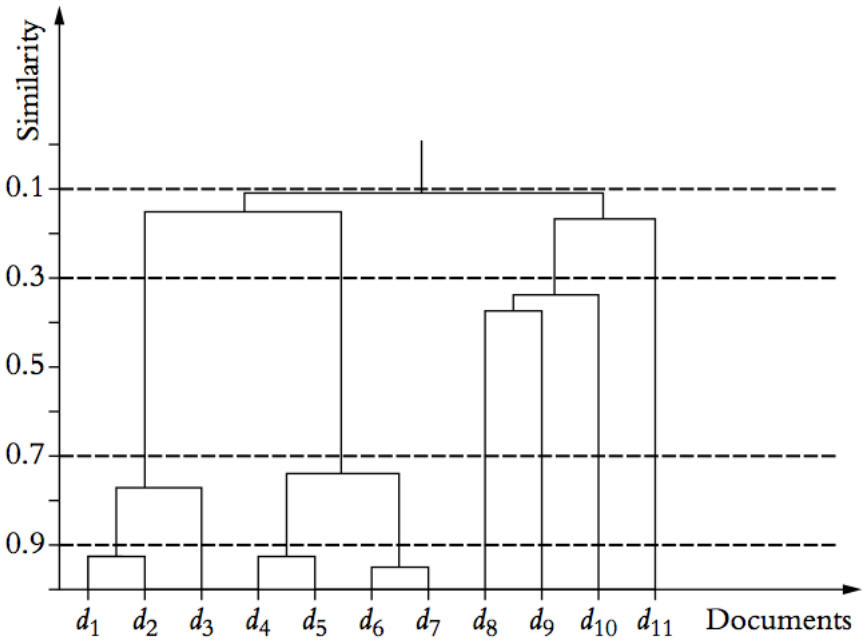

Fig. 2.2: Un dendograma presenta el progresivo y jerárquico proceso de agrupar los documentos. Tomado de [5].

El algoritmo de agrupamiento no jerárquico más utilizado es el de **K-medias**. Como su tipo es no jerárquico, toma como parámetro adicional un número k, la cantidad de grupos que se busca obtener y devuelve los k grupos como resultado. La particularidad en este caso, es que se necesita conocer k con anterioridad para poder utilizar el algoritmo. Esto puede ser bueno o malo dependiendo del caso para el cual sea utilizado.

Tanto en el caso jerárquico como no jerárquico, los resultados van a cambiar sustancialmente dependiendo de la medida de distancia que se tome entre los documentos. Como se dijo anteriormente, en el caso de textos suele utilizarse la distancia coseno entre vectores, a diferencia de un dominio numérico donde quizás tendría más sentido la norma vectorial 2 (euclídea).

Profundizando el caso de las páginas web, se han investigado variaciones que aprovecharan conocimiento específico sobre este dominio para mejorar los grupos encontrados. Un caso práctico que se utilizó para el desarrollo de este trabajo fue el de **Lingo** [12], un algoritmo en el cual se extraen las frases o $n-$gramas más frecuentes de los documentos y se inducen grupos a partir de ellas. A su vez, se hace uso de la descomposición en valores singulares de la **matriz TF-IDF** para encontrar sinónimos y deshacerse de términos que introduzcan ruido en la técnica conocida como *Latent semantic analysis* [13].

Una vez definidos los *clusters* de páginas web para un determinado conjunto, sería interesante poder detectar, ante nuevos documentos, a qué grupo o clase pertenecen. En este aspecto, lo usual es asistirse de técnicas de aprendizaje supervisado o clasificación. Como primer paso, se entrena un clasificador con un corpus de documentos que ya se sabe

de antemano que pertenecen a un tópico específico. El clasificador analiza las relaciones entre los términos en los documentos para formar un modelo. Luego, ante nuevos documentos, se proveen como entrada a los clasificadores de cada tópico para determinar una probabilidad (continua o binaria) de pertenencia a cada clase y se elige según un criterio, por ejemplo, la que tenga mayor probabilidad.

Un problema bastante grande del aprendizaje supervisado es que requiere un gran número de ejemplos etiquetados de cada clase, lo cual es hecho generalmente en forma manual. Como complemento, surgen métodos llamados **semi-supervisados**, que a partir de una pequeña porción de datos etiquetados se realiza un proceso iterativo para ir asignando etiquetas a otros documentos y reentrenando hasta cubrir toda la colección.

Al igual que para la tarea de *clustering*, existen numerosos algoritmos de clasificación con sus ventajas y desventajas para cada dominio. Entre los más utilizados se puede mencionar **Naïve Bayes**, **Support Vector Machines** y **Regresión logística**. Por su relevancia y su uso en este trabajo, describiremos brevemente la teoría detrás del clasificador **Naïve Bayes**. Se puede encontrar más información sobre clasificación en el capítulo 3 de [1] y en los capítulos 13, 14 y 15 de [2].

2.4.1. Clasificador Naïve Bayes

La tarea de clasificación puede ser pensada desde un punto de vista probabilístico como estimar la probabilidad de que un documento pertenezca a una clase determinada. Dado un conjunto de clases o etiquetas $c_1, ..., c_j, ..., c_n$, se quisiera analizar la probabilidad de que el documento d pertenezca a cada una de esas clases y encontrar aquella que la maximiza. Si C representa la clase del documento d, denotamos:

$$Pr(C = c_j \mid d)$$

Formalmente, sean $A_1, A_2, ..., A_{|A|}$ el conjunto de atributos con valores discretos en el set de datos D con todos los documentos. Sea C el atributo correspondiente a la clase con $|C|$ posibles valores $c_1, c_2, ..., c_{|C|}$. Dado un ejemplo d con atributos observados $a_1, ..., a_{|A|}$, donde a_i es el valor correspondiente a A_i, la predicción para la clase es el c_j tal que la probabilidad sea máxima. Es decir que queremos encontrar c_j que maximice

$$Pr(C = c_j \mid A_1 = a_1, ..., A_{|A|} = a_{|A|})$$

Por la regla de Bayes, puede ser reescrito como

$$\frac{Pr(A_1 = a_1, ..., A_{|A|} = a_{|A|} \mid C = c_j) Pr(C = c_j)}{Pr(A_1 = a_1, ..., A_{|A|} = a_{|A|})}$$

$Pr(C = c_j)$ es la probabilidad *a priori* de la clase, la cual puede ser estimada a partir de los datos de entrenamiento. Es simplemente la fracción de los datos D que son de clase c_j.

En el problema de clasificación, el denominador es irrelevante, pues no depende de la clase, es una constante. Entonces, lo único que se necesita calcular es

$$Pr(A_1 = a_1, ..., A_{|A|} = a_{|A|} \mid C = c_j) =$$
$$Pr(A_1 = a_1 \mid A_2 = a_2, ..., A_{|A|} = a_{|A|}, C = c_j) Pr(A_2 = a_2, ..., A_{|A|} = a_{|A|} \mid C = c_j)$$

Recursivamente puede aplicarse la misma reducción en la parte derecha. Para ello, vamos a asumir la condición de que todos los atributos son condicionalmente independientes, dada la clase $C = c_j$. Esto se conoce como ***conditional independence assumption*** y es una hipótesis muy fuerte de este método. Formalmente, se está asumiendo que

$$Pr(A_1 = a_1 \mid A_2 = a_2, ..., A_{|A|} = a_{|A|}, C = c_j) = Pr(A_1 = a_1 \mid C = c_j)$$

y análogamente para todos los A_i. Entonces, lo que se obtiene es

$$Pr(A_1 = a_1, ..., A_{|A|} = a_{|A|} \mid C = c_j) = \prod_{i=1}^{|A|} Pr(A_i = a_i \mid C = c_j)$$

De esta forma, estimar las probabilidades *a priori* de las clases y las probabilidades condicionales de los atributos A_i a partir de los datos de entrenamiento es directo.

$$Pr(C = c_j) = \frac{cantidad\ de\ documentos\ de\ clase\ c_j}{cantidad\ total\ de\ documentos}$$

$$Pr(A_i = a_i \mid C = c_j) = \frac{cantidad\ de\ documentos\ de\ clase\ c_j\ con\ A_i = a_i}{cantidad\ total\ de\ documentos\ de\ clase\ c_j}$$

Para tomar la decisión de asignar una clase a un documento desconocido, lo único que se debe hacer es calcular c en

$$c = arg\ max_{c_j} Pr(C = c_j) \times \prod_{i=1}^{|A|} Pr(A_i = a_i \mid C = c_j)$$

es decir, la clase que maximiza la probabilidad de que el documento pertenezca a esa clase, dados los atributos que contiene.

2.4.2. Métricas de clasificación

Para algunas aplicaciones, como la clasificación de un tópico en páginas web, uno está interesado en una sola clase. La clase que interesa al usuario es comúnmente llamada **clase positiva**, mientras la otra se conoce como **clase negativa**. Por ejemplo, si se buscan sitios de historia que hablen sobre la segunda guerra mundial, un artículo sobre el desembarco de Normandía pertenecería a la clase positiva. En cambio, una página con información sobre la revolución de mayo de mil ochocientos diez sería parte de la clase negativa ya que nada tiene que ver con la segunda guerra mundial.

En los casos donde la clase positiva puede resultar minoritaria, mirar el número de elementos clasificados correctamente (métrica conocida como *accuracy*) puede ser una mala idea. Por ejemplo, si el 99 % de los casos son negativos y el clasificador no es capaz de identificar ningún caso positivo, esta métrica daría un engañoso resultado del 99 %.

En estas aplicaciones, se suelen utilizar otras métricas más adecuadas conocidas como **precision** y **recall**. Estas métricas pueden medir cuán precisa y cuán completa es la clasificación de la clase positiva. Una forma conveniente de introducirlas es utilizando una "matriz de confusión".

		Etiqueta clasificada		
		p	**n**	**total**
Etiqueta real	**p′**	Verdadero Positivo	Falso Negativo	P′
	n′	Falso Positivo	Verdadero Negativo	N′
	total	P	N	

Basado en esta matriz, **precision** y **recall** de la clase positiva se definen como:

$$p = \frac{VP}{VP + FP}$$

$$r = \frac{VP}{VP + FN}$$

donde

- **VP** es el número de resultados clasificados positivos que efectivamente lo eran
- **FP** es el número de resultados clasificados positivos que no lo eran
- **FN** es el número de resultados clasificados negativos que no lo eran

Sin embargo, comparar clasificadores basado en dos métricas no relacionadas puede ser complicado. En principio, se quisieran maximizar ambas, pero al comparar podría ser una más alta que la otra y viceversa. Si bien en teoría no son métricas relacionadas, aumentar una en la práctica suele ser a costa de disminuir la otra.

Si se necesita una única medida para comparar, usualmente se utiliza el F-score, también conocido como F_1-score o F_1-measure. Es una media armónica de ambas, definida como

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

Viendo este cálculo, para que el F-score sea alto, tanto p como r deben serlo, logrando medir con mejor precisión a los clasificadores sobre clases sesgadas.

2.5. Análisis de Hipervínculos

Como se dijo anteriormente, una parte importante de los datos incluídos en hipertexto son los hipervínculos que conectan una página web con otras. Un área importante de estudio en la recuperación de la información aplicada a la búsqueda web es la de analizar los datos que provee el grafo subyacente de una colección de documentos.

Un ejemplo de por qué esto es importante es el siguiente: si se realiza una búsqueda genérica como “universidad” dentro de un determinado conjunto de páginas web utilizando

el *vector space model* de **IR**, las páginas relevantes como la oficiales de universidades como la Universidad de Buenos Aires, o la Universidad Tecnológica Nacional podrían tener menor puntaje asignado que una página donde se hable sobre temas académicos. Sin embargo, hay miles de páginas que tienen hipervínculos a la página de la U.B.A., lo cual la hace destacada en este tema.

La relevancia es muy importante para un buscador, ya que debería poder priorizar una página oficial de una universidad, frente a una noticia o un debate sobre temas relacionados. Esto aplica a cualquier sitio web que sea una fuente confiable de información, frente a por ejemplo un blog. Lo importante en este caso es entender qué hace a esos sitios ser relevantes frente a otros.

Esto llevó al desarrollo de algoritmos para medir el *score* de prestigio a un sitio como el **PageRank** [14], luego utilizado para el desarrollo de Google, o **HITS** (*hyperlink induced topic search*) [15]. Estos algoritmos asignan mayor relevancia a páginas que tengan una mayor cantidad de hipervínculos hacia ellas.

La idea intuitiva detrás puede pensarse de distintas maneras. Un ejemplo es la relación con las publicaciones científicas, que pueden modelarse como un grafo donde las adyacencias son salientes hacia las publicaciones que referencia, y entrantes para aquellas que la referencian. De esta forma, sería conveniente que un *paper* que fue citado muchas veces aparezca primero en una búsqueda sobre algún tema que se describe. Otra forma de pensarlo es: si nos moviéramos aleatoriamente dentro del grafo de la web, ¿Cuál sería la probabilidad de caer en un sitio determinado? Si una página es vinculada desde muchas otras páginas, está claro que la probabilidad de moverse hacia ella es mucho más alta que moverse a otra poco referenciada.

Un concepto importante que presenta **HITS** es el de **hubs** y **authorities**. Un **authority** es una página con muchos *in-links* o hipervínculos hacia ella. La idea es que es una página que tiene contenido de autoridad o importante, y por eso es tan referenciado, al igual que el caso de las publicaciones. Esto es similar al prestigio de **PageRank**.

En contraposición, un **hub** es una página con muchos *out-links* o hipervínculos salientes, un organizador de información en un tópico particular y apunta a muchas páginas de buen contenido como **authorities**.

Estos conceptos son importantes para este trabajo ya que permiten analizar la estructura de las páginas luego de un proceso de *crawling*, pudiendo encontrar componentes altamente conectadas y sitios que probablemente sirvan como semillas en un nuevo *crawling*.

3. DESARROLLO

En este capítulo se explica cómo se aplicaron las técnicas descriptas en el capítulo 2 a lo largo de las distintas etapas de este trabajo. El enfoque es introductorio, dejando los detalles y resultados de cada experimento para el capítulo 5.

3.1. Primeros pasos

3.1.1. Enfoque manual

El primer acercamiento al problema consistió en utilizar los buscadores tradicionales para la búsqueda de algunas palabras clave, particularmente **Google**, **Bing** y **Yahoo!**. Si bien previamente se mencionó que este tipo de buscador no cumplía con lo requerido por nuestros usuarios, la idea fue intentar explorar el espacio de resultados, e intentar capturar la mayor cantidad posible de sitios relevantes, ya sea en forma manual o con algún análisis automático.

Las **keywords** utilizadas en estas búsquedas fueron las siguientes:

- "Cine Documental"
- +Revista "Cine Documental"
- +Festival "Cine Documental"
- +Biblioteca "Cine Documental"
- +Latinoamericano "Cine Documental"
- +Hispanoamericano "Cine Documental"

Es decir, la frase completa "Cine Documental" sola, y luego con el agregado (obligatorio) de las palabras **revista, festival, biblioteca, latinoamericano, hispanoamericano**.

Analizando manualmente los resultados, se extrajeron los primeros sitios en forma manual para tener como punto de partida. Los 25 que se consideraron más apropiados según un criterio propio, que luego fueron validados como portales de interés por la UNTREF, se encuentran en el Anexo A.

Con estos sitios identificados, se hicieron las primeras pruebas de *crawling* utilizando las *url* como punto de partida. La hipótesis fue que utilizando una cantidad de saltos baja (en este caso, se comenzó con 2), los documentos indexados serían relativamente cercanos e interesantes sobre el dominio que se estaba buscando. En este punto, se realizaron distintas pruebas, variando tanto la cantidad de saltos, como la restricción de mantenerse dentro de los dominios web de los sitios iniciales o explorar otros desconocidos.

3.1.2. API de búsqueda

En busca de ampliar el espacio encontrado con los primeros intentos manuales, se utilizaron las API de **Google Custom Search**[1] y **Bing Search**[2] para realizar una búsqueda en forma automática en base a un conjunto de palabras clave. Sobre los sitios obtenidos, se realizaron nuevamente pruebas de *crawling* e *indexing* con una cantidad de saltos siempre menor a 3. A diferencia del caso anterior, hubo que tener en cuenta la alta probabilidad de encontrar mucho ruido, por lo cual se intentó mantener una cantidad baja de saltos y observar el contenido arrojado por las API.

Para analizar estos resultados, se tomó como una primera medida de interés la existencia o no de la palabra **documental** en algún campo relevante del sitio, i.e. el título de la página, el contenido, la url, etc.

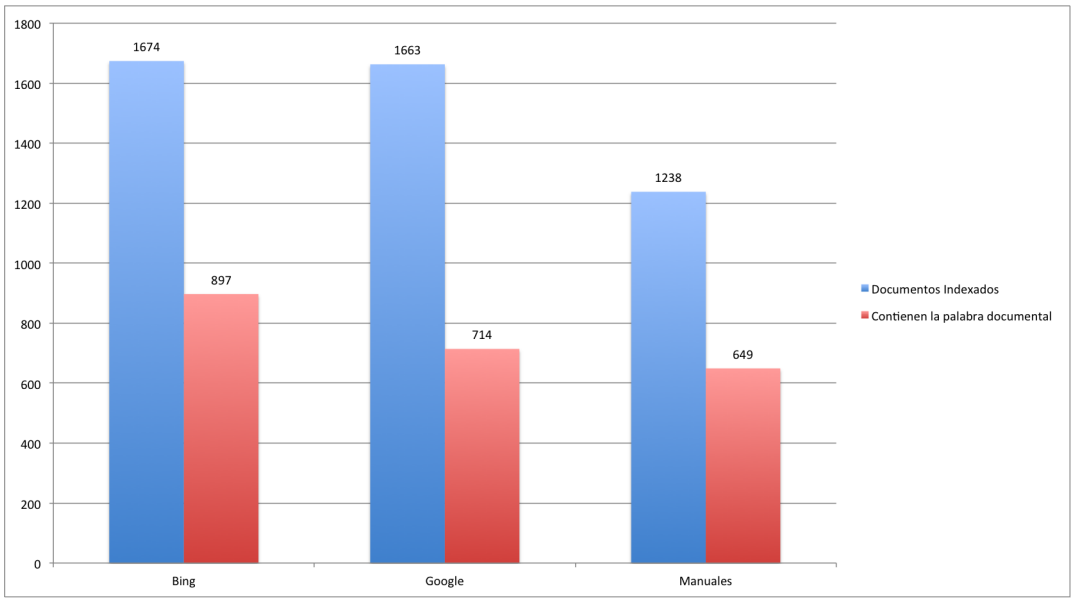

Fig. 3.1: Comparación entre documentos indexados y aquellos que contienen la palabra documental para cada uno de los métodos utilizados

[1] https://developers.google.com/custom-search/
[2] http://www.bing.com/toolbox/bingsearchapi

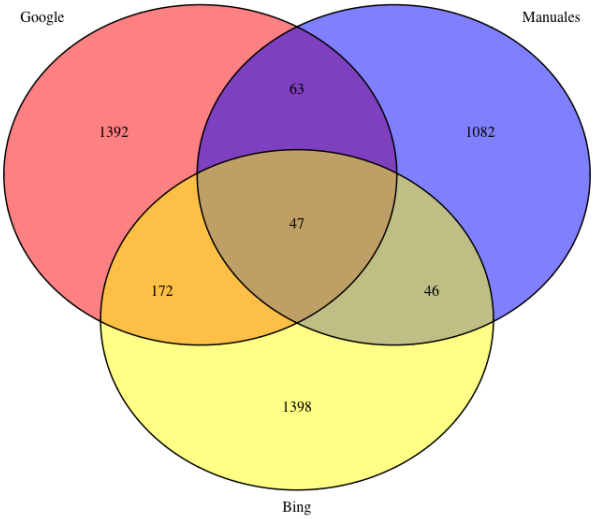

Fig. 3.2: Comparación entre documentos indexados e intersección entre ellos para cada uno de los métodos utilizados

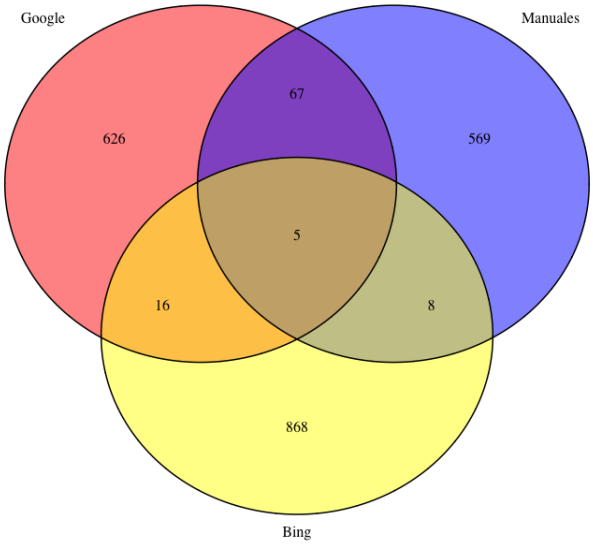

Fig. 3.3: Comparación entre documentos indexados e intersección entre ellos para cada uno de los métodos utilizados en páginas que contienen la palabra documental

En las figuras 3.2 y 3.3 puede verse que la poca intersección existente entre los métodos utilizados se vuelve mucho más baja cuando se toma el subconjunto de aquellas páginas que contienen la palabra documental. Esto podría indicar que filtrando adecuadamente, podrían obtenerse diferentes sitios de interés para cada fuente.

3.2. Introducción de categorías

Para ampliar el corpus obtenido hasta el momento, se expandió el espacio de búsqueda incluyendo como *keywords* algunas categorías definidas como de interés. Las mismas fueron definidas por la UNTREF:

- Documental político
- Documental social
- Documental histórico
- Documental bélico
- Documental de biografías
- Documental antropológico
- Documental científico
- Documental de investigación
- Documental de propaganda política
- Documental de denuncia
- Documental ecológico
- Documental musical
- Documental sobre artes
- Documental del Yo

Además, fueron sugeridas las categorías de **Docu-ficción** y **Falso documental** como de no interés.

Como primera prueba, se realizó un proceso de *crawling* tomando como semillas los primeros 100 resultados de la búsqueda literal de cada una de las categorías en el buscador Bing. Para su análisis y como filtro heurístico para identificar la pertenencia al dominio, se utilizó la presencia del término **cine**.

Aplicando ese filtro, se obtuvo la mayor cantidad de resultados en las categorías de **propaganda política, denuncia y musical**. Sobre estas y otras tres categorías de mayor interés general (documental **político, social e histórico**) se realizó un análisis más profundo de los resultados.

Sobre los datos recolectados de cada categoría se intentó agruparlos y extraer palabras o *topics* representativos de cada grupo. Esta es una técnica muy utilizada ya que permite

identificar tópicos relevantes o emergentes en un gran conjunto de documentos. En particular y a modo exploratorio, puede identificar características de los textos indexados o incluso títulos de sitios interesantes. Este fue el caso de **FilmAffinity** y **Ojos Abiertos**.

Los gráficos y resultados obtenidos pueden verse en el capítulo 5.

3.2.1. Exclusión de ruido

Utilizando la presencia del término cine como filtro y analizando manualmente los grupos o *clusters* formados en cada búsqueda, se identificó una cantidad de ruido significativa en los documentos obtenidos. El criterio manual se basó en detectar las etiquetas o tópicos de cada grupo formado y ver si las páginas de cada *cluster* contenían información acerca del dominio buscado.

A partir de la identificación de ruido, se hizo una prueba más específica en la cual se realizaron búsquedas y *clustering* progresivo, duplicando la cantidad de resultados utilizados como semillas en cada iteración. Es decir, analizar los grupos formados en los primeros 10 resultados, luego en los primeros 20, 40 y 80. La hipótesis detrás de esto fue que quizás el ruido era introducido luego de una cierta cantidad de páginas de resultados. La conclusión ante esta prueba fue que aún entre los primeros resultados de las búsquedas realizadas podían encontrarse páginas que nada tenían que ver con el dominio buscado, reforzando lo dicho sobre el método que tienen los buscadores tradicionales para ordenar los resultados.

Luego de esto, se buscaron sitios que aparecieran para las mismas búsquedas tanto en Bing como en Google, nuevamente tratando de obtener dominios o sitios que pudieran ser relevantes. Dado que cada uno tiene su propio algoritmo para ordenar resultados, aquellos que aparezcan en ambas pueden ser interpretados como potencialmente más relevantes. Un ejemplo de esto es el portal **AtlantiDoc** (http://www.atlantidoc.com/), correspondiente a un festival internacional de cine documental uruguayo.

3.3. Aprendizaje automático - Clasificación

Tras no encontrar una forma clara de obtener resultados de interés, o bien de excluir el ruido de las búsquedas, se intentó recurrir a métodos de aprendizaje automático que pudieran aprender a hacer esta tarea en forma automática. Habiendo ya explorado el espacio de búsqueda dado por los buscadores tradicionales, se abrió un nuevo subproblema que fue el de distinguir automáticamente entre páginas que fueran de interés y páginas que no lo fueran.

Como primera prueba, se tomó una muestra de los primeros 30 resultados arrojados por Google para cada categoría, que se envió a la UNTREF para que anotaran en cada uno si era o no de su interés. Mientras tanto, se pensó en cómo incorporar esto a la idea de buscador que se había concebido originalmente. A la funcionalidad de búsqueda y despliegue de resultados, se agregaron dos botones con la posibilidad de marcar si el resultado que se estaba viendo era de interés o no. El objetivo era poder ir retroalimentando constantemente la base de datos y proveer resultados más exactos.

Fig. 3.4: Prototipo de buscador con posibilidad de clasificar resultados manualmente

Con los resultados obtenidos, se entrenaron tres clasificadores distintos (**Naive Bayes, SVM y Max Entropy**) para realizar las pruebas. El objetivo fue tener una cantidad interesante de datos ya clasificados manualmente para tener un clasificador que pueda asistir al *crawling* de nuevas páginas, filtrando los resultados de interés.

El problema con este proceso es que es difícil de escalar. Los clasificadores tienen buenas métricas cuando los datos que se tienen son muchos, para lo cual fue necesario incrementar el corpus de documentos de interés. Una técnica para esto es la de *bootstrapping*, que consiste en partir de unos pocos documentos, clasificar una nueva porción, confiar en que la etiqueta es buena y repetir sucesivas veces.

Antes de recurrir a esto, lo que se hizo fue tomar los sitios obtenidos originalmente como de interés puro (las semillas de las primeras pruebas, sumado a otras que se encontraron en el camino) e indexar páginas dentro de esos dominios, bajo la hipótesis de que todos los documentos recuperados iban a ser de interés. Vale la pena observar que para que esto sea cierto, es necesario filtrar a sitios realmente idóneos, ya que un portal de cine general, con una categoría de documentales no lo cumple.

En contraposición, se tomaron páginas genéricas de noticias o de sitios encontrados en las búsquedas anteriores, catalogadas fácilmente como ruido, como Facebook o Wikipedia. Estos casos se encuentran en el Anexo B y sirvieron para poder tener una clase de no interés con una cantidad de ejemplos similar a la positiva, y no sólo unos pocos, lo cual pudiera sesgar fuertemente al clasificador.

Como contenido adicional para el corpus, se incorporaron todas las páginas obtenidas de búsquedas en sección documental de los diarios Clarín y La Nacion. Por ejemplo, se generaron como semillas todas las URL dentro de cada buscador, reemplazando números de página desde 1 a 100 en `http://buscador.clarin.com/documental?filter=content_section:Cine;&page=pagina`.

3.3.1. Pruebas realizadas

Una vez obtenida una porción interesante de documentos, se utilizó **R** para algunas pruebas cuantitativas. Para estas, se tomaron primero 1000 documentos seleccionados al azar de los previamente etiquetados, un 80 % para entrenamiento, y otro 20 % para *test* del clasificador. Además, se utilizaron otros 150 documentos aleatorios para una validación cruzada y entendimiento de los resultados. Estos fueron seleccionados de los 30 sitios de cada categoría enviados a la UNTREF para un etiquetado manual, con el objetivo realizar la validación con el criterio adecuado. A lo largo de las pruebas se expandió la cantidad de documentos involucrados, llegando hasta los 16000 documentos como entrada del entrenamiento y 4000 para *test*.

Se estudió la diferencia e impacto en los resultados del tipo de clasificador, así como distintos sistemas de pesos para la matriz de términos. Se utilizó por un lado clasificación bayesiana, SVM (*support vector machines*) y Max Entropy (regresión logística), y por otro, pesos como TF y TF-IDF. También se realizaron pruebas eliminando términos ralos para reducir el *overfitting* del clasificador y aplicando *latent semantic analysis* para mejorar la información extraída del uso de cada término en un documento. Los resultados se encuentran en la sección 5.3.

3.4. Contexto de búsqueda

Luego de analizar y contrastar resultados etiquetados manualmente por la UNTREF con clasificaciones hechas con criterio propio, se encontró que la decisión de si una página es de interés o no, depende de más factores que sólo su texto. El modelo utilizado para clasificar se basó en características derivadas del *vector space model* de cada documento. Sin embargo, se encontraron casos donde una misma página podía ser de interés para una búsqueda, y no de interés para otra.

Esto genera un problema que permitió pensar en refinamientos del proceso, como tener etiquetas o grupos pre-armados dentro del índice, como podrían ser **Videos, Festivales, Críticas, Blogs, Asociaciones, Fichas** entre otras. En lugar de tener sólo dos clases como **Interesa** y **No interesa**, un documento podría clasificarse en cada uno de esos grupos con cierta probabilidad, y por otra parte a la clase negativa. Es una alternativa que daría una ayuda extra al usuario a la hora de buscar, aunque aún no fue implementada.

4. IMPLEMENTACIÓN

En este capítulo se describen los aspectos relacionados a la arquitectura del prototipo construído. Entre ellos se encuentran las decisiones tecnológicas tomadas, herramientas utilizadas, problemas encontrados y las soluciones planteadas frente a cada uno. Se introduce brevemente un diagrama de los componentes, ampliando luego en cada sección sobre los detalles específicos.

La construcción del prototipo de buscador estuvo compuesta de diferentes partes, cada una con funciones específicas dentro de un flujo de búsqueda. Esto comprende desde la obtención de páginas y el almacenamiento e indexado de su contenido, hasta la interfaz gráfica donde un usuario realiza una búsqueda y se le muestran resultados. En la figura 4.1 se puede ver un esquema general del funcionamiento de la herramienta.

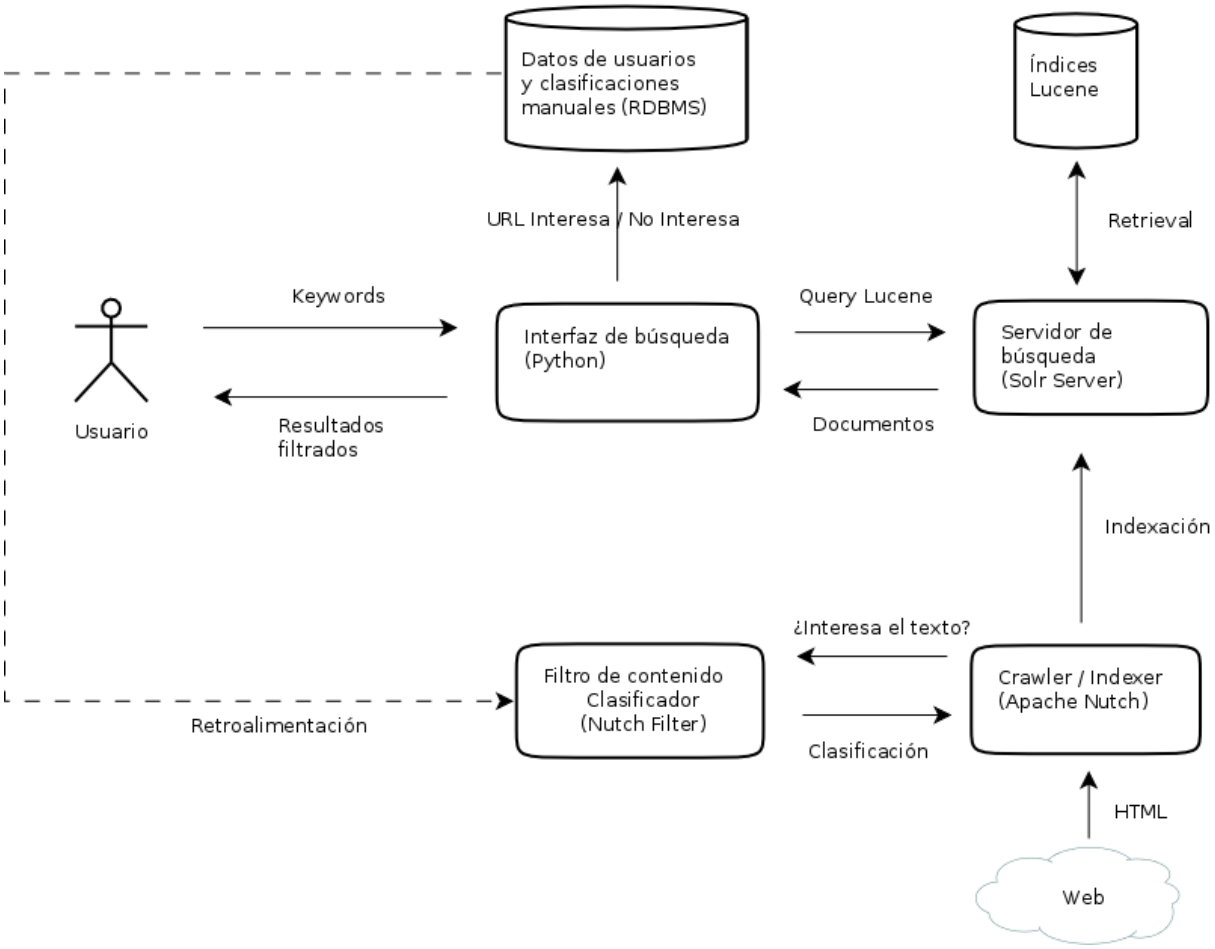

Fig. 4.1: Diagrama de componentes de la arquitectura del prototipo

4.1. Crawler / Indexer

Para la etapa de *crawling* se implementó el conocido y ampliamente utilizado **Apache Nutch**[1], en su versión 1.7. Además de sus amplias funcionalidad e integración con otras herramientas de indexado de texto, puede ser utilizado desde pruebas locales hasta un despliegue productivo en un *cluster* de servidores, a través de su integración con Hadoop para el almacenamiento de los datos obtenidos.

Los parámetros de configuración que tiene son muchos y en el caso de este trabajo se los aprovechó para diversas pruebas. Los más explorados fueron:

- La cantidad de redirecciones a seguir al obtener una página.
- Si seguir links para obtener nuevas páginas o quedarse únicamente con las semillas.
- Si seguir links a hosts externos o sólo quedarse con aquellos dentro del dominio de las semillas. Este parámetro es muy útil, aunque trae algunas complicaciones cuando un mismo sitio utiliza varios subdominios.
- Aplicar filtros con expresiones regulares a las URL que se siguen. Es una forma alternativa, más compleja y completa del parámetro anterior. En este caso se puede resolver el problema de los subdominios. Un ejemplo para esto sería el buscador de clarín, ubicado en *buscador.clarin.com*, pero que contiene links hacia el dominio *clarin.com*. Si se aplica el parámetro anterior, provoca no obtener resultados porque difiere su subdominio. Sin embargo, puede resolverse capturando URLs que cumplan con la expresión regular:

  ```
  (buscador\.)?clarin\.com
  ```

Por otra parte, Nutch permite una fácil extensibilidad, dando la posibilidad de programar filtros personalizados a aplicar a los documentos, lo cual se utilizó para la clasificación entrenada con anterioridad. Se detalle más sobre esto en la sección 4.3.

4.2. Servidor de búsqueda

En cuanto al indexado y posterior búsqueda de los textos obtenidos en la etapa de *crawling*, se utilizó **Apache Solr**[2]. Es un servidor de búsqueda open-source que permite la abstracción y exposición de una capa de servicios sobre índices **Lucene**. Lucene[3] es una biblioteca con muchos años de maduración para tareas de *information retrieval* y *full text search* que brinda una gran cantidad de funcionalidades como el almacenamiento, procesamiento y la realización de consultas o búsquedas sobre un repositorio de textos.

Solr utiliza Lucene internamente y a su vez le agrega funcionalidades, dando la posibilidad de definir estructuras de datos para los índices y la exposición de servicios a través de una interfaz web. Esto permite realizar consultas o agregar documentos utilizando un estándar de *requests* HTTP, sin necesidad de conocer la estructura interna de los archivos que contienen los índices.

[1] http://nutch.apache.org/
[2] http://lucene.apache.org/solr/
[3] http://lucene.apache.org/

La integración de estas herramientas con Nutch es automática, almacenando los resultados del *crawling* en forma de índices invertidos y permitiendo luego realizar consultas.

Los parámetros de configuración en este caso son también muy amplios, con posibilidad de integrar *plugins* externos para una gran cantidad de tareas como *Stemming*, Normalización, *Stop words*, Tokenización, entre otras. En las pruebas realizadas en este trabajo, se utilizó por ejemplo la variación del operador lógico utilizado para la búsqueda. Si se toman *keywords* individuales sin la inclusión de comillas, es muy distinto dar como resultados a aquellos documentos que contengan alguna de esas palabras o a aquellos que contengan todas. Otro parámetro con el que se experimentó fue el sistema de pesos utilizado para la etapa de *retrieval*, por defecto **TF-IDF**. En este caso, lo que se ve modificado es el orden en que se muestran los resultados, variando la relevancia adquirida por cada documento con respecto a las palabras clave.

4.3. Clasificador

Para la clasificación, se utilizó como primera prueba la herramienta **Mallet**[4], un paquete desarrollado en Java para procesamiento de lenguaje y clasificación de documentos, entre otros. Permite con facilidad entrenar clasificadores de varios tipos (Bayes, Árboles de Decisión, Máxima Entropía, C45) con tan solo disponer de un archivo csv con los documentos y las clases previamente asignadas, generando un archivo para su posterior reutilización.

A partir de esto se puede, ya sea desde su ejecutable, o a través de su API en un programa Java, clasificar nuevos documentos que se vayan obteniendo. Esto sirvió para poder integrarlo en un *custom filter* de Nutch, programable también en lenguaje Java, para decidir al momento del *crawling* si indexar o no una página para la posterior alimentación del buscador.

Además de Mallet, se utilizó para realizar pruebas más específicas el lenguaje **R** con los paquetes **RTextTools** para aprendizaje automático y clasificación de textos. La ventaja en este caso es que, a diferencia de lo anterior, se tiene mayor control sobre las *features* extraídas de los documentos, de las matrices generadas, los filtros aplicados (normalización, *stop words*, etc.) para un análisis más profundo que utilizar un ejecutable como caja negra.

Una ventaja en la utilización de R fue la de comparar configuraciones de los clasificadores, verificando resultados similares a los de Mallet y a su vez pudiendo entender mejor su entrada y funcionamiento. Mallet, como está provisto, funciona como caja negra, no pudiendo ver ni analizar las matrices y resultados intermedios.

4.4. Interfaz de búsqueda

La interfaz de búsqueda realizada a modo de prototipo fue bastante simple, presentando una caja de búsqueda y un listado con los resultados, resaltando con letra de mayor tamaño la coincidencia con las *keywords* utilizadas. Para el *matching* entre el cuerpo del texto y las palabras buscadas se utilizó la funcionalidad de *highlighting* incluída con Solr.

Por otra parte, se incluyó la posibilidad de marcar el resultado como positivo o negativo, con el objetivo de retroalimentar al clasificador y obtener mejores resultados, como se explicó anteriormente. En la figura 4.2 puede verse una captura de pantalla de una

[4] http://mallet.cs.umass.edu/

búsqueda de ejemplo, remarcando en rojo los botones relacionados al interés en el resultado.

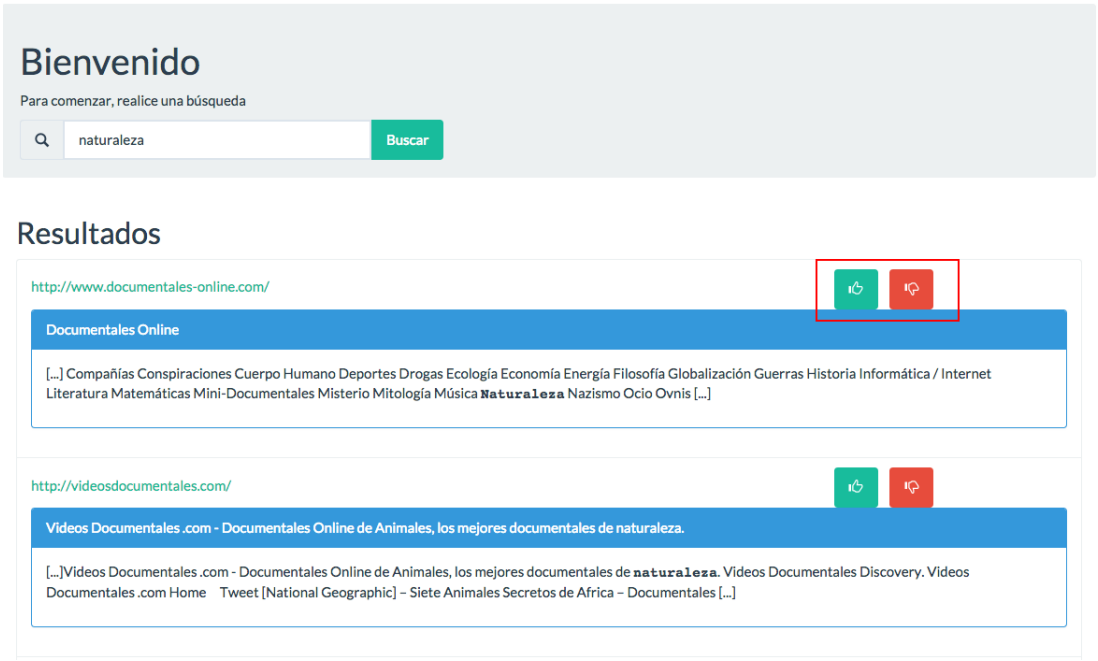

Fig. 4.2: Captura de pantalla con la búsqueda "naturaleza"

4.4.1. Filtro de resultados

Un problema encontrado por los usuarios fue el de resultados aparentemente duplicados al realizar algunas búsquedas. Al analizar ejemplos, se entendió que un problema muy recurrente es el de sitios que comparten una porción en común para todas las páginas, como un menú o un encabezado. Si se busca alguna palabra contenida en ese menú o encabezado, presente en todas las páginas, cualquiera de ellas es igualmente relevante. Por ejemplo, el sitio `http://www.fidba.com.ar/` contiene un menú y el título con la palabra festival. Más allá de alguna página específica, todo el resto del sitio es candidato frente a la búsqueda "festival", siendo todas de una importancia similar, ya que la porción donde aparece esa palabra es la misma y en el mismo contexto.

Este es un problema muy complejo, que requiere analizar el hipertexto de cada sitio por separado para darle semántica específica y peso a cada nodo HTML. Como alternativa, se diseñó un algoritmo para detectar potenciales duplicados en los resultados y ocultarlos, basado en la distancia de Levenshtein[5] entre los resultados. Recordemos que si bien las páginas son diferentes, lo que coincide es la porción de texto alrededor de las palabras buscadas, es decir el resultado devuelto por el servidor de búsqueda. Esta porción que rodea a las palabras buscadas se conoce como *snippet*.

[5] La distancia de Levenshtein o de edición es el número de operaciones de inserción, modificación o eliminación de caracteres necesarias para transformar una cadena en otra.

```
Data: lista de resultados de tamaño n, con campos url y snippet
Result: conjunto de potenciales duplicados
duplicados ← conjunto();
for x ← resultados[0...n) do
    for y ← resultados[0...x) do
        if x.url ≠ y.url then
            distancia ← levenshtein(x.snippet, y.snippet);
            distNorm ← distancia / max(longitud(x.snippet), longitud(y.snippet));
            if distNorm < 0.5 then
                duplicados ← duplicados ∪ {y};
            end
        end
    end
end
```

Algorithm 1: Detección de resultados de búsqueda duplicados

En cada paso, se compara un resultado contra todos los demás (salvo sí mismo), normalizando la distancia por el largo de alinear las cadenas, es decir el máximo entre ambas. Si la distancia normalizada es menor a 0,5, se lo considera duplicado y no será mostrado entre los resultados.

En cuanto al costo computacional de este proceso en el peor caso, se tiene una doble iteración por el tamaño de una página de resultados, el cálculo de la distancia, una operación aritmética y el agregado a un conjunto. Dejando de lado el costo asociado al conjunto de resultados, que depende de su implementación, la complejidad total en peor caso es:

$$\mathcal{O}(\sum_{i=1}^{n}\sum_{j=1}^{i} levenshtein(snippet(resultado_i), snippet(resultado_j))) \tag{4.1}$$

siendo n el tamaño de la página de resultados.

El algoritmo de distancia de Levenshtein tiene como complejidad $\Theta(s * t)$, siendo s y t los largos de las cadenas comparadas. En este caso, las cadenas tienen siempre tamaño constante, ya que el *snippet* o fragmento extraído del texto es el mismo, consistiendo de 250 caracteres alrededor de las palabras clave encontradas. Esto resulta en un costo constante en cada iteración. Reemplazando en la expresión 4.1 se obtiene:

$$\mathcal{O}(\sum_{i=1}^{n}\sum_{j=1}^{i} 1) = \mathcal{O}(n^2) \tag{4.2}$$

La cantidad de resultados obtenidos es constante, siendo por defecto 20 por página. De esta manera, la complejidad final asociada a este filtro resulta ser $\mathcal{O}(1)$, es decir constante.

Si bien no se logró solucionar el problema de fondo, se pudo mejorar mucho la calidad de los resultados. Queda pendiente como trabajo futuro analizar la estructura de los sitios para no sólo eliminar duplicados, sino para darle menor peso a las palabras dentro de las secciones poco relevantes como el menú. Esto lograría que no se muestren entre los primeros resultados de búsqueda, sino que queden para las últimas páginas, dado que su importancia para el usuario es baja.

5. RESULTADOS Y ANÁLISIS

En este capítulo se muestran y analizan los resultados obtenidos para los experimentos descriptos en el capítulo 3. Se utilizan diferentes métodos de visualización, algunos más ordinarios como tablas o gráficos de barras y otros más específicos del dominio como las matrices de confusión, introducidas en la sección 2.4.2.

Para analizar los resultados de agrupar páginas web se utilizó una herramienta llamada **Aduna**[1]. Ésta toma como entrada *clusters* previamente formados con un algoritmo de agrupamiento y tópicos o palabras representativas asignadas a cada uno. Para estos experimentos se utilizó el algoritmo Lingo [12], mencionado previamente en el capítulo 2, el cual asigna etiquetas automáticamente a los grupos luego de hacer un procesamiento del texto mediante las técnicas descriptas en 2.3.1.

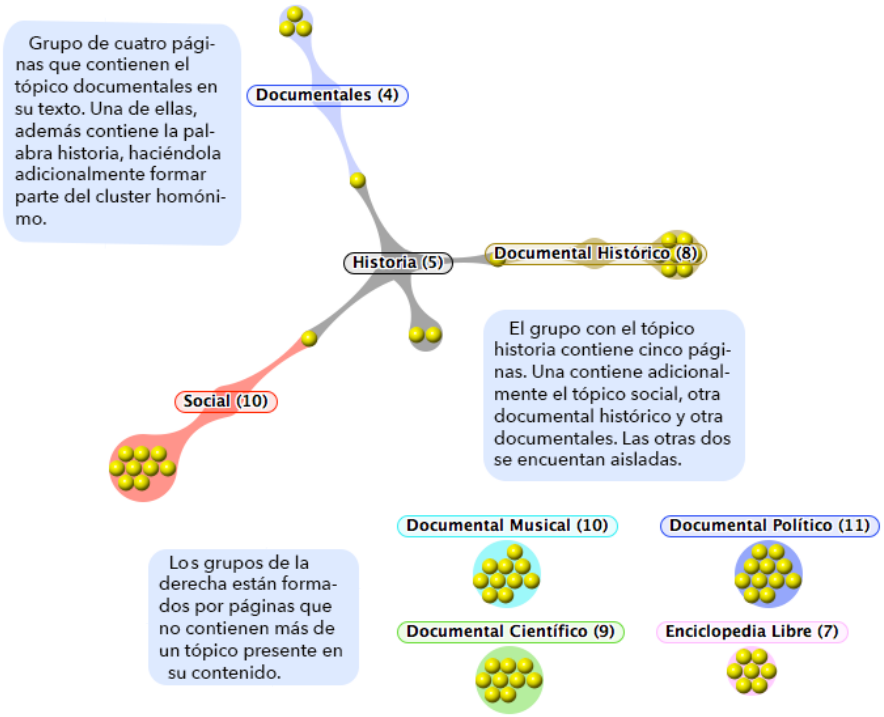

Fig. 5.1: Gráfico producido por la herramienta Aduna para un conjunto de resultados de ejemplo

En la figura 5.1 puede verse un ejemplo del tipo de gráfico que se utilizó para entender los *clusters* encontrados. En el caso de los colores, son asignados aleatoriamente y no conllevan ningún significado particular. Una particularidad para que esta visualización tenga sentido es que el algoritmo utilizado debe poder asignar más de un cluster a cada elemento; de otra manera quedarán todos aislados.

[1] Aduna Map Visualization - http://www.aduna-software.com/technology/clustermap

5.1. Crawling - Agrupamiento

A continuación se presentan los resultados del desarrollo planteado en las secciones 3.1 y 3.2. Como primer enfoque, se realizaron procesos de *crawling* sobre los resultados obtenidos del buscador bing para las búsquedas no literales de cada categoría, es decir sin inclusión de comillas. Es importante recordar que una búsqueda con comillas implica encontrar aquellos textos que contengan completa a la expresión encerrada entre comillas. Una búsqueda sin comillas puede dar como resultado páginas que contengan a las palabras clave en diferentes lugares del texto.

Sobre las páginas o documentos indexados a partir del *crawling*, se aplicó como prueba exploratoria el algoritmo Lingo en busca de tópicos emergentes en cada búsqueda por categoría. Los resultados obtenidos se presentan mediante una tabla que describe cada búsqueda realizada, la cantidad de resultados devueltos por el buscador o semillas para el crawling, los documentos indexados a partir de esas semillas y los tópicos de los *clusters* de mayor tamaño encontrados.

En forma complementaria, se incluye un gráfico de barras con la cantidad de resultados o semillas obtenidas del buscador y la cantidad de páginas o documentos que pudieron obtenerse a partir de esos resultados. Este tipo de gráfico permite una comparación visual más apropiada que sólo mirar los números en una tabla.

Resultados indexados a partir de las categorías en Bing con agrupamiento Lingo

Búsqueda	**Semillas**	**Docs. indexados**	**Tópicos emergentes**
cine documental	869	1122	Documental, Festival, BLOG
festival cine documental	873	1058	Documental Cine, Festival Internacional
biblioteca cine documental	735	1101	Documental Revista, Películas, Cultura
documental político	818	992	Documental Político, **Ojos Abiertos**
documental social	867	923	Documental Social, Fotografía
documental histórico	849	1072	Histórico, Archivo Histórico, Historia
documental bélico	802	1075	Bélico, **Gratis**, Estrenos
documental de biografías	558	960	Biografías, Documentales, **Descargar MP3**
documental antropológico	861	911	Antropología, Facultad
documental científico	896	1060	Documental, Ciencia, Investigación
documental de investigación	765	919	Investigación, Documental de Investigación, Documental
doumental de propaganda política	757	1092	Política, Propaganda, Documentales en la Red
documental de denuncia	867	1130	Denuncia Contra, Video
documental ecológico	603	947	Ecológico, Documental Ecológico, Ecología
documental musical	709	1061	Documental Musical, **Descargar Música**
documental sobre artes	853	1061	Video, Documental Arte, Video Documental, **Buscar y Encontrar Videos Online**
documental del yo	865	1060	Blog, Videos, Blog del documental

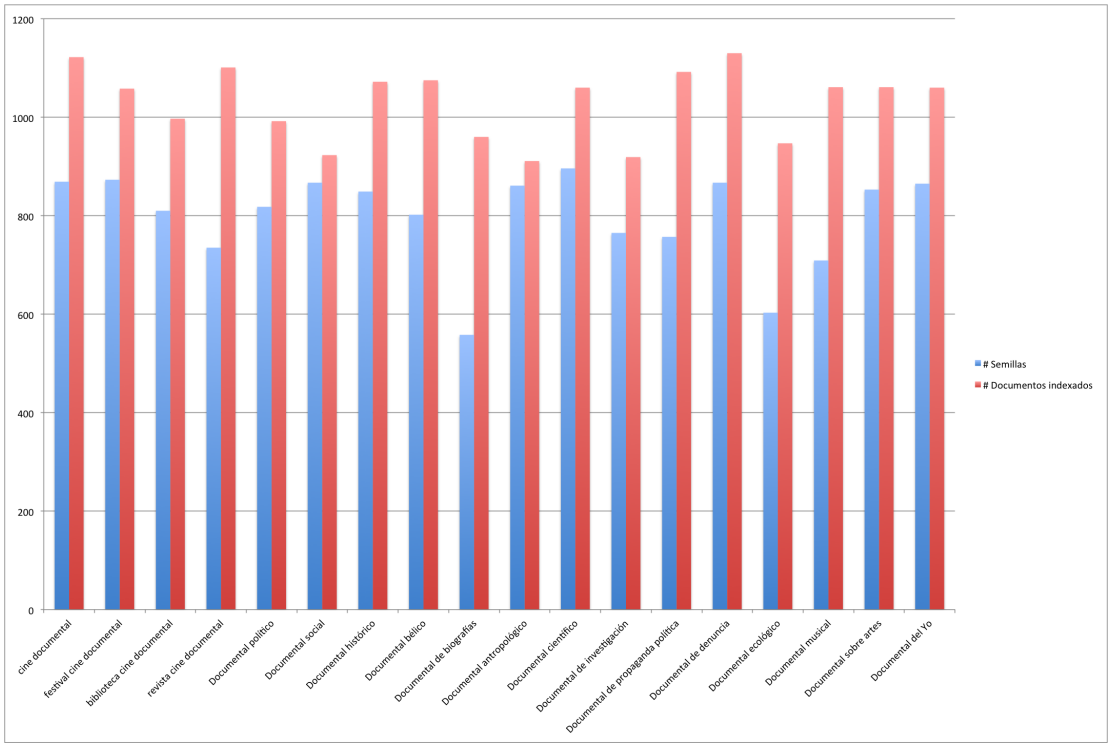

Fig. 5.2: Resultados de búsqueda y documentos indexados por cada categoría en el buscador Bing

En la figura 5.2 puede verse una cantidad de resultados similares para cada categoría, salvo para **documental de biografías** con una cantidad de semillas obtenidas ligeramente menor. En cuanto a los tópicos, se obtienen etiquetas que tienen sentido dentro de la búsqueda, algunas más triviales que otras. Se destacan en negrita algunos casos particulares, como el de **Ojos Abiertos** para la búsqueda de documental político. Se trata de un blog de análisis y críticas de festivales, el cual se pudo descubrir a partir de estos agrupamientos. Otros tópicos son simplemente ruido como **Descargar Música** o **Descargar MP3**.

Si bien la mayoría de los tópicos expuestos anteriormente es coherente con cada búsqueda, una gran parte de los resultados obtenidos fue catalogado por el algoritmo como **Otros tópicos**. Esto evidencia una gran cantidad de ruido, ya que se deja una gran cantidad de resultados por fuera de los grupos con tópicos relevantes. Teniendo esto en cuenta, se realizó otra prueba similar, pero incluyendo comillas en las categorías. Como se mencionó anteriormente, para un motor de búsqueda implica encontrar la frase exacta que se está buscando, en lugar de alguna de las *keyword*.

Además, como una validación mínima de aporte, se calculó el porcentaje de resultados que incluyeran la palabra **cine**. Ante la ausencia, se puede concluir que el resultado no aporta a lo buscado. La presencia de **cine** es un dato interesante, porque al buscar categorías como **"Documental ecológico"** se pueden obtener resultados de otro tipo de documental, no audiovisual, como puede ser un libro. A estos casos se los querría filtrar ya que el dominio de interés es el cinematográfico. Se tomó como criterio que es mejor un mayor porcentaje de inclusión de **cine**, por su importancia para el dominio buscado.

Resultados indexados a partir de las categorías en Bing con comillas

Búsqueda	**Semillas**	**Docs. indexados**	**Contienen *cine***	**Contienen *cine* (%)**
cine documental	856	1071	774	72,27 %
festival cine documental	286	906	440	48,57 %
biblioteca cine documental	6	74	15	20,27 %
documental político	379	891	383	42,98 %
documental social	559	820	279	34,02 %
documental histórico	701	538	133	24,72 %
documental bélico	209	919	362	39,39 %
documental de biografías	80	740	362	48,92 %
documental antropológico	416	852	255	29,93 %
documental científico	551	946	231	24,42 %
documental de investigación	759	1041	304	29,20 %
doumental de propaganda política	56	446	250	**56,05** %
documental de denuncia	812	1059	520	**49,10** %
documental ecológico	361	908	347	38,22 %
documental musical	577	1045	513	**49,09** %
documental sobre artes	161	841	217	25,80 %
documental del yo	19	96	45	46,88 %

Para esta prueba se incluyen dos columnas adicionales en la tabla que contienen información sobre la presencia del término **cine** en las páginas. Esto puede verse gráficamente en la figura 5.3, donde mayor altura de las barras azules significa que la búsqueda dio mayor cantidad de resultados, mientras que la altura de las barras rojas implica que una mayor cantidad de páginas incluyó a la palabra cine.

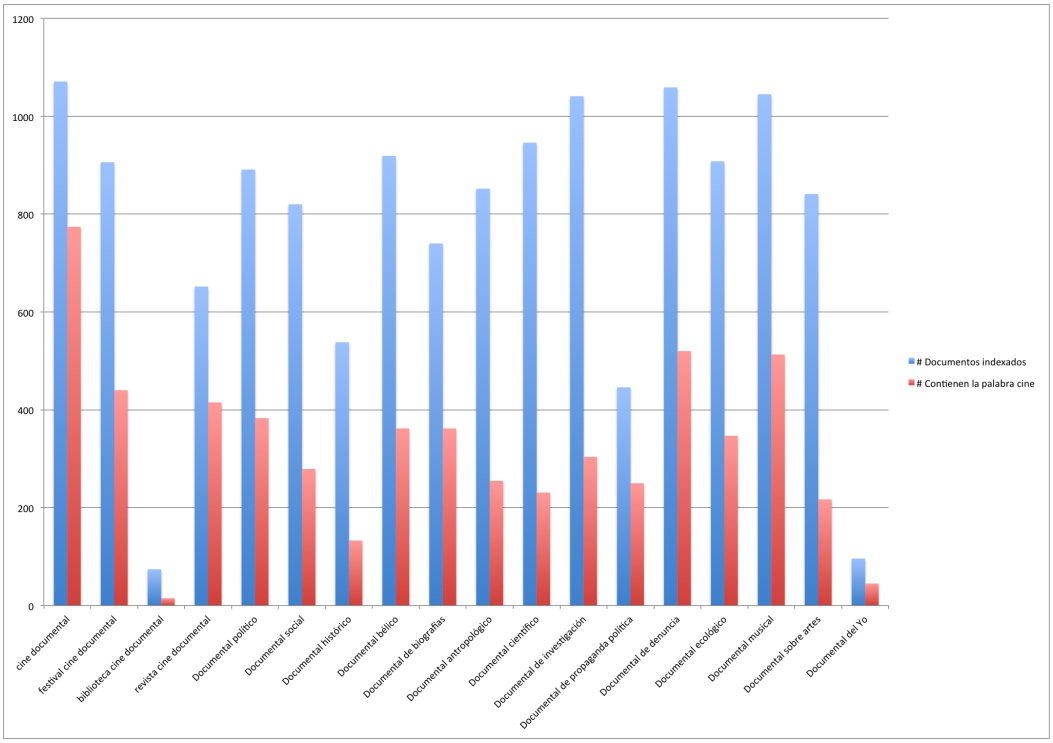

Fig. 5.3: Resultados de búsqueda y documentos indexados por cada categoría en el buscador Bing (con comillas)

En primer lugar, puede mencionarse que aún las búsquedas de **cine documental** que de por sí incluyen la palabra cine, llevan a resultados con ruido. Esto tiene sentido, ya que difícilmente una página contenga todos sus links salientes hacia lugares completamente relacionados. En el resto de las categorías se ven resultados variados, destacando en la tabla con negrita las de mayor porcentaje de inclusión de **cine**. Para ellas se aplicó la técnica de clustering para entender mejor los resultados. A continuación se muestran las visualizaciones realizadas, que pueden interpretarse utilizando la figura 5.1 como ejemplo.

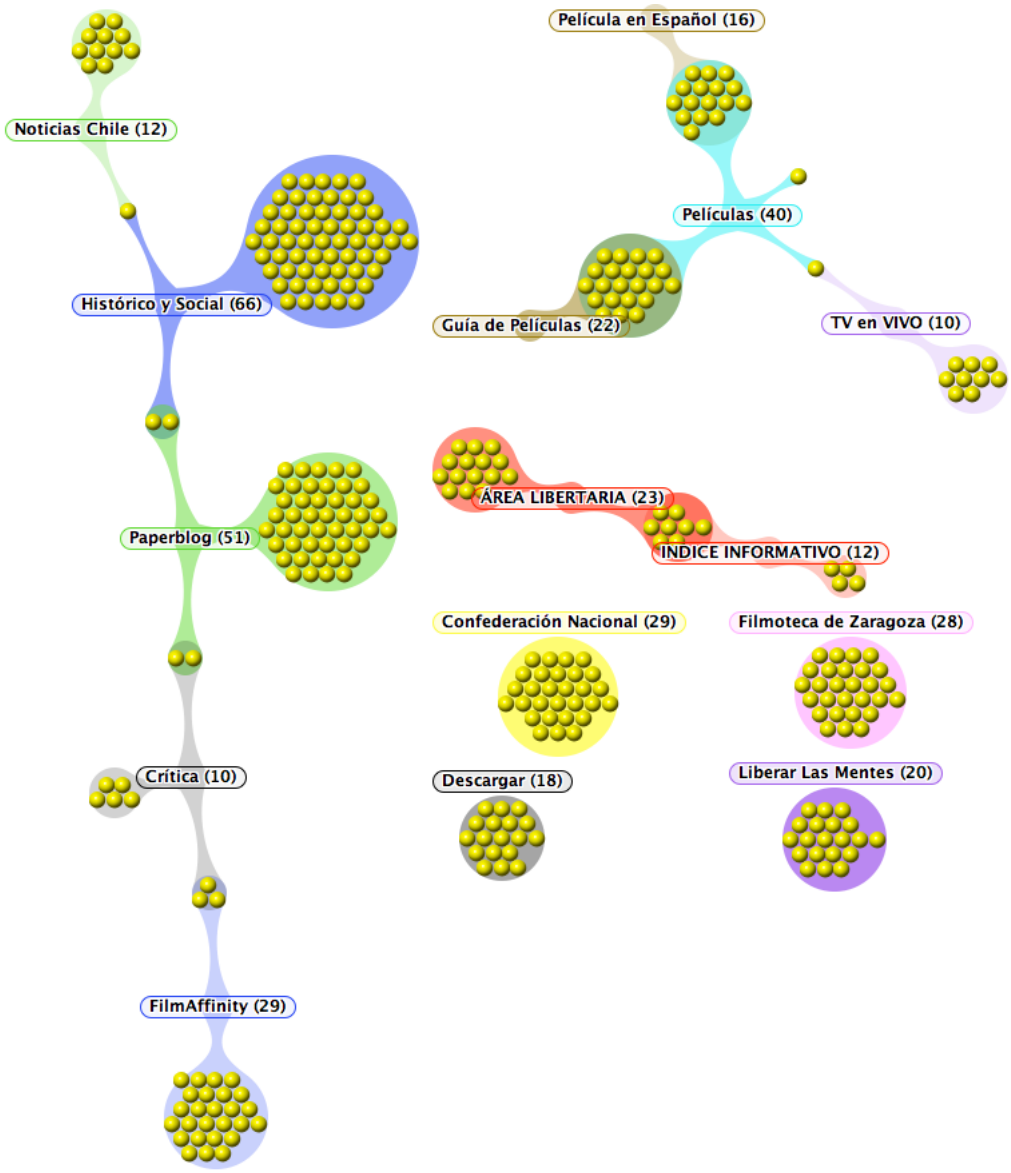

Fig. 5.4: Grupos más importantes para la categoría de propaganda política

En la figura 5.4 puede verse que los grupos más grandes son **Histórico y social**, **Paperblog** y **Películas**. Para esta etapa, se estudió con mayor detalle cada categoría, con el objetivo de entender la estructura de las búsquedas y los sitios encontrados. El grupo **Histórico y social** es heterogéneo, dado que es un tópico bastante inclusivo desde lo semántico.

En el caso de **Paperblog**, es una revista de blogs, cluster en el cual caen muchos artículos sobre fichas de películas, no necesariamente relacionadas con documentales. Se llega a partir de una semilla http://es.paperblog.com/el-triunfo-de-la-voluntad-435633/ "...Y así nació el que para muchos es el mejor documental de propaganda política jamás filmado...".

Para **Películas**, se tienen todas páginas provenientes de http://www.fulltv.com.ar/peliculas/bajo-el-signo-libertario.html "Documental de propaganda política que, con un lenguaje militante, expone las concepciones anarquistas sobre la organización de la sociedad revolucionaria".

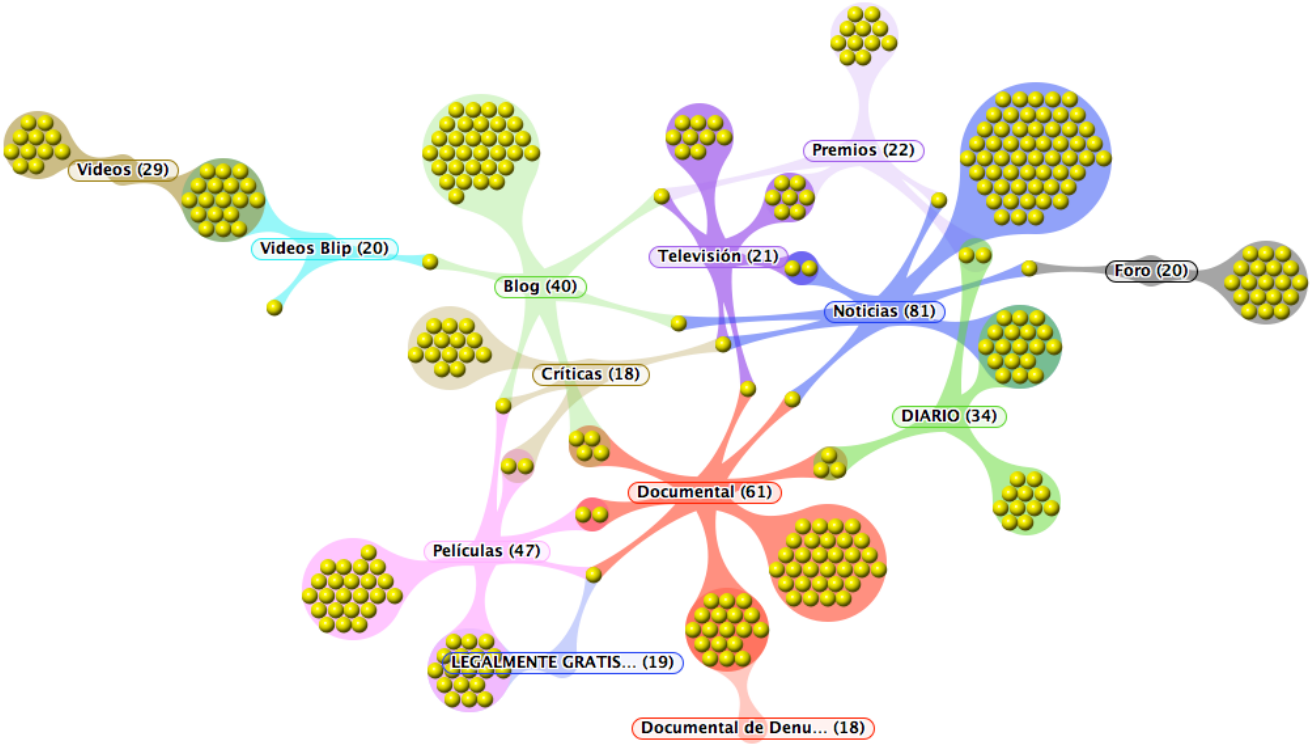

Fig. 5.5: Grupos más importantes para la categoría de denuncia

Para la categoría de denuncia, los clusters no tienen ningún tópico fuera de lo que se esperaría, aunque con categorías interesantes como opiniones, premios, blog y crítica sobre cine en general. Sí destaca el de **Legalmente gratis** (http://legalmentegratis.com.ar/category/documental/) sitio de películas online con categorías como documental, incluyendo documentales de denuncia como **El mundo según Monsanto** contra la empresa multinacional.

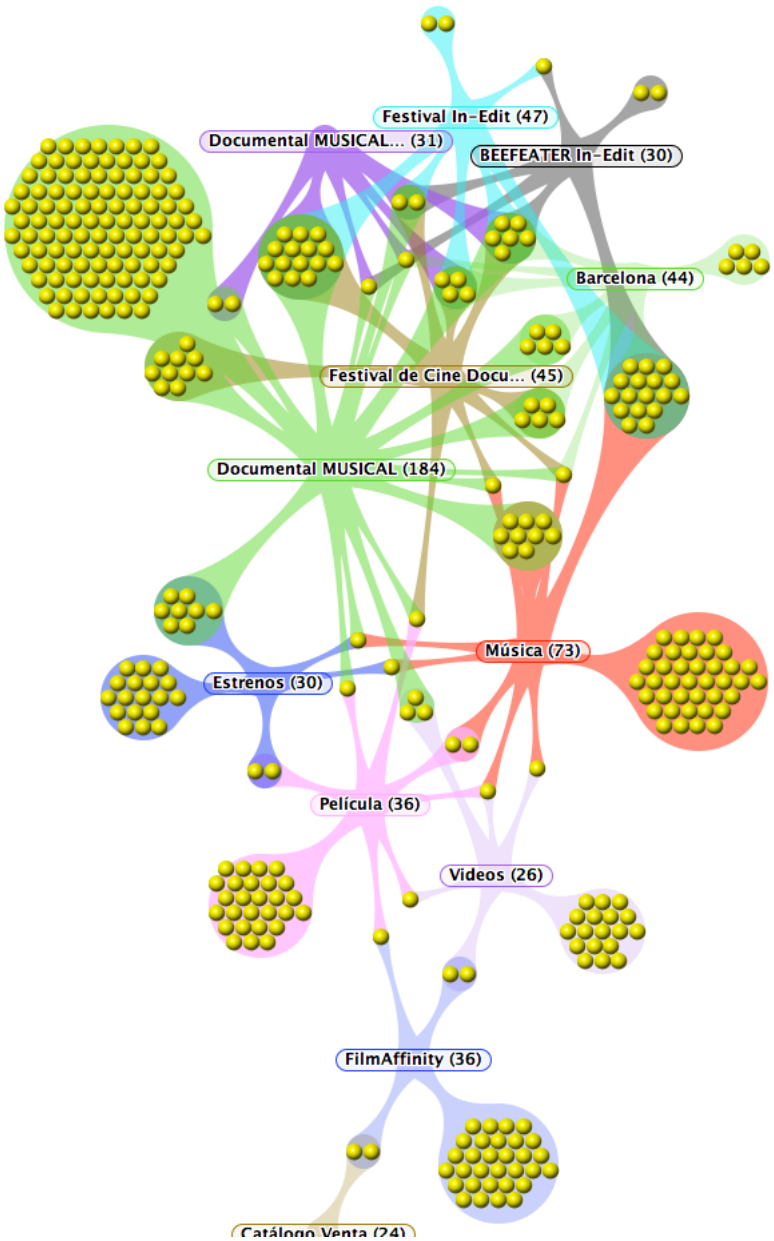

Fig. 5.6: Grupos más importantes para la categoría de musical

En la categoría de musical pueden verse mayormente sitios de festivales como In-edit, con un *cluster* específico http://www.in-edit.org/webapp/. Una curiosidad en este caso es que el mapa muestra mayor cercanía y mezcla entre los resultados, girando en torno al grupo relacionado a la búsqueda.

En forma complementaria, se realizaron las mismas búsquedas utilizando comillas en el buscador Google. Se tomaron los tópicos emergentes de cada categoría, y se analizaron las páginas incluídas en cada *cluster*. En este caso, por una limitación en su API gratuita sólo se pudieron obtener como máximo 100 resultados para cada búsqueda.

Resultados indexados a partir de las categorías en Google con comillas

Búsqueda	**Semillas**	**Docs. indexados**	**Contienen *cine***	**Contienen *cine* (%)**
cine documental	100	825	450	54,55 %
festival cine documental	95	784	427	54,46 %
biblioteca cine documental	0	0	0	0,00 %
documental político	100	828	516	**62,32** %
documental social	100	826	340	41,16 %
documental histórico	100	812	239	29,43 %
documental bélico	100	851	479	56,29 %
documental de biografías	43	178	108	**60,67** %
documental antropológico	100	828	308	37,20 %
documental científico	100	836	287	34,33 %
documental de investigación	100	855	225	26,32 %
doumental de propaganda política	72	110	73	**66,36** %
documental de denuncia	100	773	289	37,39 %
documental ecológico	100	794	341	42,95 %
documental musical	100	754	294	38,99 %
documental sobre artes	60	496	239	48,19 %
documental del yo	70	69	25	36,23 %

Si bien la cantidad de semillas que pueden obtenerse es limitada, se puede ver que en cuanto al porcentaje de inclusión del término **cine**, los rangos son bastante similares a Bing. Sin embargo, los mejores casos, destacados en negrita, tienen un porcentaje mayor que sus equivalentes para el buscador anterior. Esto podría explicarse, en principio, con la hipótesis de que Google da una mejor calidad de resultados.

Para evaluar esa hipótesis, se analizaron manualmente los grupos formados en cada caso. Se pudo observar que la mejora encontrada se debió en realidad a que la menor cantidad de semillas utilizadas lleva a que los documentos indexados provengan de estos (pocos) resultados, formándose *clusters* alrededor de pocos sitios. Una forma de automatizar esta validación podría ser calcular el porcentaje presente de semillas originales en los grupos de mayor tamaño.

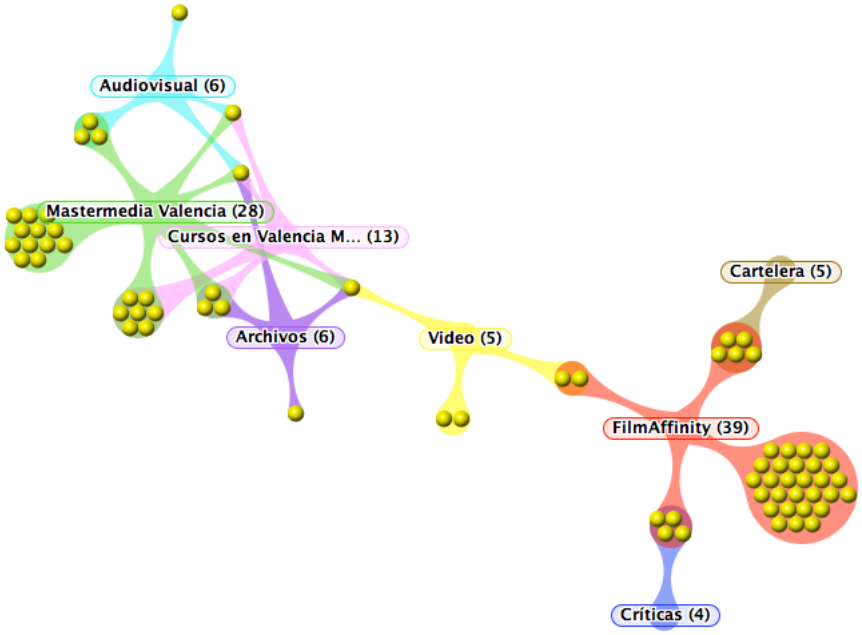

Fig. 5.7: Grupos más importantes para la categoría de propaganda política

Comparando con los grupos obtenidos para el buscador Bing, puede verse la aparición **FilmAffinity** nuevamente, sitio importante dentro del ecosistema de cine español. También aparecen, dentro de lo esperable, sitios de críticas, videos, cartelera, etc.

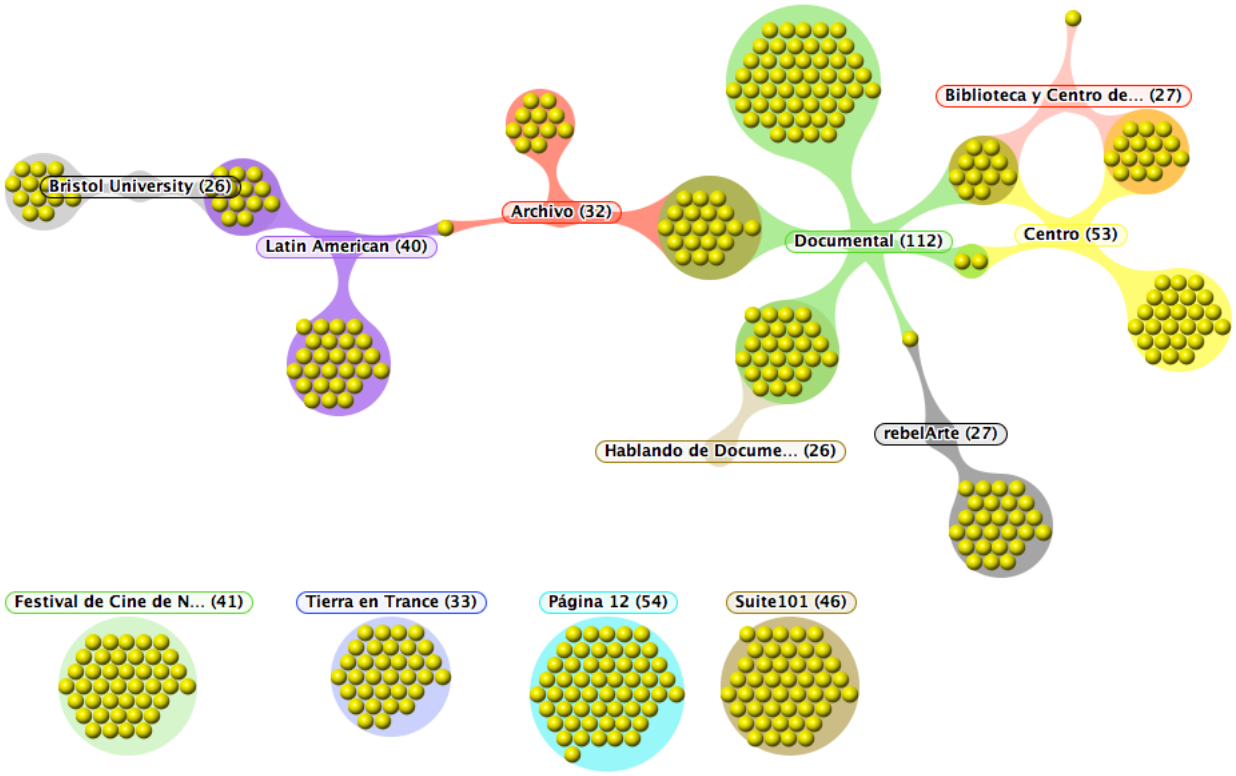

Fig. 5.8: Grupos más importantes para la categoría de documental político

El caso de documental político es bastante más interesante, ya que puede verse una gran cantidad de resultados, con grupos mucho más densos y más distantes que en los casos anteriores. Por un lado, se tiene una componente girando en torno al término **Documental**. Estos serían grupos interesantes, como rebelArte, sitio documental uruguayo.

Por otro, emergen cuatro grupos desconectados que se condicen con la realidad, ya que son lo que se consideraría ruido. Entre ellos, **Suite101**, sitio de publicidad, el diario **Página 12**, **Festival de Cine de No-Ficción** (sin menciones de documental) y **Tierra en Trance**, un blog general sobre cine latinoamericano.

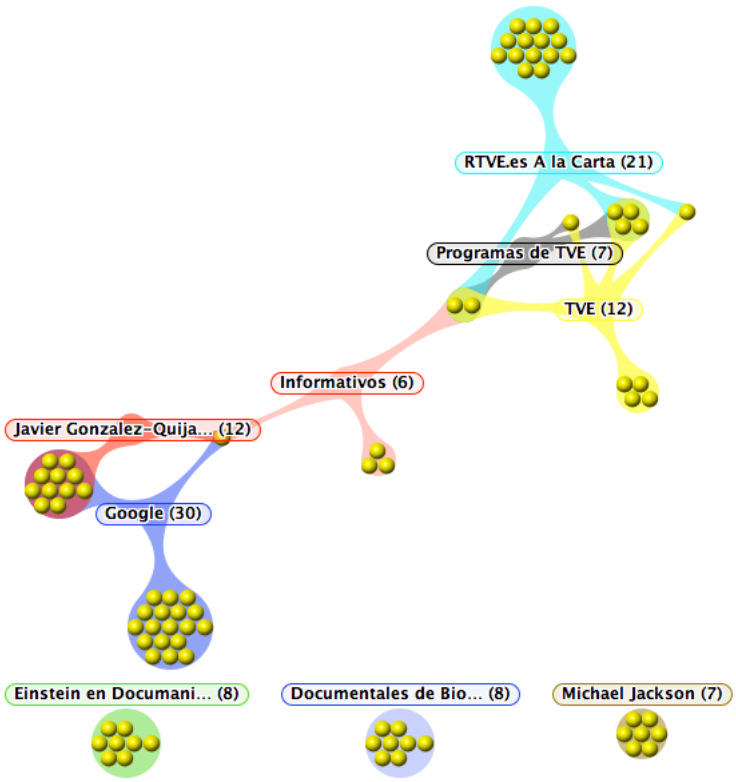

Fig. 5.9: Grupos más importantes para la categoría de biografías

Para la categoría de biografías resulta interesante la formación de grupos precisamente alrededor de algunos nombres propios como **Michael Jackson**, **Einstein** o **Javier González Quijano**. En los primeros dos casos, son efectivamente vínculos de sitios como documaníaTV con documentales biográficos. El tercero es bastante curioso y su conexión con el *cluster* Google es relevante, ya que se trata de múltiples perfiles y páginas de la red social Google+.

Intersección de resultados entre Google y Bing

Como experimento adicional, se decidió comparar entre sí los resultados obtenidos para cada buscador. La motivación fue la de cuantificar los sitios o páginas que en los grupos formados se veían en común para ambos buscadores. Se presenta una tabla que contiene los documentos compartidos en cada búsqueda y qué porcentaje representa del total obtenido de Google. Se tomó el total de Google como una forma de normalización, ya que como se mencionó, no pudieron obtenerse más de 100 resultados por una limitación de la API.

Búsqueda	**Docs. en común**	**Relación con Google(%)**
cine documental	6	6 %
festival cine documental	6	6,31 %
documental político	10	10 %
documental social	14	14 %
documental histórico	9	9 %
documental bélico	3	3 %
documental de biografías	4	9,31 %
documental antropológico	9	9 %
documental científico	11	11 %
documental de investigación	6	6 %
doumental de propaganda política	7	9,72 %
documental de denuncia	5	5 %
documental ecológico	7	7 %
documental musical	5	5 %
documental sobre artes	4	6,67 %
documental del yo	1	1,49 %

El hecho de que en cada categoría la intersección varíe y no supere el 14 % de los resultados obtenidos por Google muestra las diferencias entre ellos en cuanto a su contenido y su forma de obtener y ordenar los resultados.

En cuanto a los sitios en particular, el análisis individual sirvió para la obtención de nuevas semillas catalogadas como de interés, desde las cuales se pueda hacer un *crawling* interno del dominio con la seguridad de obtener documentos del vertical buscado.

Lamentablemente, el análisis individual mencionado no es escalable, con lo cual sería necesaria una automatización de este proceso para que el tamaño del índice de páginas alcanzado creciera. Para lograrlo, se necesitaría contar con un *corpus* de páginas web en el que ya se conociera de antemano si resulta de interés o no para el dominio. Con ese *corpus* se podría entrenar un algoritmo de clasificación de texto que reconociera automáticamente si los resultados arrojados por cada buscador deben ser indexados o no.

Todas las páginas obtenidas en este análisis manual fueron utilizadas con el objetivo de armar y ampliar dicho *corpus* en busca de un buen desempeño en algoritmos de clasificación.

5.2. Análisis de los grafos

En esta sección se presentan los resultados relacionados con el análisis realizado sobre grafos obtenidos de la etapa de *crawling*. Como se explicó en el capítulo 2, la estructura de páginas web puede modelarse como un grafo dirigido, donde un hipervínculo representa una arista desde la página que lo contiene hacia la página referenciada.

Para esta prueba, se tomó la estructura de páginas que se obtuvieron al realizar un proceso de *crawling* sobre los resultados de la búsqueda "cine documental" en Bing y Google. La profundidad o cantidad de saltos utilizada para este proceso fue de 2; es decir que si el buscador da como resultado una página, se toman todos los *links* salientes de esa página y a su vez los salientes de las páginas referenciadas por esos hipervínculos.

El objetivo de tomar sólo dos saltos fue el de analizar si los resultados de cada buscador tenían conexiones entre sí, pudiendo encontrar potencialmente sitios concentradores de información sobre cine documental.

Los grafos se presentan en gráficos donde cada punto representa una página, y las líneas, los hipervínculos que las vinculan. La distancia en el espacio no representa ninguna dimensión particular, sino que están dispuestos de forma aleatoria en el plano.

Para poder encontrar sitios relevantes o de autoridad, se aplicaron los algoritmos de PageRank y HITS, tomando como entrada las páginas obtenidas del crawling con dos niveles de profundidad. En el caso de HITS, esto se conoce como un *base set*, ya que se está expandiendo el conjunto inicial de resultados con sus páginas adyacentes.

Como último experimento, se tomaron los resultados devueltos por cada buscador y a partir de cada nodo en el grafo se calculó el tamaño de la componente conexa, siguiendo *links* salientes. Si con sólo dos saltos, partiendo de un sitio se llega a conectar una cantidad mayor de páginas en el grafo, podría indicar que ese sitio es un buen concentrador de información. Esto fue una hipótesis que se analizó manualmente en cada caso para entender la calidad de los resultados.

Cine Documental en Bing

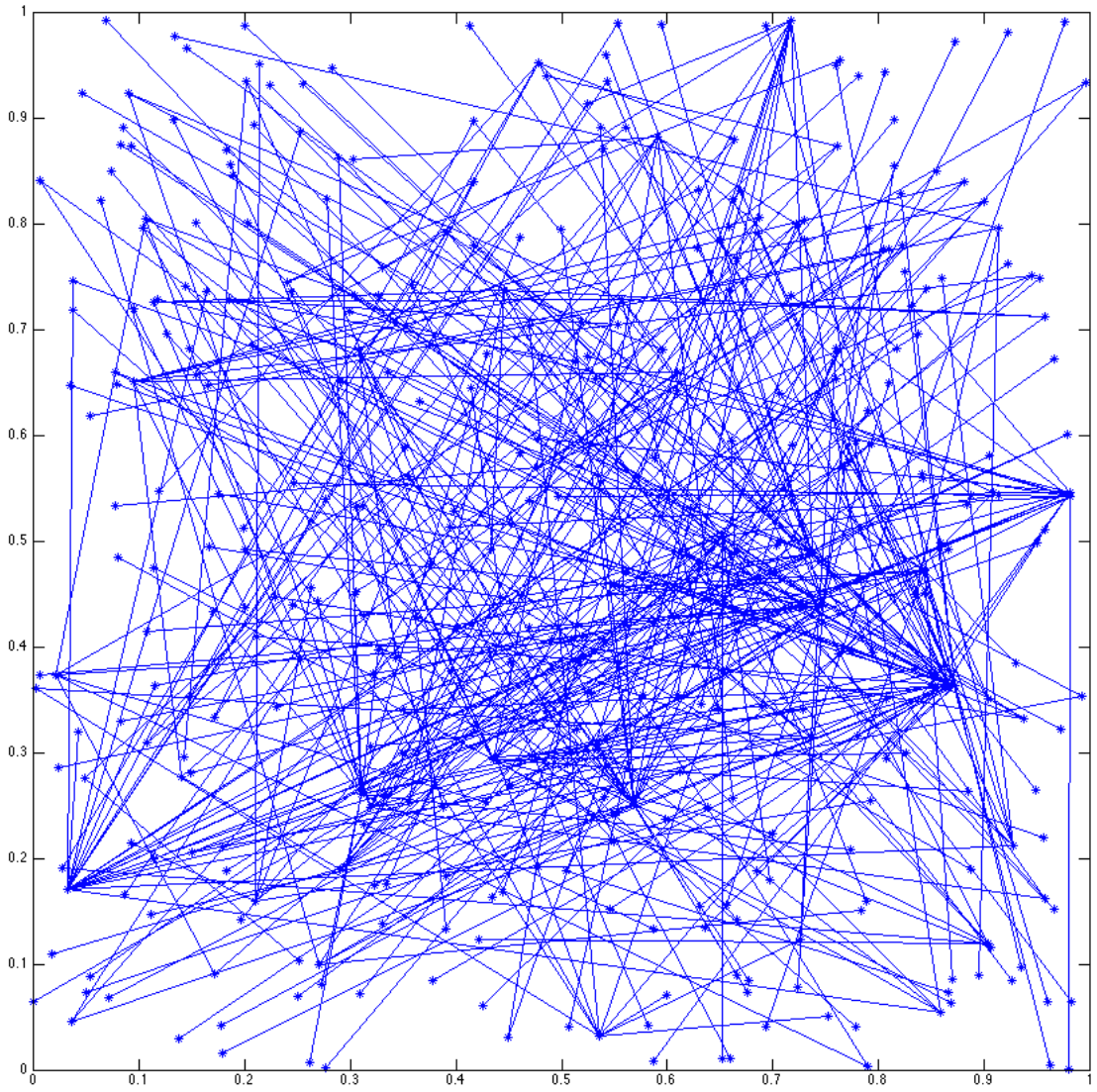

Fig. 5.10: Representación del grafo resultante del *crawling* de los resultados de la búsqueda "cine documental" en Bing

El tamaño total del grafo de la figura 5.10 es de 2668 vértices y 5475 aristas. Sin embargo, en el gráfico presentado no se ven representados los vértices aislados, ya que el propósito es el de observar la conectividad entre ellos. A priori, sin observar el gráfico pareciera ser una cantidad baja de aristas, dando un promedio de dos *links* por página, ya sea entrantes o salientes.

Visualizar este grafo sin información adicional no pareciera aportar mucha información, aunque da una idea inicial del espacio que se está recorriendo. Algo que se puede observar es la existencia de algunos puntos que parecieran concentrar vínculos, como puede verse en la parte media derecha o inferior izquierda del gráfico.

Identificar estos puntos puede ser de interés para utilizar como potenciales semillas

para recorrer e indexar su contenido, asumiendo por supuesto que éstos también lo son. Si un sitio es referenciado por muchos otros, la suposición es que se trata de un portal importante, el cual nos interesaría guardar y poder buscar entre sus páginas.

A continuación se muestran en orden descendente los 15 *links* de mayor autoridad para los algoritmos de PageRank y HITS dentro de esta búsqueda.

PageRank

1. http://gmpg.org/xfn/11
2. http://www.puntodevistafestival.com/
3. https://plus.google.com/118261503393207428058
4. http://blog.scribd.com/
5. http://revista.cinedocumental.com.ar/numeros
6. http://elcinedocumental.blogspot.com/
7. http://www.puntodevistafestival.com/blog
8. http://demo.vinaora.com/
9. http://www.cannabiscafe.net/foros
10. https://plus.google.com/116414028085828667311
11. http://www.google.com/jsapi
12. http://twitter.com/rebeldemule
13. http://www.facebook.com/pages/RebeldeMule-Comunidad-Alternativa/295464882228
14. http://www.fedochi.cl/
15. https://twitter.com/share

HITS Authorities

1. http://www.joomla.org/
2. http://miarroba.es/
3. http://www.uchile.cl/
4. http://info.yahoo.com/privacy/es/yahoo/
5. http://eardevol.wordpress.com/
6. http://www.revistas.uchile.cl/
7. http://www.icei.uchile.cl/
8. http://www.infonews.com/
9. http://es.wordpress.com/
10. http://www.us.es/
11. http://www.ecuadoradio.ec/
12. http://theme.wordpress.com/themes/imbalance2/
13. http://www.unavarra.es/
14. http://wordpress.org/
15. http://asistencia.foroactivo.com/

Viendo estos resultados, queda claro que confiar sólo en el *ranking* para el propósito buscado no es suficiente, ya que pueden encontrarse elementos considerados como ruido, como el caso de diarios o sitios que brindan un servicio (Joomla, Wordpress).

Sin embargo, si se analiza de forma asistida puede dar lugar a encontrar algunos portales interesantes. Tal es el caso de (5), (6) y (14) en PageRank. Del lado de HITS se ven sitios que tienen menos que ver con el tema de interés aunque como dato en común se observan algunas páginas de universidades como (3), (6), (7) y (13).

Lamentablemente, no se encontró otra forma de validar la calidad de estos resultados más que revisar cada página en forma manual y ver si cumplían con el dominio buscado, es decir que su contenido tuviera que ver con cine documental hispanoamericano.

Componentes conexas

A continuación se presentan las semillas encontradas con mayor tamaño de la componente conexa generada mediante su *crawling*.

URL	Tamaño de la comp. conexa
http://www.fedochi.cl/	59
http://francagonzalez.wix.com/documentales	27
http://www.naranjasdehiroshima.com/	16
http://www.docsdf.org/	14
http://www.slideshare.net/Jaime.1190/cine-documental	12
http://www.atlantidoc.com/	11
http://www.revista.cinedocumental.com.ar	6

Los resultados obtenidos para esta prueba fueron todos buenos concentradores de información, tratándose de sitios de festivales, revistas o simplemente información acerca del cine documental. Si bien parecen componentes chicas por la cantidad de páginas involucradas, es importante tener en cuenta que la prueba realizada se hizo utilizando sólo dos grados de profundidad en el *crawling*.

Nuevamente, el análisis fue hecho en forma manual, revisando cada uno de estos sitios y validando su contenido.

Cine Documental en Google

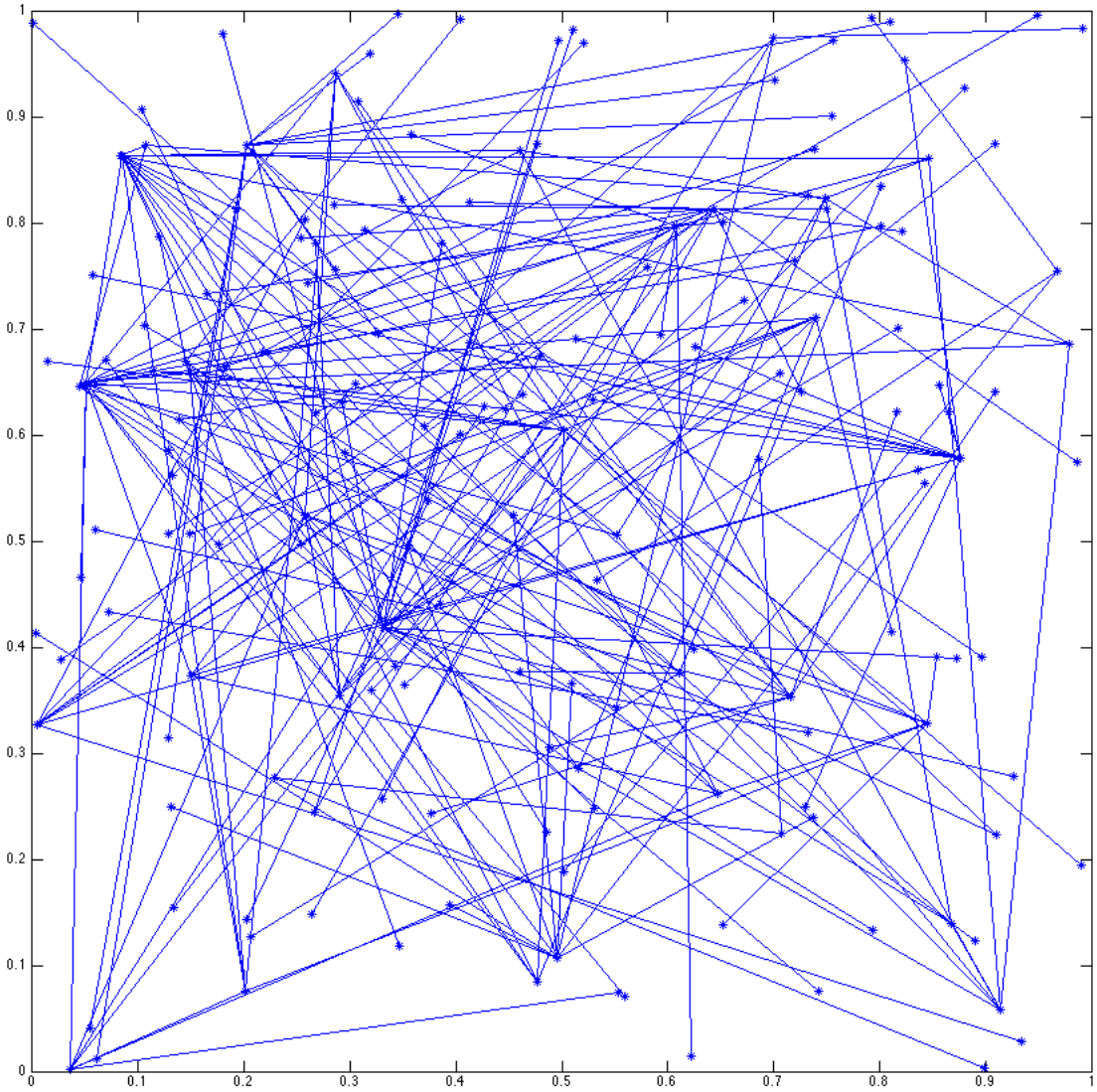

Fig. 5.11: Representación del grafo resultante del *crawling* de la búsqueda "cine documental" en Google

El tamaño total de este grafo es de 686 vértices y 1058 aristas. Al igual que en el caso de Bing, en el gráfico presentado no se ven representados los vértices aislados. Tomando un promedio de aristas por vértice, observamos un resultado similar de dos *links* en promedio por página, lo cual es bastante bajo.

A continuación se muestran en orden descendente los 15 *links* de mayor autoridad para los algoritmos de PageRank y HITS dentro de esta búsqueda. La entrada y los parámetros utilizados fueron los mismos que los descriptos para el caso de Bing.

PageRank

1. https://twitter.com/twitter
2. http://twitter.com/diasporg
3. https://diasporafoundation.org/
4. https://github.com/diaspora/diaspora
5. https://github.com/diaspora/diaspora/blob/2171bf2ba443085e3efa88a6ab1616ac43add335/Changelog.md
6. https://wiki.diasporafoundation.org/
7. http://status.twitter.com/
8. https://dev.twitter.com/
9. http://www.salaberlanga.com/
10. http://www.documentamadrid.com/
11. http://www.tallerimagen.com.ar/
12. https://business.twitter.com/
13. https://support.twitter.com/
14. https://twitter.com/cinedoc
15. http://twitter.com/twitterapi

HITS Authorities

1. http://www.cinepaint.org/more/
2. http://www.mivoz.cl/
3. http://www.unsam.edu.ar/
4. http://www.us.es/
5. http://www.radiounochile.cl/
6. http://busca.biobiochile.cl/author/cacosta/
7. http://wordpress.org/
8. http://www.unavarra.es/
9. http://www.fidba.com.ar/
10. http://www.avzeta.es/
11. http://www.blender.org/
12. http://www.flickr.com/photos/fedochi/
13. http://www.biobiochile.cl/
14. http://themeforest.net/item/mingle-multipurpose-wordpress-theme/235056
15. http://en.wikipedia.org/wiki/Mark_Achbar

A diferencia de lo sucedido para el buscador anterior, los resultados de PageRank son mayormente de ruido, incluyendo entre otros, muchos links de **twitter**. Para HITS no fueron mucho mejores, habiendo sólo uno o dos resultados rescatables como el caso de **fidba**.

Una posible explicación para este fenómeno es que al haber una considerable menor cantidad de resultados iniciales (por limitación en la API de Google), la probabilidad de caer en Twitter o sitios de mucho PageRank en toda la web (precisamente por su definición) es mucho más alta.

Teniendo más semillas iniciales, la probabilidad de encontrar conexiones entre ellas aumenta, mejorando para el caso de Bing la posibilidad de darle sentido al PageRank en la anterior búsqueda.

Al igual que para Bing, se calcularon las mejores semillas dado el tamaño de la componente conexa.

URL	Tamaño de la comp. conexa
http://www.fedochi.cl/	59
https://twitter.com/cinedoc	48
http://www.fidba.com.ar/	44
http://www.documentamadrid.com	39
http://www.puntodevistafestival.com	32
http://www.atlantidoc.com/	11
http://festivaledoc.org/	6

Nuevamente, tomar este *ranking* da buena calidad de resultados, tratándose de festivales y sitios afines al cine documental en distintos lugares de iberoamérica. En particular, 2 de los 7 coinciden con el resultado de Bing: **fedochi**, festival documental de chilloé y **atlantidoc**, festival de documental uruguayo.

5.3. Clasificación

Con el objetivo de automatizar el filtrado de documentos dentro de la etapa de *crawling*, se utilizaron algoritmos de clasificación automática de texto. Esta es una parte fundamental en este prototipo de buscador vertical, ya que es el componente que decide si una página web pertenece o no al dominio de interés.

Para llegar a realizar estas pruebas, se necesitó tener un índice disponible con diversos sitios y páginas potencialmente catalogadas como **cine documental** o como **ruido**, ya que son los textos que se utilizan como entrenamiento para los algoritmos.

En este caso, se utilizaron sitios que manualmente se fueron identificando a lo largo de las pruebas anteriores (ver Anexo A), realizando un *crawling* dentro de ellos, restringiendo únicamente páginas dentro del mismo dominio web.

En contraposición, se tomaron sitios web identificados como ruido puro, como el caso de diarios, redes sociales, etc. (Ver Anexo B).

Para las pruebas, se tomaron datos del *crawling* como entrenamiento y test en un porcentaje de 80-20. En todos los casos, las muestras con esos porcentajes fueron realizadas de forma aleatoria. Como datos de validación, se tomaron los 150 documentos etiquetados manualmente por la UNTREF, siendo estos el objetivo principal a optimizar para entender su criterio.

En cuanto a métricas, se tomó la matriz de confusión con los resultados y se calculó ***precision***, ***recall*** y F_1 ***measure***. Para este caso, la interpretación para cada uno fue:

- **Precision:** probabilidad de que un documento identificado como de interés, efectivamente lo sea.
- **Recall:** probabilidad de que un documento efectivamente de interés haya sido identificado como tal.
- F_1 **measure:** indicador combinado y ponderado de las dos anteriores.

Naïve Bayes - Entrenamiento/Test con 800/200 documentos

Test

Predicción

Real	**p**	**n**	**total**
p′	**TP** 103	**FN** 1	P′
n′	**FP** 74	**TN** 22	N′
total	P	N	

- **Precision:** 0.5819209
- **Recall:** 0.9903846
- F_1**-Measure:** 0.7330961

Cross Validation

Predicción

Real	**p**	**n**	**total**
p′	**TP** 81	**FN** 1	P′
n′	**FP** 68	**TN** 0	N′
total	P	N	

- **Precision:** 0.5436242
- **Recall:** 0.9878049
- F_1**-Measure:** 0.7012987

Fig. 5.12: Matrices de confusión para el clasificador Naïve Bayes con 800 documentos de entrenamiento y 200 de test

Analizando estos primeros resultados, se puede ver que este clasificador tiene un fuerte sesgo por la clase positiva, tanto en el test con documentos del mismo origen como con los de validación. Una razón atribuible a esto es la diversidad de documentos de la clase positiva, como festivales, portales de cine, revistas, etc.

Teniendo en cuenta que las *features* tomadas para clasificar son la frecuencia de las palabras, esta diversidad puede afectar negativamente, asignando alta probabilidad de ser positivo a un documento poco relacionado.

Una prueba realizada para mejorar esto fue eliminar de la matriz términos muy ralos, es decir palabras que ocurren en una porción pequeña de los documentos. De esta forma, no sólo se mejora potencialmente el desempeño del clasificador, sino que también se reduce el tamaño de la matriz, haciendo el proceso computacionalmente menos costoso.

Test

Real \ Predicción	p	n	total
p′	**TP** 82	**FN** 22	P′
n′	**FP** 34	**TN** 62	N′
total	P	N	

- **Precision:** 0.7068966
- **Recall:** 0.7884615
- F_1**-Measure:** 0.7454545

Cross Validation

Real \ Predicción	p	n	total
p′	**TP** 61	**FN** 21	P′
n′	**FP** 44	**TN** 24	N′
total	P	N	

- **Precision:** 0.5809524
- **Recall:** 0.7439024
- F_1**-Measure:** 0.6524064

Fig. 5.13: Matrices de confusión para el clasificador Naïve Bayes con 800 documentos de entrenamiento y 200 de test, eliminando términos ralos

Si se mira sólo el F_1 en la figura 5.13, podría decirse que los resultados al eliminar términos ralos son ligeramente peores. Sin embargo, es importante analizar la matriz y ver que el problema del sesgo positivo disminuyó, teniendo ahora muchos documentos clasificados como negativos. Esto se refleja en una mejora en la ***precision*** y disminución en el ***recall***.

Para cuantificar la disminución en el sesgo positivo, se puede utilizar la métrica conocida como $specificity = \frac{TN}{TN+FP}$, es decir la fracción de verdaderos negativos que el clasificador identificó como tales. En la figura 5.12, la $specificity$ para *test* es 0,229 y para *cross validation* es 0. En la figura 5.13 en cambio, la $specificity$ para *test* es 0,646 y para *cross validation* es 0,353. Se puede concluir entonces, que la capacidad de detectar documentos negativos o de no interés se vio casi triplicada para documentos de *test* y más aún en el caso de *cross validation* donde antes era nula.

En la prueba anterior, el ***recall*** era casi 1, ya que había sólo un falso negativo, al tener una fuerte tendencia a clasificar positivo. Al eliminar este problema, los resultados obtenidos tienen mejor tendencia a generalizar a nuevos casos, algo fundamental en cualquier problema de *machine learning*.

Como siguiente paso, se tomaron muestras de mayor tamaño para intentar mejorar las métricas de *precision*, *recall* y F_1.

Naïve Bayes sin términos ralos - Entrenamiento/Test con 12000/3000 documentos

Test

Real \ Predicción	p	n	total
p′	**TP** 1288	**FN** 250	P′
n′	**FP** 456	**TN** 1006	N′
total	P	N	

- **Precision:** 0.7385321
- **Recall:** 0.8374512
- F_1**-Measure:** 0.7848873

Cross Validation

Real \ Predicción	p	n	total
p′	**TP** 59	**FN** 23	P′
n′	**FP** 43	**TN** 25	N′
total	P	N	

- **Precision:** 0.5784314
- **Recall:** 0.7195122
- F_1**-Measure:** 0.6413043

Fig. 5.14: Matrices de confusión para el clasificador Naïve Bayes con 12000 documentos de entrenamiento y 3000 de test, eliminando términos ralos

Naïve Bayes sin términos ralos - Entrenamiento/Test con 16000/4000 documentos

Test

Real \ Predicción	**p**	**n**	**total**
p′	**TP** 1731	**FN** 291	P′
n′	**FP** 623	**TN** 1355	N′
total	P	N	

- **Precision:** 0.7353441
- **Recall:** 0.8560831
- F_1**-Measure:** 0.7911335

Cross Validation

Real \ Predicción	**p**	**n**	**total**
p′	**TP** 58	**FN** 24	P′
n′	**FP** 39	**TN** 29	N′
total	P	N	

- **Precision:** 0.5979381
- **Recall:** 0.7073171
- F_1**-Measure:** 0.6480447

Fig. 5.15: Matrices de confusión para el clasificador Naïve Bayes con 16000 documentos de entrenamiento y 4000 de test, eliminando términos ralos

A continuación se presentan cuatro gráficos con la variación de las métricas de *precision*, *recall* y f_1 en las pruebas de *test* y *cross validation* al aumentar la cantidad de documentos utilizados.

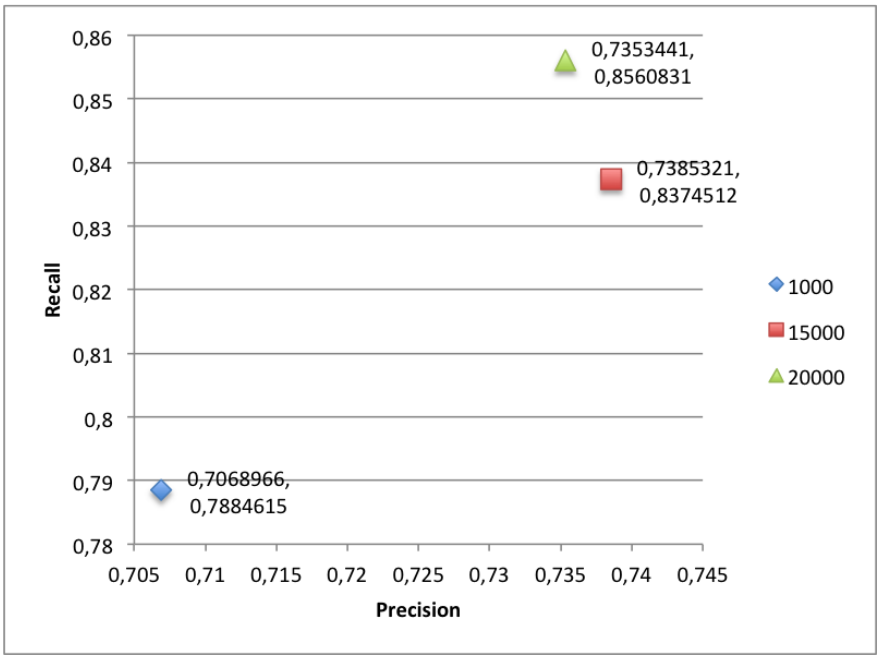

Fig. 5.16: Variación de *precision* y *recall* en la clasificación de *test* al aumentar la cantidad de documentos utilizados

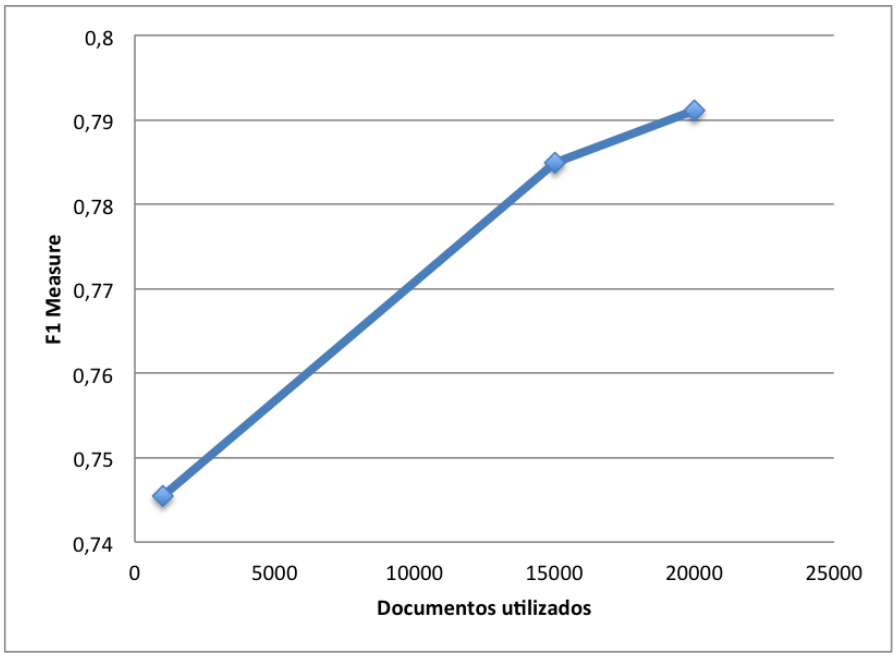

Fig. 5.17: Variación del F_1 en los documentos de *test* al aumentar la cantidad de documentos utilizados

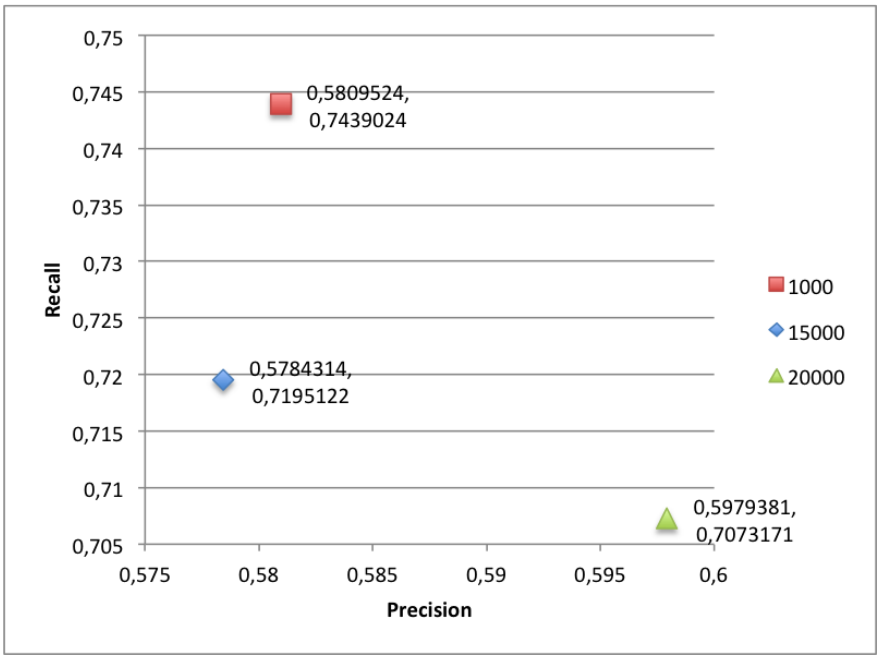

Fig. 5.18: Variación de *precision* y *recall* en la clasificación de *cross validation* al aumentar la cantidad de documentos utilizados

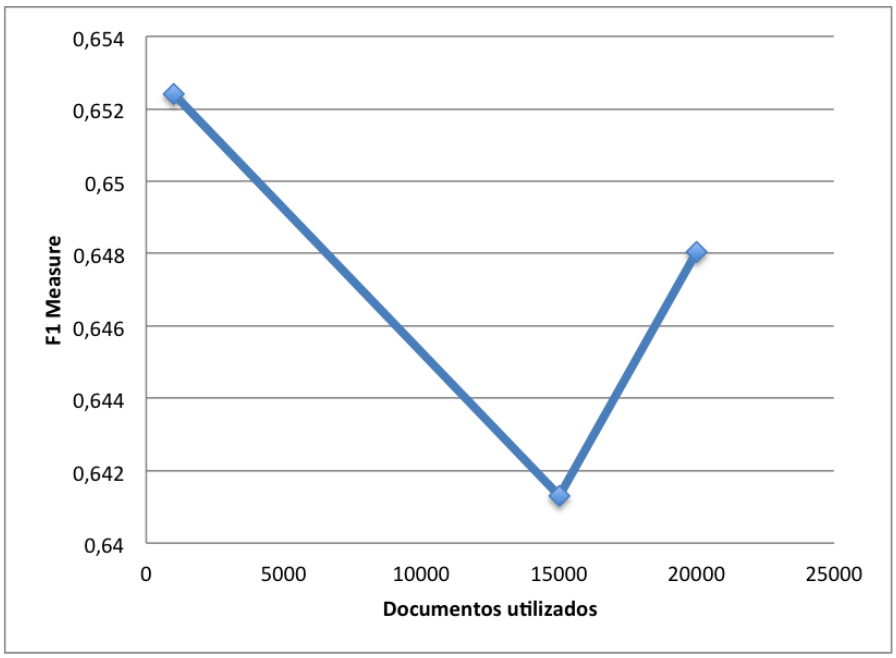

Fig. 5.19: Variación del F_1 en los documentos de *cross validation* al aumentar la cantidad de documentos utilizados

Como se puede ver en las figuras 5.14, 5.15 y 5.17, los resultados de *test* del clasificador mejoran mucho al aumentar el tamaño del corpus utilizado. Recordemos que para estas pruebas, los documentos clasificados son una porción no utilizada para entrenar de los documentos obtenidos de la etapa de *crawling*.

Sin embargo, para los datos utilizados como ***cross validation*** el F_1 empeora 1,6 % entre 1000 y 16000 documentos y 0,6 % entre 1000 y 20000 (figura 5.19), habiendo un retroceso del *recall* hacia el lado de la *precision* (figura 5.18). Una explicación para esto es que, por un lado no hay relación entre estos documentos (clasificados por la UNTREF) y los utilizados para entrenar. Por otro, tampoco se pudieron obtener mayor cantidad de éstos para validar, mientras que sí se aumentó el espacio de entrenamiento. Esto puede influenciar en tener mayor variedad de documentos involucrados en el clasificador entrenado, modificando las probabilidades asignadas para cada clase.

Lo ideal para esta prueba habría sido aumentar proporcionalmente la cantidad de documentos utilizados para validación, como sí se pudo hacer para la prueba de *test*, ya que de esta forma se evaluaría mejor cuán bien generaliza el algoritmo utilizado.

Otro algoritmo utilizado para la tarea de clasificación fue el de *Support Vector Machines*, abreviado SVM. Se presentan a continuación los resultados obtenidos.

SVM - Entrenamiento/Test con 800/200 documentos

Test

Real \ Predicción	p	n	total
p′	**TP** 95	**FN** 9	P′
n′	**FP** 11	**TN** 85	N′
total	P	N	

- **Precision:** 0.8962264
- **Recall:** 0.9134615
- F_1**-Measure:** 0.9047619

Cross Validation

Real \ Predicción	p	n	total
p′	**TP** 77	**FN** 5	P′
n′	**FP** 60	**TN** 9	N′
total	P	N	

- **Precision:** 0.5620438
- **Recall:** 0.9390244
- F_1**-Measure:** 0.7031963

Fig. 5.20: Matrices de confusión para el clasificador SVM con 800 documentos de entrenamiento y 200 de test

Comparando la figura 5.20 con la figura 5.12 (misma prueba para el clasificador Naïve Bayes), SVM da mejores resultados tanto en *test* como en *cross validation*. En particular, dentro de los documentos del *crawling* da excelentes resultados, clasificando mal sólo una pequeña porción. En el caso de la validación, la tendencia se asemeja a lo ocurrido con el clasificador bayesiano sin eliminar los términos ralos, dando una cantidad de resultados positivos mucho mayor a los negativos.

Al igual que en el caso anterior, se probó aumentar la cantidad de documentos involucrados en vistas de mejorar la *performance* del clasificador.

SVM - Entrenamiento/Test con 16000/4000 documentos

Test

Real \ Predicción	p	n	total
p′	**TP** 2018	**FN** 30	P′
n′	**FP** 206	**TN** 1746	N′
total	P	N	

- **Precision:** 0.9073741
- **Recall:** 0.9853516
- F_1**-Measure:** 0.9447566

Cross Validation

Real \ Predicción	p	n	total
p′	**TP** 73	**FN** 10	P′
n′	**FP** 58	**TN** 10	N′
total	P	N	

- **Precision:** 0.5572519
- **Recall:** 0.8795181
- F_1**-Measure:** 0.6822430

Fig. 5.21: Matrices de confusión para el clasificador SVM con 16000 documentos de entrenamiento y 4000 de test

Al aumentar la cantidad de documentos involucrados en el entrenamiento, puede verse en la figura 5.21 la misma tendencia que en Naïve Bayes. Los resultados sobre los datos propios mejora notablemente, obteniendo un F_1 score superior al anterior en un 4,4 %. Para los datos de validación ocurre también que aumentar el corpus sin aumentar los documentos a validar da peores resultados (reduce el F_1 en 2,9 %), ya que posiblemente se estén introduciendo muchos documentos a ambas clases que puedan influenciar como ruido a las decisiones del clasificador.

El último algoritmo utilizado fue el de máxima entropía o *maximum entropy* en inglés.

MaxEntropy - Entrenamiento/Test con 800/200 documentos

Test

Real \ Predicción	**p**	**n**	**total**
p′	**TP** 101	**FN** 3	P′
n′	**FP** 2	**TN** 94	N′
total	P	N	

- **Precision:** 0.9805825
- **Recall:** 0.9711538
- F_1**-Measure:** 0.9758454

Cross Validation

Real \ Predicción	**p**	**n**	**total**
p′	**TP** 52	**FN** 30	P′
n′	**FP** 49	**TN** 20	N′
total	P	N	

- **Precision:** 0.5148515
- **Recall:** 0.6341463
- F_1**-Measure:** 0.5683060

Fig. 5.22: Matrices de confusión para el clasificador MaxEntropy con 800 documentos de entrenamiento y 200 de test

MaxEntropy - Entrenamiento/Test con 16000/4000 documentos

Test

Real \ Predicción	p	n	total
p′	**TP** 2006	**FN** 42	P′
n′	**FP** 58	**TN** 1894	N′
total	P	N	

- **Precision:** 0.9718992
- **Recall:** 0.9794922
- F_1**-Measure:** 0.9756809

Cross Validation

Real \ Predicción	p	n	total
p′	**TP** 49	**FN** 34	P′
n′	**FP** 43	**TN** 25	N′
total	P	N	

- **Precision:** 0.5326087
- **Recall:** 0.5903614
- F_1**-Measure:** 0.56

Fig. 5.23: Matrices de confusión para el clasificador MaxEntropy con 16000 documentos de entrenamiento y 4000 de test

Para el clasificador de máxima entropía o *MaxEntropy*, los resultados obtenidos son, por un lado los mejores y excelentes para el conjunto de *test*, pero considerablemente peores para los de validación.

Se puede ver en los casos de *test* de las figuras 5.22 y 5.23 que la cantidad mal clasificada (fuera de la diagonal en la matriz de confusión) es realmente muy baja. Esto no se condice con lo que ocurre en el caso de *cross validation*. Este fenómeno se conoce como *overfitting* o sobre ajuste, dado que el clasificador funciona muy bien para datos propios, pero falla en generalizar a datos nuevos.

Comparación

Para propósitos prácticos, el clasificador de máxima entropía no es de utilidad ya que la tarea más importante en el caso de este trabajo es poder identificar y clasificar nuevas páginas a través del *crawling*. Como se dijo, los resultados de este clasificador se ven afectados por sobreajuste, fallando en generalizar. Respecto a Naïve Bayes y SVM, la validación dio mejores resultados para SVM. Sin embargo, elegir uno u otro ameritaría otras pruebas, ya que la diferencia no es tan grande como para justificar el costo computacional que tiene entrenar un clasificador por SVM ($O(N^3)$), comparado con el clasificador Bayesiano ($O(N)$), siendo N la cantidad de documentos.

6. CONCLUSIONES Y TRABAJO FUTURO

El problema tratado es muy complejo: el criterio para identificar si el contenido de un sitio trata o no de cine documental puede ser muy subjetivo.

A lo largo de este trabajo se hizo un estudio exploratorio, utilizando distintas técnicas de *Information Retrieval* y *Web Mining*. En primer lugar, se intentó comprender qué tipo de páginas se estaban buscando y con qué herramientas se contaba para ello. Esto llevó a una base de sitios y páginas con los cuales trabajar, permitiendo agruparlas para encontrar tópicos y utilizarlas como corpus para entrenar un clasificador.

Este análisis llevó a encontrar que el espacio de sitios en los cuales se planteaba interés era considerablemente chico para los propósitos planteados, sumado a una poca conectividad entre ellos. Dentro de lo que se consideró interés puro, se obtuvo un índice de casi 23.000 documentos.

Por otra parte, se desarrolló un prototipo de buscador web que respondiera *queries* o consultas a partir de la base de páginas obtenida en el paso anterior. Además de las funcionalidades básicas de un típico buscador, se agregó la posibilidad de marcar como positivo o negativo un resultado, buscando una retroalimentación al clasificador que permitiera mejorar los resultados a través del tiempo.

Uno de los resultados obtenidos tiene que ver con una mejora significativa en los resultados del clasificador al aumentar la cantidad de documentos del entrenamiento. Las mejoras anteriores, sumadas a una mayor cantidad de semillas para hacer *crawling*, llevarían a mejorar los resultados ofrecidos por este buscador.

Al discutir sobre el prototipo con la UNTREF, principal interesado en el proyecto, se identificó que el interés en una página no es global, sino que muchas veces depende de la búsqueda realizada.

Si el interés depende de la búsqueda, aplicar un clasificador en la etapa de *crawling* produce indexar resultados que para el usuario pueden no interesar. En consecuencia, se pensó en otra estrategia como usar otro clasificador que se ejecute en el momento de la búsqueda, teniendo en cuenta las palabras clave utilizadas.

Otra mejora pensada tiene que ver con armar más clases o tópicos dentro de lo que se considera interés. Parte de lo analizado tiene que ver con que difiere mucho la estructura de un blog de una página de un festival, o éstas con páginas de poco contenido en texto como las que muestran videos. Algunas de estas categorías serían:

- Videos
- Festivales
- Críticas
- Blogs
- Sitios de películas
- Fichas técnicas

Estas categorías permitirían un filtro adicional al usuario, pudiendo marcar no sólo el interés o no en un resultado, sino también una o más de estas clases.

Luego de todo el estudio y análisis realizado, se puede concluir que la construcción de un buscador vertical sobre cine documental hispanoamericano es posible, aunque se encontraron limitaciones para avanzar en los componentes implementados. El espacio web que pudo encontrarse a partir de buscadores tradicionales fue muy chico como para alcanzar todos los sitios deseados. Muchas de las tareas realizadas para descubrir sitios de interés fueron hechas de forma manual, limitando la escalabilidad de estos procesos. La única opción posible para automatizarlos sería contar con un clasificador que pueda distinguir sitios del dominio, y utilizarlo en un *crawling* indiscriminado de toda la web. Ese es el componente principal de la arquitectura planteada, para el cual se necesitaría contar con un *corpus* más grande de páginas web etiquetadas manualmente que pudieran mejorar la *performance* lograda en este trabajo.

Apéndice

A. ANEXO I: SITIOS UTILIZADOS COMO SEMILLAS

- http://www.documaniatv.com/
- http://revista.cinedocumental.com.ar/numeros/
- http://www.adndoc.com.ar/
- http://www.fidba.com.ar/
- http://www.docacine.com.ar/
- http://www.uhu.es/cine.educacion/cineyeducacion/cinedocumental.htm
- http://www.cine.ar/contenidos/17-Cine-documental/
- http://rdidocumental.com.ar/
- http://www.docupolis.org/
- http://www.cinedocumental.es/
- http://cinedocumentalcaromg.blogspot.com.ar/
- http://www.elcinedocumental.blogspot.com.ar/
- http://cinedocumental-carolina.blogspot.com.ar/
- http://www.nochedecine.com/tag/cine-documental/
- http://tercer-ojo.com/
- http://cine-invisible.blogs.fotogramas.es/category/cine-documental-2/
- http://webstore-cinedocumental.blogspot.com.ar/
- http://documentales.blogspot.com.ar/
- http://www.atlantidoc.com/
- http://www.puntodevistafestival.com/
- http://docma.es/
- http://www.docsbarcelona.com/es/index.php?edicion=2013
- http://www.documentales-online.com/
- http://www.cinenacional.com/search/node/documental
- http://ernestoardito.wordpress.com/

B. ANEXO II: SITIOS EXCLUIDOS, UTILIZADOS COMO FUENTE DE RUIDO

- http://www.clarin.com
- http://www.pagina12.com.ar
- http://www.lanacion.com.ar
- http://www.facebook.com
- http://www.youtube.com
- http://es.wikipedia.org
- http://www.facebook.com
- http://www.mercadolibre.com.ar

Bibliografía

[1] B. Liu, *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006.

[2] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. New York, NY, USA: Cambridge University Press, 2008.

[3] R. A. Baeza-Yates and B. A. Ribeiro-Neto, *Modern Information Retrieval - the concepts and technology behind search, Second edition*. Pearson Education Ltd., Harlow, England, 2011.

[4] R. Feldman and J. Sanger, *Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. New York, NY, USA: Cambridge University Press, 2006.

[5] S. Chakrabarti, *Mining the Web: Discovering Knowledge from HyperText Data*. Science & Technology Books, 2002.

[6] I. H. Witten, A. Moffat, and T. C. Bell, *Managing Gigabytes (2Nd Ed.): Compressing and Indexing Documents and Images*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999.

[7] C. Aggarwal and C. Zhai, *Mining Text Data*. Springer-Verlag New York Inc, 2012.

[8] M. Chau and H. Chen, "A machine learning approach to web page filtering using content and structure analysis," *Decis. Support Syst.*, vol. 44, pp. 482–494, Jan. 2008.

[9] M. Chau, H. Chen, J. Qin, Y. Zhou, Y. Qin, W.-K. Sung, and D. McDonald, "Comparison of two approaches to building a vertical search tool: A case study in the nanotechnology domain," in *Proceedings of the 2Nd ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '02, (New York, NY, USA), pp. 135–144, ACM, 2002.

[10] G. Almpanidis, C. Kotropoulos, and I. Pitas, "Combining text and link analysis for focused crawling-an application for vertical search engines," *Inf. Syst.*, vol. 32, pp. 886–908, Sept. 2007.

[11] G. Almpanidis, C. Kotropoulos, and I. Pitas, "Focused crawling using latent semantic indexing: an application for vertical search engines," in *Proceedings of the 9th European Conference on Research and Advanced Technology for Digital Libraries*, ECDL'05, (Berlin, Heidelberg), pp. 402–413, Springer-Verlag, 2005.

[12] S. Osinski, J. Stefanowski, and D. Weiss, "Lingo: Search results clustering algorithm based on singular value decomposition," in *Intelligent Information Systems*, pp. 359–368, 2004.

[13] M. W. Berry, S. Dumais, G. O'Brien, M. W. Berry, S. T. Dumais, and Gavin, "Using linear algebra for intelligent information retrieval," *SIAM Review*, vol. 37, pp. 573–595, 1995.

[14] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," in *Proceedings of the Seventh International Conference on World Wide Web 7*, WWW7, (Amsterdam, The Netherlands, The Netherlands), pp. 107–117, Elsevier Science Publishers B. V., 1998.

[15] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM*, vol. 46, pp. 604–632, Sept. 1999.