Query-Sets: Using Implicit Feedback and Query Patterns to Organize Web Documents

Barbara Poblete
Web Research Group
University Pompeu Fabra
Barcelona, Spain
barbara.poblete@upf.edu

Ricardo Baeza-Yates
Yahoo! Research &
Barcelona Media Innovation Center
Barcelona, Spain
ricardo@baeza.cl

ABSTRACT

In this paper we present a new document representation model based on implicit user feedback obtained from search engine queries. The main objective of this model is to achieve better results in non-supervised tasks, such as clustering and labeling, through the incorporation of usage data obtained from search engine queries. This type of model allows us to discover the motivations of users when visiting a certain document. The terms used in queries can provide a better choice of features, from the user's point of view, for summarizing the Web pages that were clicked from these queries. In this work we extend and formalize as *query model* an existing but not very well known idea of *query view* for document representation. Furthermore, we create a novel model based on *frequent query patterns* called the *query-set model*. Our evaluation shows that both *query-based* models outperform the vector-space model when used for clustering and labeling documents in a website. In our experiments, the query-set model reduces by more than 90% the number of features needed to represent a set of documents and improves by over 90% the quality of the results. We believe that this can be explained because our model chooses better features and provides more accurate labels according to the user's expectations.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Clustering
; H.2.8 [**Information Systems**]: Data Mining
; H.4.m [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms, Experimentation, Human Factors

Keywords

Feature Selection, Labeling, Search Engine Queries, Usage Mining, Web Page Organization

1. INTRODUCTION

As the Web's contents grow, it becomes increasingly difficult to manage and classify its information. Optimal organization is especially important for websites, for example, where classification of documents into cohesive and relevant topics is essential to make a site easier to navigate and more intuitive to its visitors. The high level of competition in the Web makes it necessary for websites to improve their organization in a way that is both automatic and effective, so users can reach effortlessly what they are looking for. Web page organization has other important applications. Search engine results can be enhanced by grouping documents into significant topics. These topics can allow users to disambiguate or specify their searches quickly. Moreover, search engines can personalize their results for users by ranking higher the results that match the topics that are relevant to users' profiles. Other applications that can benefit from automatic topic discovery and classification are human edited directories, such as DMOZ[1] or Yahoo![2]. These directories are increasingly hard to maintain as the contents of the Web grow. Also, automatic organization of Web documents is very interesting from the point of view of *discovering new interesting topics*. This would allow to keep up with user's trends and changing interests.

The task of automatically clustering, labeling and classifying documents in a website is not an easy one. Usually these problems are approached in a similar way for Web documents and for plain text documents, even if it is known that Web documents contain richer and, sometimes, implicit information associated to them. Traditionally, documents are represented based on their text, or in some cases, also using some kind of structural information of Web documents.

There are two main types of structural information that can be found in Web documents: HTML formatting, which sometimes allows to identify important parts of a document, such as title and headings, and link information between pages [15]. The formatting information provided by HTML is not always reliable, because tags are more often used for styling purposes than for content structuring. Information given by links, although useful for general Web documents, is not of much value when working with documents from a particular website, because in this case, we cannot assume that this data has any *objectiveness*, i.e.: any information extracted from the site's structure about that same site, is a reflection of the webmaster's criteria, which provides no warranty of being thorough or accurate, and might be completely arbitrary. A clear example of this, is that many websites that have large amounts of contents, use some kind of content management system and/or templates, that give practically the same structure to all pages and links within

WWW 2008, April 21–25, 2008, Beijing, China.
ACM 978-1-60558-085-2/08/04.

[1] http://www.dmoz.org
[2] http://www.yahoo.com

the site. Also, structural information does not reflect what users find more interesting in a Web page. It only reflects what webmasters find interesting.

Since neither content or structural data seem to be complete information sources for the task of clustering, labeling and classification of Web documents, we propose the incorporation of *usage data*, obtained from the usage logs of the site. In particular, we suggest to use the information provided by user queries in search engines. This information is very easy to retrieve, from the *referer*[3] field of the usage log, and provides a very precise insight into what user's motivations are for visiting certain documents. Terms in queries can be used to describe the topic that users were trying to find when they clicked on a document. These queries provide implicit user feedback that is very valuable. For example, we can learn that if many users reach a document using certain keywords, then it is very likely that the important information in this document can be summarized by those keywords.

Following this motivation, we propose a different document representation model, mainly for clustering and labeling, but that can also be used for classification. Traditional models for document representation use the notion of a *bag of words*, the *vector space model* being the most well known example of this. Our approach is based on these models but selects features using what seems more appropriate to refer to as a *bag of query-sets* idea. The representation is very simple, yet intuitive, and it reduces considerably the number of features for representing the document set. This allows to use all of the document features for clustering, and in our experiments this shows to be very effective for grouping and labeling documents in a website.

The main contributions of this paper are,

- to present two document models which use implicit user feedback from search queries:

    1. a model that formalizes and extends the previously existing concept of *query view* [9] into a more general *query document model*,
    2. a *new* document representation based *only on frequent sets of clicked queries*, the *query-set model*, that improves the previous model,

- propose a new methodology based in known algorithms for clustering and labeling Web documents, using the query-set model. This model can be applied to organize documents within a website, general Web documents and search engine results.

- We also present an initial experimental evaluation to corroborate our models.

This paper is organized as follows: Section 2 presents related work. Section 3 describes the query-based document models. Section 4 discusses the evaluation and results, and finally in Section 5 we present conclusions and future work.

2. RELATED WORK

Web mining [31] is the process of discovering knowledge, such as patterns and relations, from Web data. Web mining generally has been divided into three main areas: *content mining*, *structure mining* and *usage mining.* Each one of these areas are associated mostly, but not exclusively, to these three predominant types of data found in the Web:

Content: The *real* data that the document was designed to give to its users. In general this data consists mainly of text and multimedia.

Structure: This data describes the organization of the content within the Web. This includes the organization inside a Web page, internal and external links and the website hierarchy.

Usage: This data describes the use of a website or search engine, reflected in the Web server's access logs, as well as in logs for specific applications.

Web usage mining has generated a great amount of commercial interest [12]. There is an extensive list of previous work using Web mining for improving websites, most of which focuses on supporting adaptive websites [21] and automatic personalization based on Web Mining [20]. Amongst other things, using analysis of frequent navigational patterns, document clustering, and association rules, based on the pages visited by users, to find interesting rules and patterns in a website [8, 30, 11, 19].

Web usage mining is a valuable way of discovering data about Web documents, based on the information provided implicitly by users. For instance, [36] combines many information sources to solve navigation problems in websites. The main idea is to cluster pages based on their link similarities using visitor's navigation paths as weights to measure semantic relationships between Web pages. In [28] they create implicit links for Web pages by observing different documents clicked by users from the same query. They use this information to enhance Web page classification based on link structure.

Document clustering, labeling, automatic topic discovery and classification, have been studied in previous work with the purpose of organizing Web content. Organizing Web pages is very useful withing two areas, *search engine enhancement*, and *improving hierarchical organization of documents.* Search engines can benefit greatly from effective organization of their search results into clusters. This allows users to navigate into relevant documents quickly [33]. In a similar way, automatic organization of Web content can improve human edited directories.

In general, all existing clustering techniques must rely on four important components [16]: a data representation model, a similarity measure, a cluster model, and a clustering algorithm that uses the data model and this similarity measure. In particular, Web documents pose three main challenges for clustering [7]:

- very high dimensionality of data,
- very large size of collections, and
- the creation of understandable cluster labels.

Many document clustering and classification methods are based on the *vector space* document model. The vector space model [26] represents documents as vectors of terms, in an Euclidean space. Each dimension in the vector represents a term from the document collection, and the value of

[3] Although this is a misspelling of *referrer*, this is the term used in the HTTP specifications.

each coordinate is weighted by the frequency in which that term appears in the document. The vector representations are usually normalized according to *tf-idf* weighting scheme used in Information Retrieval [2]. The similarity between documents is calculated using some measure, such as the cosine similarity. The vector space model does not analyze the co-occurrence of terms inside a document or any type of relationship amongst words.

There are several approaches that try to improve the vector space model with the purpose of improving Web document clustering. In general, they are based in discovering *interesting associations between words in the text of the document*. In [16] they propose a system for Web clustering based on two key concepts: the use of weighted phrases as features for documents, and an incremental clustering of documents that watches the similarity distribution inside each cluster. A similar notion is developed in [7] where they define a document model based on frequent terms obtained from all the words in a document, aiming at reducing the dimensionality of the document vector space. In [23] they also use term sets in what they call a *set-based model*, which is a technique for computing term weights for index terms in Information Retrieval that uses sets of terms mined by using association rules on the full text of documents in a collection. Another type of feature selection from document contents is done by using *compound words* [34] provided by WordNet[4]. In [32] they use a document model for clustering based on the extraction of relevant keywords for a collection of documents. The keyword with the highest score within each cluster is used as the label. The application described in [14] also uses extraction of keywords based on frequency and techniques such as using a *inlinks* and *outlinks*. All of these methods use the information provided by the contents of Web pages, or their structure, but they *do not incorporate actual usage information* (i.e., implicit user feedback) into their models.

Implicit user feedback, such as *clicked answers for queries submitted to search engines* are a valuable tool for improving websites and search results. Most of the work using queries has been focused on enhancing website search [35] and to make more effective global Web search engines [3, 17, 29, 25]. Also, queries have been studied to improve clustering of Web documents. This idea, to the best of our knowledge, has only been considered previously in [6], [9] and [24]. In [6] they represent a query log as a bipartite graph, in which queries are on one side of the graph and URLs clicked from queries are on the other. They use an agglomerative clustering technique to group similar queries and also similar documents. This algorithm is *content-ignorant* as it makes no use of the actual content of the queries or documents, but only how they co-occur within the click through data. In [9] they present the idea of using a *query view* for document representation, which mines queries from a site's internal search engine as features to model documents in a website. The goal of this research was to improve an on-line customer support system. The third work, described in [24] introduces a document representation called *query vector model*. In this approach they use query logs to model documents in a search engine collection to improve document selection algorithms for parallel information retrieval systems. The model represents each document as a vector of queries (extracted from the search engine query log) weighted by the search engine rank of the document for each particular query in the feature space. Although this work is similar to [9] and [6], due to the fact that both base their clustering models on queries, they differ in the fact that in [24] the query log is *only used to extract queries* but does not use the click through information of the documents clicked for each query. Whether or not users clicked on a particular document from the result set of a query, is not taken into account for the document model and by doing so [24] only considers the search engine rank algorithm output, even if no users considered the results as appropriate for their query.

User feedback to search engine queries has also been considered in other document representation models. In [25] they rank search results using feature vectors for documents, which are learned from *query chains* or queries with similar information needs. Using a learning algorithm they incorporate different types of preference judgments that were mined from query chains into their document model. This model takes advantage of user query refinement. Another approach that considers user feedback to search queries is [33]. In [33] they classify search results into categories discovered using a pseudo document representation of previous related queries to the input query. The pseudo representation of related queries incorporates user feedback by including the text from snippets of previously clicked documents for the related queries. In [33] the terms of queries are considered as a *brief summary* of its pseudo document. The idea that queries can be good descriptors for their clicked documents is also discussed in detail in the query mining model described in [5].

Our work is based on idea that search queries and their clicked results provide valuable user feedback about the relevance of documents to queries. In this way, our work is similar to [6], [25] and [33]. Nevertheless, our approach differs because we consider that the *queries from which documents are clicked* are *good summaries* of user's intent when viewing a document. Hence, in our model we use global search engine queries (and not only internal searches as in [9]) as surrogate text for documents and choose the document's features from the terms of the queries from which it was clicked from. Our model extends this notion by mining frequent sets from queries in a similar way to [7] and [23], with the difference that they mine patterns from the original full text of the document (and not considering user feedback from queries).

3. DOCUMENT CLUSTERING AND LABELING

Search engines play a key role in finding information on the Web. They are responsible directly, or indirectly, for a large part of the traffic in websites, and documents visited as a result of queries compose the actual *visible* pages of the site. Furthermore, we can say that for many websites the only important documents are the ones that are reachable from search engines. Due to the significance of queries for aiding users to find content on the Web, in this section we discuss the issue of using queries to understand and model documents.

The traditional *vector model* representation for documents, although it can be used to model Web documents, lacks a proper understanding on what are the most relevant topics

[4] http://wordnet.princeton.edu

of each document from the *user's point of view* and which are the best words (or features) for summarizing these topics. It is important to note that a visitor's perception of what is relevant in a document is not necessarily the same as the site author's perception of relevance. Thus, a webmaster's organization of documents of a website can be completely different of what user's would expect. This would make navigation difficult and documents hard to find for visitors, see [22]. Also, what users find interesting in a Web page does not always agree with the most descriptive features of a document according to a *tf-idf* type of analysis. This is, the most distinctive words of a document are not always the most descriptive features of its vector space representation.

For modeling Web documents, we believe that it is better to represent documents using queries instead of their actual text contents, i.e.; using queries as surrogate text. We extend and modify previous work based on the intuition that queries from which visitors clicked on Web documents, will make better features for the purposes of automatically grouping, labeling, and classifying documents. By associating *only* queries to the documents that were clicked from them, we bring implicit user feedback into the document representation model. By using clicked pages we are trusting the relevance judgements of the real users and not the search engines judgements (which may be different for different engines), and hence we are filtering non relevant pages, in particular spam pages that may bring noise to our technique.

There are two main data sources for obtaining clicked queries for documents, and depending on the source we might have *partial* queries or *complete* queries:

- **partial queries:** This is the case when the usage data is obtained from a search engine's query log. This situation is most likely to occur when organizing general Web documents or search results. Query clicks to documents discovered from this log are *only* the ones that were submitted to the particular search engine that generated the log. Therefore, the more widely used the search engine is, the better it will represent the real usage of documents.

- **complete queries:** This is the case when the usage data is obtained from a website's access logs. This situation is most likely when organizing documents belonging to a particular website. Standard combined access logs allow (very easily) to discover *all* of the queries from Web search engines that directed traffic to the site (i.e., queries from which documents in the site were clicked). This log may also contain information about queries to the internal search engine of the website (if one is available).

We present two document models based on queries and their clicked URLs, the *query document model*, which uses query terms as features, and an enhanced *query-set document model*, which uses query-sets as features.

3.1 Query Document Model

As a first approach to using queries to represent documents, we present the *query document model*. This model is a formalization and extension of the *query view* idea [9]. We extend [9] by not limiting queries only to those from internal searches, but including *all* possible queries available (complete or partial queries). The query document model consists of representing documents using as features *only query terms*. The queries used to model a document are only those for which users clicked on that document.

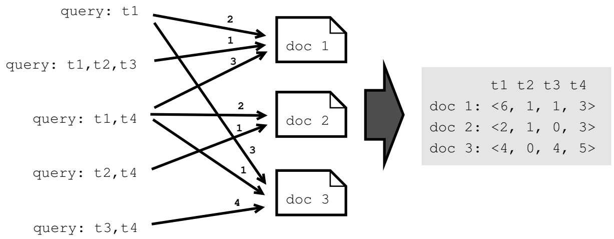


Figure 1: Example of the query document representation, without normalization.

The query document model reduces the feature space dimensions considerably, because the number of terms in the query vocabulary is smaller than that of the entire website collection. This model is very similar to the vector model, with the only difference that instead of using a weighted set of keywords as vector features, we will use a weighted set of query terms. The weight of each term corresponds to the frequency with which each query that contains the term appears in the usage log as a referrer for the document. In other words: how many times users reach a document by submitting a query that contains a particular term. These query representations of Web documents are then normalized according to the well-known *tf-idf* scaling scheme. Figure 1 shows a simple example (without normalization) of a query document representation. This example shows a set of queries, the terms included in each query, the documents that were reached by users from the queries, and the number of times that this happened. This information is processed to create the query document representations.

More formally we define the query document model as:

Let $d_1, d_2, \ldots, d_n$ be a collection of documents, and let V represent the vocabulary of all queries found in the access log L. Moreover, let $t_1, t_2, \ldots, t_m$ be the list of terms in vocabulary V. Let $Q(d_i)$ be the set of all the queries found in L from which users clicked at *least one time* on a document d_i, and let the *frequency* of t_j in $Q(d_i)$ be the total number of times that queries that contained t_j were used to visit d_i. The query representation of d_i is defined as:

$$\overrightarrow{d_i} = \langle C_{i1}, C_{i2}, \ldots, C_{im} \rangle$$

where

$$C_{ij} = tf - idf(t_j, Q(d_i))$$

and $tf - idf(t_j, Q(d_i))$ is the $tf - idf$ weight assigned to t_j for $Q(d_i)$.

Besides reducing the feature space, another result of this representation is that documents are now described using terms that summarize their relevant contents according to the users point of view. In subsection 3.3 we discuss the case when $C_{ij} = 0, \forall j$.

Although we use queries for modeling documents, this approach differs from [24] in the following way: the queries considered for document features are *only* those from which users clicked on the document as a result of the query in the search engine. This way, implicit user feedback is used to relate queries to documents, and the frequency of clicks is considered for feature weight, and not the rank.

It is important to note that not all visits to a Web page in response to a query are relevant, i.e., some users could click on a result to find out that the page that they are visiting is not what they thought it would be. The only guide that users have to click on a Web page are the snippets displayed on the search engine results. To counteract this effect, the frequencies of clicks from a query to a document are considered in the vectors as a heuristic to attempt to give more importance to highly used queries and reduce noise due to errors.

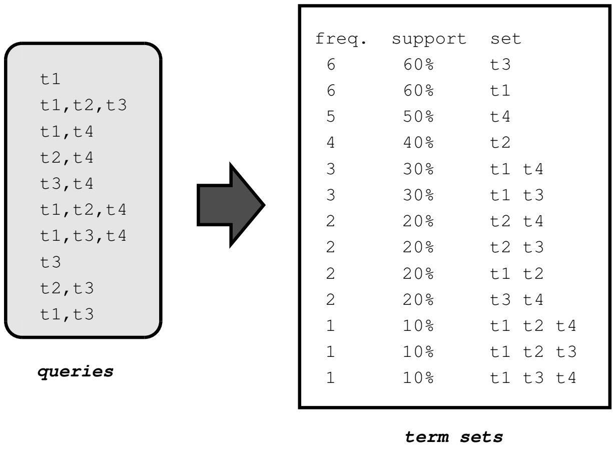
3.2 Query-Set Document Model

The main drawback for the query model, is that it considers terms independent from each other even if they occur many times together in a same query. This can cause the loss of important information since many times more than one term is needed to express a concept. Also a term occurring inside a set can have different meanings if we change other elements in that set. For example, the two term queries *class schedule* and *class diagram* have different meanings for the word *class*. The first refers academic classes, and the second more likely to UML classes. To address this problem, which happens frequently in Web queries, we have created an enhanced version of the query model, called *query-set document model*.

The query-set model uses frequent query-sets as features, and aims at preserving the information provided by the co-occurrence of terms inside queries. This is achieved by mining frequent itemsets or frequent query patterns. Every keyword in a query is considered as an item. Patterns are discovered through analyzing all of the queries from which a document was clicked, to discover recurring terms that are used together. The difference with this model and the previous is that instead of using queries directly as features in a vector, we use all the frequent itemsets that have a certain support. The novelty of this approach relies on the combination of user feedback for each document, and mining frequent query sets to produce an appropriate document model. Previous work such as [7, 23] use itemsets for feature selection over the full text of documents, our model on the other hand, applies this only to queries. We believe that frequent sets mined from queries are more powerful and expressive than the sets extracted from the full text of the documents. Frequent sets mined from the full text of documents have a similar problem to that of the vector space model, i.e.: not selecting sets from the terms that user's consider relevant. Queries on the other hand, already have the selected keywords that summarize the document from the user's perspective.

The support for frequent itemsets is decided for each collection experimentally, based on the frequency distribution of queries in a usage log. In general, the support decreases as the number of terms in a set increase. Figure 2 shows an example of all term sets found for a sample of queries. From this it is possible to determine the minimal support allowed for queries with 1, 2 and 3 terms, to obtain only the most relevant sets.

After the relevant sets of terms for each document are extracted, a weighed vector is created for the query-set document representation. Each dimension of the feature space is given by all the unique relevant term sets found in the usage logs. Each *term set* is a unit, and it cannot be split. The weight of each feature in the vector is the number of times that the pattern appears in a query that clicked on the document.



```
t1
t1,t2,t3
t1,t4
t2,t4
t3,t4
t1,t2,t4
t1,t3,t4
t3
t2,t3
t1,t3
```

queries

freq.	support	set
6	60%	t3
6	60%	t1
5	50%	t4
4	40%	t2
3	30%	t1 t4
3	30%	t1 t3
2	20%	t2 t4
2	20%	t2 t3
2	20%	t1 t2
2	20%	t3 t4
1	10%	t1 t2 t4
1	10%	t1 t2 t3
1	10%	t1 t3 t4

term sets

Figure 2: Example of all the terms sets found for a group of queries and their supports.

More formally we define the query-set document model as follows:

Let $d_1, d_2, \ldots, d_n$ be a collection of documents, and let V' represent the vocabulary of all relevant terms sets found in the access log L. Moreover, let $ts_1, ts_2, \ldots, ts_m$ be the list of term sets in vocabulary V'. Let $Q'(d_i)$ be set of queries found in L from which users clicked at least one time on a document d_i, and let the *frequency* of ts_j in $Q'(d_i)$ be the total number of times that queries that contained ts_j reached d_i. The *query-set* representation of d_i is defined as:

$$\overrightarrow{d_i} = \langle C'_{i1}, C'_{i2}, \ldots, C'{}_{im} \rangle$$

where

$$C'_{ij} = tf - idf(ts_j, Q'(d_i))$$

and $tf - idf(ts_j, Q'(d_i))$ is the $tf - idf$ weight assigned to ts_j for $Q'(d_i)$.

3.3 Modeling Documents that do not have Queries

Since all the query-based approaches represent documents only by using queries it is necessary to consider documents that do not register visits from queries. This is needed if we want to model a complete collection of documents. There are several alternatives for modeling and clustering these remaining documents. A straightforward approach is to model all documents with queries using the query-set model and for the remaining documents use a partial set-based model (see [23]), but only using the feature space of the query-sets (V'). If the query model was being used (instead of the query-set model) then the remaining documents would have to be modeled using the traditional vector space approach, but using only the query vocabulary (V).

General Log Statistics

Period	November 2006	
Sessions	610,668	
Documents clicked from queries	29,826	
	Website	***Top-100***
Total queries	158,481	126,849
Unique queries	96,733	26,152

Table 1: General log statistics for the website.

	% of Documents	*# of Visits*
Not visited	0.52	0
Only from queries	9.40	0.94
Only by navigation	18.20	12.54
Both	71.87	86.51

Table 2: General statistics on how users reached the documents of the website.

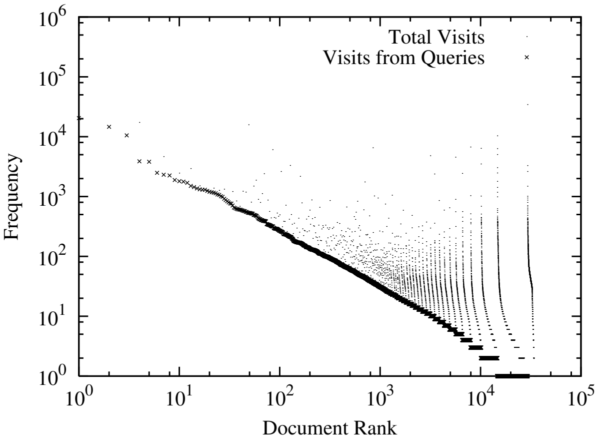


Figure 3: Distribution of query clicks and visits to documents of a website (documents ranked by # of queries then by # of visits).

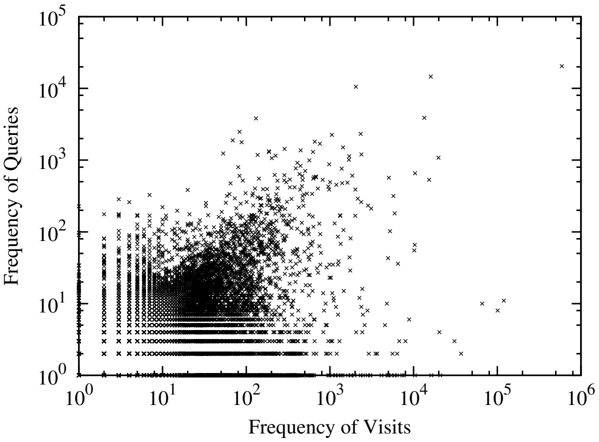


Figure 4: Scatter plot of the frequency of queries and the frequency of navigational visits in a website (each dot represents a document from the site).

The main focus of our work, is on documents that were reached by queries. Evaluation and further discussion of the best approach for documents that do not have queries will be pursued in the future, but as we show in the next section, our approach covers most of the relevant pages in a website. In addition, there are many techniques available to cluster and label text documents.

4. EVALUATION AND DISCUSSION

As a first evaluation for our query-based models we chose to use a website with its access logs. This decision was based on the fact that by using a website's logs we can have access to complete queries and this gives us a full view of the query range from which users visited the site. The second motivation for evaluating using a website is that the collection of documents already have a strong similarity, so the clusters will be specialized and not trivial.

For the evaluation we used as a case study a large portal directed to university students and future applicants. This website gathers a great amount of visits and contents from a broad spectrum of educational institutions. Table 1 shows some general statistics of the one month log used for this evaluation. Table 2 describes how the different documents in the website were *found* by users (i.e., clicked on for the first time in a session). Documents in Table 2 are divided into: URLs that were not visited by users, URLs that only had visits as a result of a query click, visits from users who browsed to the URL (navigation), and visits from both (this implies that some users visited a page by browsing while others visited the same page by clicking on a query result). We can observe that the documents clicked at some point from queries represent more than 81% of the documents and more than 87% of the traffic to the Web site. Therefore, our technique applies to a large fraction of the pages and the most important ones in the site.

In Figure 3 we show the documents in the site sorted by query click frequency and by visit frequency. We can observe that the query click frequency distribution over documents is a power law with exponent -0.94. This is a very low absolute value, in comparison the usual power law found in Web search engine data. We believe this can be explained by the fact that the power law distribution of words in queries is correlated with the power law distribution of words in documents [1]. Therefore, since the query-based models only study queries with clicked results, the query-document distribution decays at a slower pace than the usual word-document distribution. On the other hand, in Figure 4 we can see the correlation between the frequency of queries and the frequency of navigational visits (without considering clicks from queries) for the URLs is low, as shown in Figure 4. This implies something that we expected: queries are being used to find documents that are not found usually by navigation. Consequently, the organization of documents into topics using queries can provide additional information to the current structure of the site.

To evaluate the performance of the query-based models, we have divided the evaluation process into two main steps. First, we compared the traditional vector model representation with the query representation, and secondly, we compared the results from the query document model to the enhanced version that uses query patterns. These documents were selected from the top 100 with most queries in the site

Number of Clusters	Internal Similarity	External Similarity
10	0.210	0.0280
15	0.281	0.0288
20	0.345	0.0299
25	0.394	0.0316

Table 3: Average ISim and ESim values for different numbers of clusters.

and they capture a large fraction of the queries to the site (see Table 1). This choice of documents was made to have a large enough sample of query terms and also to use documents that were important to the site in terms of being the most visible ones from the Web.

Each one of the 100 documents in the sample was modeled according the three different document models that we evaluated: *vector-space*, *query* and *query-set* (i.e., 300 different representations in total). All the data (log and Web pages) were previously cleaned using standard approaches, as described in [10], which include the removal of stopwords and irrelevant requests, amongst other things. For the content based representation, only text contents from each document was considered, no hypertext characteristics were used. The queries used in this process, consisted of queries submitted by users during one month. The log used and the contents of the documents, belong to the same time period.

Each set of documents, grouped by their representation, was clustered into 15 clusters, and automatically labeled using the top most descriptive features of each group, according to the clustering system CLUTO [18]. The number of clusters was chosen experimentally by trying a few numbers that seemed appropriate for the amount of documents and desired level of granularity of topics. We tested 10, 15, 20 and 25 clusters for the vector space representation, and decided based on the one that provided the greatest increase of *internal similarity (ISim)* and at the same time less *external similarity (ESim)*, shown in Table 3. ISim is the average similarity between the objects of each cluster (i.e., internal similarities), and ESim is the average similarity of the objects of each cluster and the rest of the objects (i.e., external similarities). The choice of the correct amount of clusters in general is a complex task, and is beyond the scope of this research, so our choice was based on the ISim and ESim values and what seemed appropriate by inspecting the documents.

The clustering process used was sequential bisections, optimizing in each iteration the global clustering function:

$$\max\left(\sum_{i=1}^{k}\sqrt{\sum_{v,u\in S_i} sim(u,v)}\right)$$

where k is the total number of clusters, S_i is the number of elements in the i-cluster, u, v represent two objects in the cluster and $sim(u,v)$ correspond to the similarity between two objects. This function was experimentally found appropriate for document clustering, as discussed in [18]. The similarity in this case is measured using the cosine function between vectors.

Each clustering process, assigned automatically a cluster and a label to each document. This way, every document ended up with a different cluster and label for each one of the three document models. To evaluate the appropriateness of clusters and labels, each document representation was classified by three (out of a group of six) human experts, on the subject area of the site. Each judge measured the *quality* of a document to its label, for a number of documents (between a 100 or 200), from a total of 300 document representations. The experts were asked to evaluate using 1 or 0, whether or not the document belonged to the topic described by its label. Our goal is to evaluate the compatibility of documents to its labels, to measure the quality of the automatically generated topics as well as the groups of documents in each topic. Our main interest at this point is to group documents into relevant topics and label them accordingly. Our evaluation approach allows us to know if the topics, derived from the labels, are relevant and human understandable, as well as if the documents in them belong to these categories.

Terms	***Support***
1	10.00%
2	9.80%
3	9.00%
4	2.15%
5	0.95%

Table 4: Resulting support table for the different pattern sizes.

Model	***Quality***	***Dimensions***	***Agreement***
Vector-Space	40%	8,910	69%
Query	57%	7,718	67%
Query-Set	**77%**	**564**	**81%**

Table 5: Experimental results for each document model.

For the query-set document model the minimum support for query patterns of different sizes was determined experimentally. In order to do this we analyzed all the query patterns contained in the log sample and then plotted the histogram of the number of queries that had different support levels, the tool used for this purpose was LPMINER [27]. This was done for patterns with 1, 2, 3, 4 and 5 terms, to obtain the support level for each case. Figure 5 shows the graphs for each case, from which the support level for each case was chosen ruling out support levels that include too many query patterns. Table 4 shows a summary of the resulting support table.

In Table 5 we show the overall results obtained for each type of document representation. This includes the quality, the number of total features (or dimensions) and the level of inter-judge agreement during the classification process. The quality of a document within each representation, was decided using the vote of the *majority* (at least two judges out of three). From this table, it is important to notice that both models based on queries outperform the vector-space representation, but the query-set model makes exceptional improvements in all of its results. Table 7 shows some examples of keyword labels obtained with the different document models. It is important to note that the topics for the query-based methods are both similar, but these labels differ greatly from the vector space labels (i.e., they use prioritize very different terms).

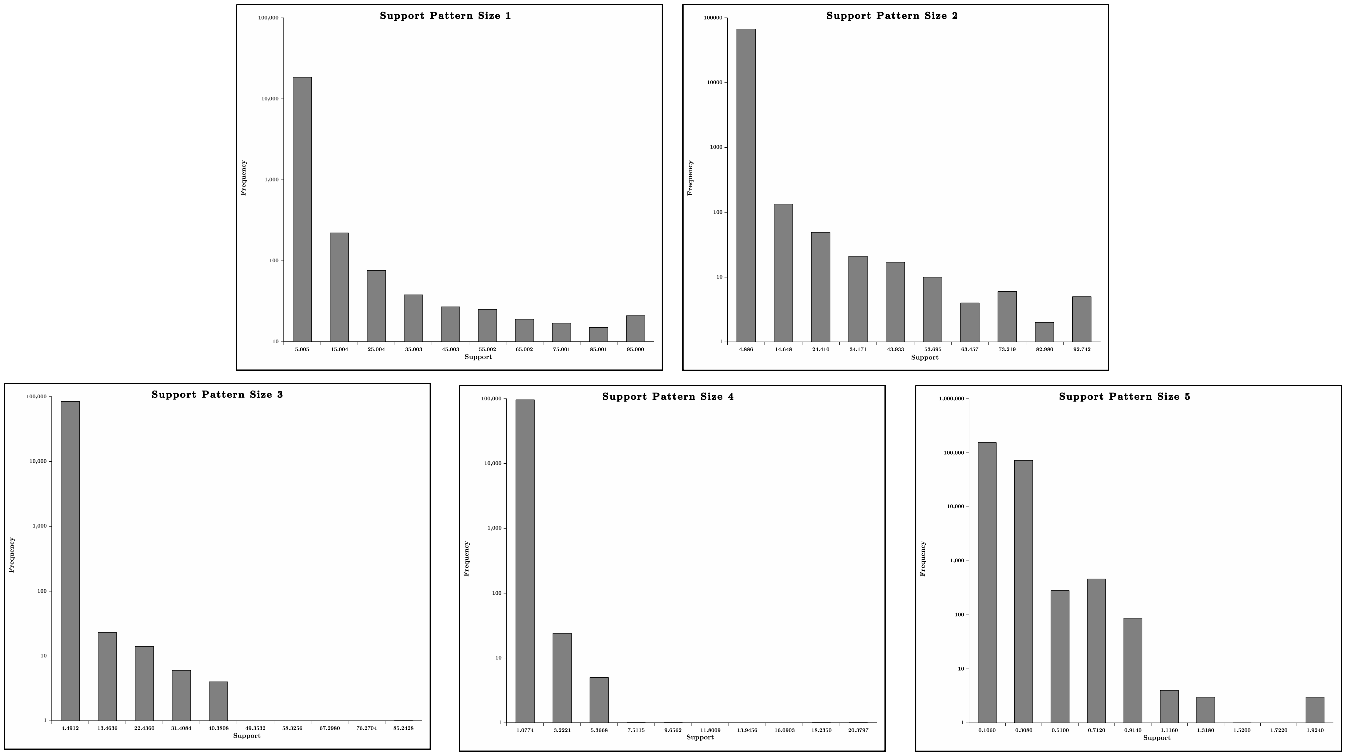


Figure 5: Support graphs for different pattern sizes.

In Table 5 we can view the level of inter-judge agreement for each model's clustering. The agreement percent is high for all models, especially for the query-set model where it reaches 81%. We believe that the query-set model has higher agreement because the features and document labels are more accurate to what users expect than in other models. The possibility of inter-judge agreement happening by random chance is extremely low and is given by:

$$P_{agreement} = \sum_{k \leq i \leq n} \binom{n}{i} P^{i} \left(1 - P\right)^{n-i}$$

where

$$P = \sum_{\lceil j/2 \rceil \leq s \leq j} \binom{j}{s} w^{s} \left(1 - w\right)^{j-s}$$

Where k is the number of documents in one document model for which the majority of experts agree (in our case, at least 2 out of 3 judges must agree), n is the total number of documents for one model, and w is the probability of an expert tagging a document with 1. We suppose an homogeneous distribution and use $w = 0.5$. In Table 6 we show the probabilities for different possible values of k, considering the number of judges, $j = 3$ for each document. Even for the largest value of k, the chance of random agreement is very low. This supports the notion that experts truly agreed on their assessment criteria.

k	$P_{agreement}$
67	4.36×10^{-04}
69	9.15×10^{-05}
81	1.35×10^{-10}

Table 6: Probability of random inter-judge agreement, with $j = 3$ and $w = 0.5$.

5. CONCLUSIONS AND FUTURE WORK

As the Web grows in number of documents and amount of content, there is an increasing interest towards supporting tasks such as maintenance, organization and website design. Queries in Web search engines play a key role in site traffic and provide valuable insight on implicit user feedback of the usage of the Web pages.

This work focuses on document modeling based on queries. In particular we formalize a *query document model* and introduce a new representation based on frequent query patterns, called the *query-set document model*. Our evaluation shows that queries are excellent features for describing documents. In our experiments, all of the query-based representations outperform the *vector space* model when clustering and labeling documents of a website. The most relevant result of our study shows that the query-set model reduces by over 90% the number of features needed to represent a set of documents and improves by more than 90% the quality. Also, the query-set model shows a higher level of inter-judge agreement which corresponds with the fact that the topics generated by this model are more relevant and comprehensive. Also, it is important to observe that the feature di-

DocId	Vector Space	Query	Query-Set
58	download, test, file, 2007, guide, publication	official, test, social, publication, module, science, guides	physics, geometry, physics topics, topics, admission topics
74	able, Europe, world, kingdom, MBA, Asia, library	degree, search, graduate, certificate, advanced, diploma, simulation	university scholarship, universities, university ranking, best universities
47	scholarship, application, loan, benefit, fill, form	dates, free, vocational, on-line, scholarship, loan	loan scholarship loan cosigner loan application
80	vitae, curriculum, presentation, job, letter, interview, experience, highlight	CV, letter, resume, recommendation, presentation, example	CV, write CV, curriculum vitae, CV example, write curriculum vitae

Table 7: Examples of keyword labels obtained with the different document models.

mensionality reduction achieved by our query-set model is very important. This applies especially for very large document collections, since it reduces computational cost while increasing quality in the results.

Future work includes conducting a larger evaluation of the query-set model using several sites as well as compare our techniques to other possible models, for example based in n-grams or frequent itemsets over the full text of documents. However, as a first evaluation we decided to focus only on a website because it has the advantage that the vocabulary is smaller and specific to certain topics, while the overall Web would be much more heterogeneous. Nevertheless, in future work we will include a broader comparison with an online directory. We want to compare how human edited topics and classification of documents differ from the ones generated by the query-set model. We would expect our method to discover new and different topics from the ones in the directory.

Also, we want to evaluate this document model within a tool for improving websites, such as [22]. Furthermore, we want to assess how this work can help to improve search engine results. Also, it would be interesting to incorporate other document features into the model, as part of a *mixed-model*, to unbias the effect of the search engine rank of documents over the likelihood of a document to be clicked by a user [4, 13]. Additionally, another related problem is how to model usage of documents accessed by queries and/or navigation. Our graphs in Section 4 give some insight in this problem, but a more detailed study is needed. One possibility is to use the anchor text of the links on the navigation path to a page as good descriptors of a document, like most search engines do. Mixing anchor text with queries can provide a more full document coverage (over 99% in our example) and combines generic labels (initial links) with more specific labels (later links), enriching our model.

Acknowledgments

This work was partially funded by Spanish Education Ministry Grant TIN2006-15536-C02-01 and Catalonian Research Agency Grant 2005 SGR 01076.

The authors thank the following people from Yahoo! Research Barcelona: Carlos Castillo, Aristides Gionis and Hugo Zaragoza for our valuable discussions and feedback. We are grateful also to Ximena Olivares, Rodrigo Meza, Christian Middleton and Hugo Salgado from the Web Research Group at University Pompeu Fabra for their help with the document assessment process.

6. REFERENCES

[1] R. Baeza-Yates. Web usage mining in search engines. In *Web Mining: Applications and Techniques, Anthony Scime, editor.*, pages 307–321. Idea Group, 2004.

[2] R. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval.* ACM Press / Addison-Wesley, 1999.

[3] R. A. Baeza-Yates, C. A. Hurtado, and M. Mendoza. Query clustering for boosting web page ranking. In J. Favela, E. M. Ruiz, and E. Chávez, editors, *AWIC*, volume 3034 of *Lecture Notes in Computer Science*, pages 164–175. Springer, 2004.

[4] R. A. Baeza-Yates, C. A. Hurtado, and M. Mendoza. Improving search engines by query clustering. *JASIST*, 58(12):1793–1804, October 2007.

[5] R. A. Baeza-Yates and B. Poblete. A website mining model centered on user queries. In M. Ackermann, B. Berendt, M. Grobelnik, A. Hotho, D. Mladenic, G. Semeraro, M. Spiliopoulou, G. Stumme, V. Svátek, and M. van Someren, editors, *EWMF/KDO*, volume 4289 of *Lecture Notes in Computer Science*, pages 1–17. Springer, 2005.

[6] D. Beeferman and A. Berger. Agglomerative clustering of a search engine query log. In *KDD*, 1999. Boston, MA USA.

[7] F. Beil, M. Ester, and X. Xu. Frequent term-based text clustering. *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 436–442, 2002.

[8] B. Berendt and M. Spiliopoulou. Analysis of

navigation behaviour in web sites integrating multiple information systems. In *VLDB Journal, Vol. 9, No. 1*, pages 56–75, 2000.

[9] M. Castellanos. Hotminer: Discovering hot topics from dirty text. In M. W. Berry, editor, *Survey of Text Mining*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2003.

[10] R. Cooley, B. Mobasher, and J. Srivastava. Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems*, 1(1):5–32, 1999.

[11] R. Cooley, P. Tan, and J. Srivastava. Websift: the web site information filter system. In *KDD Workshop on Web Mining, San Diego, CA. Springer-Verlag, in press*, 1999.

[12] R. Cooley, P.-N. Tan, and J. Srivastava. Discovery of interesting usage patterns from web data. In *WEBKDD*, pages 163–182, 1999.

[13] G. Dupret, V. Murdock, and B. Piwowarski. Web search engine evaluation using clickthrough data and a user model. In *WWW2007 workshop Query Log Analysis: Social and Technological Challenges*, 2007.

[14] M. Eirinaki, C. Lampos, S. Paulakis, and M. Vazirgiannis. Web personalization integrating content semantics and navigational patterns. *Proceedings of the 6th annual ACM international workshop on Web information and data management*, pages 72–79, 2004.

[15] J. Fürnkranz. Exploiting structural information for text classification on the www. *Intelligent Data Analysis*, pages 487–498, 1999.

[16] K. Hammouda and M. Kamel. Phrase-based document similarity based on an index graph model. *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02)*, page 203, 2002.

[17] I.-H. Kang and G. Kim. Query type classification for web document retrieval. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 64–71, New York, NY, USA, 2003. ACM Press.

[18] G. Karypis. CLUTO a clustering toolkit. Technical Report 02-017, Dept. of Computer Science, University of Minnesota, 2002. Available at http://www.cs.umn.edu/~cluto.

[19] F. Masseglia, P. Poncelet, and M. Teisseire. Using data mining techniques on web access logs to dynamically improve hypertext structure. *ACM SigWeb Letters vol. 8, num. 3*, pages 1–19, 1999.

[20] B. Mobasher, R. Cooley, and J. Srivastava. Automatic personalization based on web usage mining. *Commun. ACM*, 43(8):142–151, 2000.

[21] M. Perkowitz and O. Etzioni. Adaptive web sites: an AI challenge. In *IJCAI (1)*, pages 16–23, 1997.

[22] B. Poblete and R. Baeza-Yates. A content and structure website mining model. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 957–958, New York, NY, USA, 2006. ACM Press.

[23] B. Pôssas, N. Ziviani, J. Wagner Meira, and B. Ribeiro-Neto. Set-based vector model: An efficient approach for correlation-based ranking. *ACM Trans. Inf. Syst.*, 23(4):397–429, 2005.

[24] D. Puppin, F. Silvestri, and D. Laforenza. Query-driven document partitioning and collection selection. In *InfoScale '06: Proceedings of the 1st international conference on Scalable information systems*, page 34, New York, NY, USA, 2006. ACM Press.

[25] F. Radlinski and T. Joachims. Query chains: learning to rank from implicit feedback. In *KDD '05: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 239–248, New York, NY, USA, 2005. ACM Press.

[26] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18(11):613–620, 1975.

[27] M. Seno and G. Karypis. Lpminer: An algorithm for finding frequent itemsets using length-decreasing support constraint. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 505–512. IEEE Computer Society, 2001.

[28] D. Shen, J.-T. Sun, Q. Yang, and Z. Chen. A comparison of implicit and explicit links for web page classification. In *WWW '06: Proceedings of the 15th international conference on World Wide Web*, pages 643–650, New York, NY, USA, 2006. ACM Press.

[29] A. Sieg, B. Mobasher, S. Lytinen, and R. Burke. Using concept hierarchies to enhance user queries in web-based information retrieval. In *IASTED International Conference on Artificial Intelligence and Applications*, 2004.

[30] M. Spiliopoulou. Web usage mining for web site evaluation. *Commun. ACM*, 43(8):127–134, 2000.

[31] J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1(2):12–23, 2000.

[32] P. Tonella, F. Ricca, E. Pianta, and C. Girardi. Using keyword extraction for Web site clustering. *Web Site Evolution, 2003. Theme: Architecture. Proceedings. Fifth IEEE International Workshop on*, pages 41–48, 2003.

[33] X. Wang and C. Zhai. Learn from web search logs to organize search results. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 87–94, New York, NY, USA, 2007. ACM.

[34] Y. Wang and J. E. Hodges. Document clustering using compound words. In *IC-AI*, pages 307–313, 2005.

[35] G.-R. Xue, H.-J. Zeng, Z. Chen, W.-Y. Ma, and C.-J. Lu. Log mining to improve the performance of site search. In *WISEW '02: Proceedings of the Third International Conference on Web Information Systems Engineering (Workshops) - (WISEw'02)*, page 238, Washington, DC, USA, 2002. IEEE Computer Society.

[36] J. Zhu, J. Hong, and J. G. Hughes. Pagecluster: Mining conceptual link hierarchies from web log files for adaptive web site navigation. *ACM Trans. Inter. Tech.*, 4(2):185–208, 2004.