UNIVERSITAT
POMPEU FABRA

Query-Based Data Mining for the Web

Ph.D. Candidate: Barbara Poblete

Directors de la Tesi / Thesis Advisors
Dr. Ricardo Baeza-Yates
Dr. Myra Spiliopoulou

TESI DOCTORAL UPF
2009

Para Esperanza y Rodrigo

Acknowledgments

The author thanks her advisors Ricardo Baeza-Yates and Myra Spiliopoulou for their support, ideas and collaboration that helped immensely in the production this work. The author is also grateful to all the people at the Yahoo! Research Lab in Barcelona, specially to Aris Gionis and Carlos Castillo for their contributions to this thesis, and to Debora Donato, Vanessa Murdock, Vassilis Plachouras and Hugo Zaragoza for their valuable advise.

Abstract

The objective of this thesis is to study different applications of Web query mining for the improvement of search engine ranking, Web information retrieval and Web site enhancement. The main motivation of this work is to take advantage of the implicit feedback left in the trail of users while navigating through the Web. Throughout this work we seek to demonstrate the value of queries to extract interesting rules, patterns and information about the documents they reach. The models, created in this doctoral work, show that the "wisdom of the crowds" conveyed in queries has many applications that overall provide a better understanding of users' needs in the Web. This allows to improve the general interaction of visitors with Web sites and search engines in a straightforward way.

Resumen

El objetivo de esta tesis es estudiar diferentes aplicaciones de la minería de consultas Web para mejorar el ranking en motores de búsqueda, mejorar la recuperación de información en la Web y mejorar los sitios Web. La principal motivación de este trabajo es aprovechar la información implícita que los usuarios dejan como rastro al navegar en la Web. A través de este trabajo buscamos demostrar el valor de la "sabiduría de las masas", que entregan las consultas, para muchas aplicaciones. Estas aplicaciones permiten un mejor entendimiento de las necesidades de los usuarios en la Web, mejorando en forma directa la interacción general que tienen los visitantes con los sitios Web y los buscadores.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The Web is unlike any other repository of information that we have ever studied before; it is an immensely rich repository which grows at an astoundingly fast pace. These unique characteristics carry many new challenges for Web researchers, which include among other things, high data dimensionality and highly volatile and constantly evolving content. Due to this, it has become increasingly necessary to create new and improved approaches to traditional data mining techniques can be applied to the Web. In this regard, recognizing and separating automatically *interesting and valuable information*, has become a very relevant problem when processing such huge quantities of data. The key issues in this matter are: *how do we know which information is interesting or useful?* and *how can we find this information automatically?*.

In this thesis we focus our efforts on analyzing and extracting valuable knowledge from the behavior of users on the Web. Much of this information is provided implicitly by users and recorded in *usage logs*, which include search engine *query logs* and/or *website access logs*. In particular, we center our research on queries that users submit to Web search engines, which we believe convey in a straightforward way "wisdom of the crowds". The intuition is that queries and their clicked results implicitly convey the opinion of users about specific Web documents. As we will discuss throughout this work, queries are crucial to understanding how users interact with Web sites and search engines. Implicit user feedback provides a unique insight into users' actual needs on the Web.

Specifically, in this work we seek to use the knowledge extracted through queries, to improve the *organization*, *retrieval* and *ranking* of Web documents and Web sites. We explore several new applications and their impact, including the implications for privacy preservation of query log data mining.

There are five main contributions of this work:

1. A new document representation model based on search engine queries. The main objective of this model is to achieve better results in non-supervised tasks, such as clustering and labeling of documents, 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 summa-

rizing 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 Web site. 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.

2. A novel methodology to discover *similar Web sites* in the Web, which is useful for automatic directory generation and business intelligence applications. This approach, based on (1), is intended to find groups of sites that satisfy similar information needs from users. The main contribution of this approach is that Web sites can be compared independently of their size and structure. To do this, we model entire Web sites across feature spaces of clicked queries and use association rule discovery to detect related query terms and reduce their dimensionality. Thus we achieve highly compact and effective representations of Web sites. Formally, we present a generic framework that allows us to generate alternative query-based Web site representations, which we use to cluster similar Web sites together. Our experiments on site clustering show that cluster quality on query-based feature spaces is superior to that of clusters on conventional feature spaces and requires a much smaller number of features – down to 5% for one of the alternative feature spaces we studied.

3. A Web site data mining model, centered on user queries, that uses the access logs of a Web site to evaluate the site's usage, structure and content. By using queries, we discover in a simple way, valuable information to improve the quality of the Web site, allowing the Web site to become more intuitive and useful for the needs of its users. In particular, we present a methodology to analyze and classify different types of queries registered in usage logs, such as queries submitted by users to the site's internal search engine and queries on global search engines that lead to documents in the Web site. These queries provide useful information about topics that interest users visiting the Web site and the navigation patterns associated to these queries indicate whether or not the documents in the site satisfied the user's needs at that moment. Preliminary evaluation of our model shows significant improvements to the Web site reflected in an increase in visits and traffic from search engines.

4. A unified graph view of the Web, which includes both structural and usage data. We model this graph using a simple union of the Web's hyperlink and click graphs. The hyperlink graph expresses link structure among Web pages, while the click graph is a bipartite graph of queries and documents denoting users' searching behavior extracted from a search engine's query log. Our most important motivation is to model in a unified way the two main activities of users on the Web: *searching* and *browsing*, and at the same time to analyze the effects of random walks on this new graph. The intuition behind this task is to measure how the combination of link structure

and usage data provide additional information to that contained in these structures independently. Our experimental results show that both hyperlink and click graphs have strengths and weaknesses when it comes to using their stationary distribution scores for ranking Web pages. Furthermore, our evaluation indicates that the unified graph always generates consistent and robust scores that follow closely the best result obtained from either individual graph, even when applied to "noisy" data. It is our belief that the unified Web graph has several useful properties for improving current Web document ranking, as well as for generating new rankings of its own. In particular stationary distribution scores derived from the random walks on the combined graph can be used as an indicator of whether structural or usage data are more *reliable* in different situations.

5. An analysis of privacy issues related to the analysis and publication of search engine query logs and derived data. In particular we introduce the concern of privacy protection of *business confidential information*. We study *business privacy*, as the privacy of institutions, such as companies and people in the public eye. In particular, we relate this privacy concern to the involuntary exposure of confidential Web site information. We characterize the possible adversaries interested in disclosing Web site private data and the attack strategies that they could use. These attacks are based on different vulnerabilities found in query log data for which we present several anonymization heuristics to prevent them. We perform an experimental evaluation to estimate the remaining utility of the log after the application of our anonymization techniques. Our experimental results show that a query log can be anonymized against these specific attacks while retaining a significant volume of useful data.

Even though these contributions cover mostly independent approaches in Web data mining, they have the common goal of exploring search engine queries as a indicators of what users consider to be important in the Web. Contributions (1) and (2) use search engine query logs to model first Web documents and then full Web sites. They use queries and their properties for simple, yet effective, feature selection. Contribution (3) uses the information recorded in a Web site log to improve that same Web site. This improvements are based on the queries used to reach each document, plus other navigational behaviors. Queries are used to select which of the discovered rules and patterns are important. Contribution (4) explores the incorporation of the click graph, which reflects the querying behavior of users in search engines, to the traditional hyperlink graph of the Web. Finally, contribution (5) studies the particular effect on privacy of disclosing query log data for data mining purposes.

This thesis can be viewed as the exploration of query mining on the Web at two different granularity levels. First at a *macro level*, from the point of view of Web sites (contributions (2),(3) and (5)), and secondly, at a *micro level*, from the point of view of documents (contributions (1) and (4)).

This work has produced the following publications:

Book Chapters

- **B. Poblete**, M. Spiliopoulou, R. Baeza-Yates: *Website Privacy Preservation for Query Log Publishing*. Proceedings of the First SIGKDD International Workshop on Privacy, Security, and Trust in KDD (PinKDD'07), Lecture Notes in Computer Science, Volume 4890, Springer, 2008.
- R. Baeza-Yates, **B. Poblete**: *A Website Mining Model Centered on User Queries*. In book: Semantics, Web and Mining, editors: B. Behrendt, M. Grobelnik, A. Hotho, D. Mladenic, G. Semeraro, M. van Someren, M. Spiliopoulou, G. Stumme, V. Svatek. Lecture Notes in Computer Science. Publisher: Springer Berlin / Heidelberg. 2006.

Conference & Workshop Papers

- **B. Poblete**, C. Castillo, A. Gionis. 2008. *Dr. Searcher and Mr. Browser: A Unified Hyperlink-Click Graph*. To appear in Proceedings of the ACM 17th Conference on Information and Knowledge Management (Napa Valley, California, October 26-30, 2008). CIKM'08. ACM Press, New York, NY.
- **B. Poblete**, R. Baeza-Yates, 2008. *Query-Sets: Using Implicit Feedback and Query Patterns to Organize Web Documents*. In Proceedings of the 17th international Conference on World Wide Web (Beijing, China, April 21 - 25, 2008). WWW '08. ACM Press, New York, NY.
- **B. Poblete**, M. Spiliopoulou, R. Baeza-Yates. *Website Privacy Preservation for Query Log Publishing*. First International Workshop on Privacy, Security, and Trust in KDD (PINKDD'07), San Jose, CA, USA. August 2007. **Best Paper Award**.
- **B. Poblete**, R. Baeza-Yates. 2006. *A Content and Structure Website Mining Model*. Poster. In Proceedings of the 15th international Conference on World Wide Web (Edinburgh, Scotland, May 23 - 26, 2006). WWW '06. ACM Press, New York, NY, 957-958.
- R. Baeza-Yates, **B. Poblete**. *A Website Mining Model Centered on User Queries*. European Web Mining Forum (EWMF 2005), pp. 3-15. Oporto, Portugal.

Submitted for Review

- **B. Poblete**, M. Spiliopoulou, R. Baeza-Yates. *Business Privacy Protection in Query Log Mining*. Submitted to the Journal ACM Transactions on the Web. TWEB, 2009.
- **B. Poblete**, M. Spiliopoulou. *Discovering Similar Websites Using Search Engine Queries*. Submitted to the 15th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2009.

The presentation of the Ph.D. work is divided into eight chapters, each one presenting different contributions in the area of query mining and usage mining. Each chapter is based on a published or submitted article, which have been extended in some cases and in others unified with similar work. These chapters are organized in the following way: Chapter 2 presents general background and a preliminary State of the Art. More specific related work is included and discussed within the rest of the chapters, according to each subject. Chapters 3 and 4 present new query-based document and Web site representation models, respectively. Chapter 5 introduces a data mining model centered on user queries to improve Web site organization and contents. Chapter 6 presents a unified view of the Web graph which merges structural and usage data. Chapter 7 analyzes the implications of query log data mining for business confidentiality. Chapters 6 and 7 have more of an exploratory nature, and seek to open a new perspective for research in these areas. Finally Chapter 8 presents a final discussion of the thesis.

Chapter 2

Background

In this chapter we present an overview of the area of *Web mining*, specifically on the topics of *Web usage mining* and *query mining*. To do this, first we will discuss some preliminary data mining techniques, such as clustering and frequent itemset mining, as well as some relevant concepts, such as the vector space model. Next, we will cover the state of the art that is related to this Ph.D. work.

2.1. Preliminary Concepts

In this section we describe two data mining techniques which are used throughout this work: *clustering* and *frequent itemset mining*.

2.1.1. Clustering

Cluster analysis is a technique used to group data into sets of elements that are meaningful and/or useful [94]. In our work, we refer to clustering as a tool to *understand* data, i.e, *clustering for understanding*. In this case, clusters represent classes, or conceptually meaningful groups of objects that have common characteristics. In other words, clustering is a technique to automatically discover classes. When used for non-supervised classification, the clustering process generates labels for each cluster (or class) based only on the data contained in the cluster.

An example of clustering in this context is seen in Information Retrieval. In this case, clustering can be used to group a search engine's results into different categories. Each category representing a particular perspective of the user's query.

The main goal of cluster analysis is to group objects in a way that objects in the same cluster have high similarity to one another, and at the same time are very different from objects in other groups. The greater the homogeneity within a group and at the same time, the difference with other groups, the greater the quality of the clustering solution will be.

There are several types of clustering techniques available. In this work we use *bisecting k-means* [92]. This is a straightforward extension of the *k-means* algorithm [65].

This extension consists of a k-way clustering solution generated by a sequence of $k-1$ *repeated bisections*. In each iteration of this process a cluster is bisected optimizing a *global clustering criterion function*.

There are several global clustering criterion functions that can be used to select which cluster to bisect next, in the clustering process. In this work we use I_1, I_2, H_1 and H_2, as defined in [106]. The formulas for these functions are:

$$\begin{aligned}
\mathcal{I}_1 &= \sum_{i=1}^{k} \frac{1}{n_i} \left(\sum_{v,u \in S_i} sim(v,u) \right), \\
\mathcal{I}_2 &= \sum_{i=1}^{k} \frac{1}{n_i} \sqrt{\sum_{v,u \in S_i} sim(v,u)}, \\
\mathcal{H}_1 &= \mathcal{I}_1 \sum_{i=1}^{k} \frac{1}{n_i} \frac{\sqrt{\sum_{v,u \in S_i} sim(v,u)}}{\sum_{v \in S_i, u \in S} sim(v,u)}, \\
\mathcal{H}_2 &= \mathcal{I}_2 \sum_{i=1}^{k} \frac{1}{n_i} \frac{\sqrt{\sum_{v,u \in S_i} sim(v,u)}}{\sum_{v \in S_i, u \in S} sim(v,u)}
\end{aligned}$$

Where k is the total number of clusters, S is the total objects to be clustered, 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. For more details refer to [60]. The similarity in our case is measured using the cosine function between vectors.

Overall, all existing clustering techniques must rely on four important components [54]: a data representation model, a similarity measure, a cluster model, and a clustering algorithm that uses the data model and similarity measure.

2.1.2. Frequent Itemset Mining

An itemset is a collection of zero or more items that occur together within a same transaction. More formally, let $I = \{i_1, i_2, \ldots, i_d\}$ be the set of all possible items in a data collection, and let $T = \{t_1, t_2, \ldots, t_N\}$ be the set of all transactions. Each transaction t_j contains a subset of items, or itemset, from I [94].

An important property of itemsets are their *support count*, which refers to the number of transactions that contain a specific itemset. This property is used to generate frequent itemsets, in which case it is necessary to find all itemsets that satisfy a *minimum support* threshold. In our research, itemset mining is applied to query analysis using the analogy in which a query submitted to the search engine is a transaction, whose items are the keywords that it contains.

2.1.3. The Vector Space Model

The vector space model is generally used to measure the similarity between documents. It is also known as the *bag-of-words* approach. Many document clustering and classification methods are based on the vector space document model.

The vector space model [83] represents text 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 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 [17]. The similarity between documents is calculated using some measure, such as their cosine similarity. The vector space model does not analyze the co-occurrence of terms inside a document or any type of relationship amongst words.

2.2. Related Work

2.2.1. Web Mining

Web mining [91] is a specific area of Data Mining, and is defined as the process of discovering knowledge, such as patterns and relations, from Web data. Web mining is generally divided into three main areas: *content mining*, *structure mining* and *usage mining*. Each one of these areas are associated the to three predominant types of data found in the Web:

Content: The information that the Web documents were designed to convey. This data consists mainly of text and multimedia.

Structure: The description of the organization of the content within the Web. This includes mainly the hyperlink structure connecting documents and how they are organized in logical structures such as Web sites.

Usage: This data describes the history of usage of a Web site or search engine. This includes click through information, as well as queries submitted by users to search engines. This data is stored in the Web server's access logs, as well as in logs for specific applications.

2.2.2. Web Usage Mining

Web usage mining analyzes the behavior of users according to the data recorded in search engine and Web site access logs. In particular, search engine logs contain queries and are also called *query logs*. In this context, we refer to *query mining* [19] as the study of queries and the behavior of users that generated them.

The information recorded in access logs generally include: the *URLs* requested by visitors, the *time-stamp* of each request, the *user agent*, *IP address* and other parameters sent by users when interacting with the server. Query logs additionally include the *query* *strings* submitted by users in their searches and the URLs viewed and/or selected as a result of each search.

Data Preparation

An important step in usage analysis is the *data preparation* [38, 37, 67]. This step includes, among other things: data cleaning, session identification, merging logs from several applications and removing requests from robots. These techniques are meant to eliminate irrelevant items, so that the resulting associations and statistics reflect accurately the interactions of users with the server.

Other Relevant Concepts

Some common data abstractions used in Web usage mining, are: *users*, *sessions*, *click-streams*, and *page views*. These terms are defined in the W3C Web Characterization Activity (WCA) [6] draft of Web term definitions for analyzing Web usage. In particular, we define the following concepts:

Session: A session is a sequence of document accesses registered for one user in the Web site's usage logs within a maximum time interval between each request. This interval is set by default to 30 minutes, but can be changed to any other value considered appropriate for a Web site [37]. Each user is identified uniquely by the `IP` and `User-Agent`.

Information Scent: Indicates how well a word, or a set of words, describe a certain concept in relation to other words with the same semantics [76]. For example, polysemic words (words with more than one meaning) have less information scent due to their ambiguity.

Queries: A query consists of a set of one or more keywords that are submitted to a search engine and represents an information need of the user generating that query. Queries are very easy to retrieve, from the *referer*[1] field of the usage log of Web sites and explicitly from a request field in search engine query logs.

2.2.3. Web Usage Mining for Web Site Improvement

Web usage mining has generated a great amount of commercial interest [39, 15]. There is an extensive list of work that incorporates Web usage mining for several purposes. One of its applications is in the improvement of Web sites, which has been mostly focused on the support of "adaptive Web sites" [75] and "automatic personalization" [68]. The main purpose of these types of applications is to find interesting rules and patterns in the usage data of Web sites by using data mining techniques such as: analysis of frequent navigational patterns, document clustering, and association rules, based on the pages visited by users [28, 89, 40, 66]. Next, we present a review of these techniques.

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

A Challenge: Adaptive Web Sites

The creation and maintenance of a complex Web site is a very difficult problem, from many different points of view, including interface design. Users have individual goals which may or may not agree with others users' interests. Even worse, a same user might have different needs at different times. Also, as a site grows an evolves, its original design can no longer be appropriate, and a Web site can be initially created with a certain purpose but end up being used in an unexpected way.

Following this motivation Perkowitz and Etzioni present a new challenge for the AI community, that of creating *adaptive Web sites* [75]. They believe that AI techniques can be used to examine user access logs in order to automatically improve a site. Adaptive Web sites are sites *that automatically improve their organization and presentation by learning from user access patterns*. In [75] they divide the adaptation problem into two approaches:

Customization: Which modifies web pages in real time to fit the needs of individual users.

Optimization: Which alters the site itself to make navigation easier for all.

Whether we modify a Web site's pages on-line of off-line, we must use information about users access patterns and the structure of the site. Most of the information needed is available in the server's access logs.

In the process of customization of a Web site the following issue becomes important:

> *Can we formalize user navigation of the Web as a planning process, that is subject to goal recognition? Do user actions on the Web carry enough evidence of their purpose?*

There is much research on this topic, specially in the area of search engine optimization, [34] proposes a taxonomy of Web search, in which users can be classified into three categories according to their intent: *topic relevance task (informational)*, *homepage finding task (navigational)* and *service finding task (transactional)*. For Web sites one of the first projects in this area was the AVANTI Project [51], that relied partly on users providing information about themselves and based on this information it tries to predict both the user's goal and their likely next step.

Optimization on the other hand can be viewed as the problem of finding the "best design" for a Web site, and we can define the current design of a site as a particular point in the vast space of possible designs. Improving the site, then, corresponds to searching in this space for a "better design" than the current one. For this we need a way to measure "better", for example one possible way could be to measure the amount of effort that a visitor needs to find what they are looking for in a site. Effort defined as the number of links that are needed to navigate when reaching this goal. In [74] they sketch the design of a system with a set of transformations that aim to improve a site's organization; transformations include rearranging and highlighting links as well as synthesizing new pages. This system learns from common patterns in the user access logs and decides how to transform the site to explicit those patterns and make the site easier to navigate.

The Adaptive Web site challenge leaves open the following problem:

> *How do we formalize the concept of good design? How do we limit the potential for harm without overly limiting the potential for good?*

If we decide to put the system in the role of adviser to a human expert, then we are faced with the next open question:

> *How does our adaptive Web site communicate its suggestions to a webmaster?*

Web Mining for Web Personalization

As discussed before, Web sites can be personalized to a particular user's needs based on Web usage mining. In [67] *Web personalization* [68] is defined as any action that makes the Web experience of a user customized to the user's taste of preferences. The main elements of Web personalization include modeling of *Web objects*, such as pages or products, and *Web subjects*, such as users or customers. It also includes the categorization of these elements, matching between and across them, and finally the determination of the set of action to be recommended for personalization. The book chapter [67] discusses the mining activities required for this process, presenting a number of specific recommendation algorithms for combining the discovered knowledge with the current status of a user's activity in a Web site to provide personalized content. The approaches and techniques used in Web personalization are categorized into three general groups (as defined in [98, 67]):

Manual decision rule systems: These systems allow Web site administrators to specify rules based on static data, such as user demographics acquired through a registration process, or dynamic data, such as session histories. Those rules are then used to affect the content, the structure or the appearance of the information served to a particular user. Systems that belong to this category are Yahoo!'s personalization engine [5] and Broadvision [7].

Content-based filtering agents: They refer to systems, which rely on content similarity between Web documents and personal profiles. Their success relies on the ability to accurately represent recommendable items in terms of a suitable set of content features and to represent the user profile information in terms of a similar feature set. The relevance of a given item to a specific target user is proportional to the similarity of this item to the user's profile. The main difficulty in applying a content-based method is how to find an appropriate content description language. Also the method may be problematic for new users since their recommendations will be based on the very limited set of items represented in their very immature profiles. A system that fits in that category is WebWatcher [57].

Collaborative filtering systems: They refer to systems that take explicit information in the form of user ratings or preferences and, through a correlation engine, return information that is predicted to closely match the users' preferences. Their basic idea is to draw on the experiences of a population of users, rather than from an individual user profile. Collaborative filtering techniques look for correlation between users in terms of their ratings assigned to items in a user profile. The users with the strongest correlation to the target user are then selected to act as his recommendation partners, and items that occur in their profiles can be recommended to the target user. Systems like Net Perceptions[8] incorporate collaborative filtering techniques.

On their own most of these approaches have drawbacks, such as: the subjectiveness of the description of the users obtained from users themselves which is prone to biases, and in the case of collaborative filtering, the lack of scalability. This is why more recently, Web usage mining was proposed as an underlying approach for Web personalization. The goal of personalization based on Web usage mining is to recommend a set of objects to the current user, possibly consisting of links, ads, text, products, or services, tailored to the user's perceived preferences as determined by the matching usage patterns. This is achieved by matching the current user session with the usage patterns discovered through Web usage mining. These usage patterns in this context are called "aggregate user profiles" because they provide an aggregate representation of the common activities or interests of groups of users. The "recommendation engine" is in charge of this process and is the on-line component of the personalization system.

The key elements of Web personalization include [68]:

1. The categorization an preprocessing of Web data.
2. The extraction of correlations between and across different kinds of such data.
3. The determination of the actions that should be recommended by such a personalization system.

Figure 2.1 represents the architecture of a Web personalization system, as described in [48]. The content management module processes the Web site's content and classifies it in conceptual categories. The Web site's content can be enhanced with additional information acquired from other Web sources, using advanced search techniques. Given the structure and usage logs, a usage miner provides results regarding usage patterns, user behavior, session and user clusters, click-stream information, and so on. Additional information can be obtained through user profiles. All of this information is conceptually abstracted and classified into semantic categories. Any information extracted from the interrelation between knowledge acquired using mining techniques and knowledge acquired from content management will provide the framework for evaluating possible alternatives for restructuring the site. Then a publishing mechanism will preform the changes to the site, so that each user navigates through the optimal site structure.

Most of the commercial tools[10, 9] for log analysis generate reports which include the most frequently accessed pages, average view time of a page, average length and path through a site, common entry and exit points and other statistical characteristics. The problem with these kind of tools is that they lack the ability to find more interesting hidden information. Even so, the information provided by these tools can be very useful to support marketing decision and require little processing power. In the past few years some

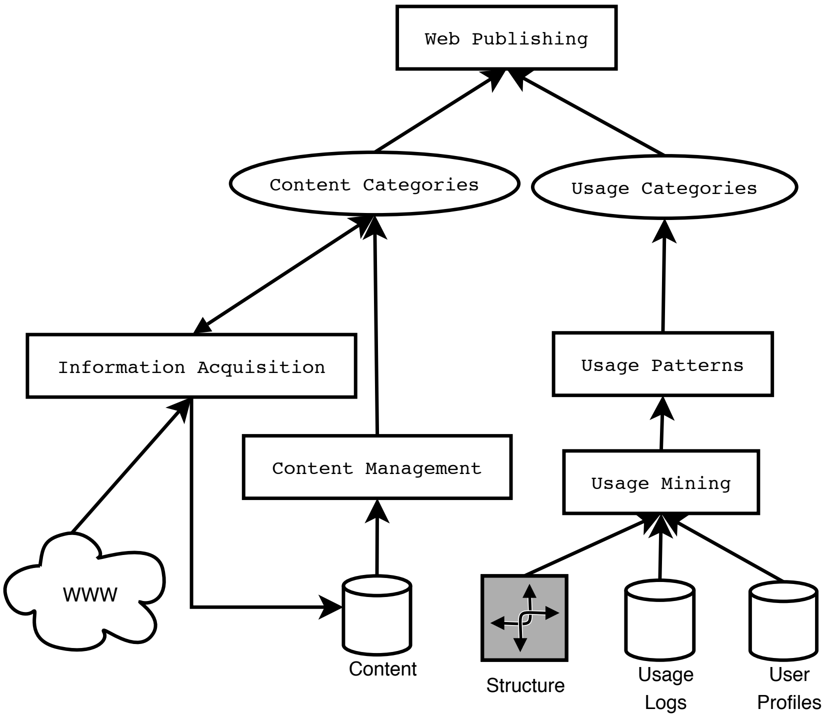


Figure 2.1: Modules for a Web personalization system.

commercial products are incorporating data mining tools to discover deeper knowledge from the usage data, for example [2].

A form of analysis on integrated usage data is OLAP (On-line Analytical Processing). The server log data, can be stored in a multi-dimensional data structure for OLAP analysis [104]. The dimensions analyzed can be based on various fields from the log files, which can include time duration, domain, requested file, user agent, referrers, and so on. This allows the analysis to be performed, for example, on portions of the log related to a specific time interval, or at a higher level of abstraction with respect to the URL path structure. This is applied in the WebLogMiner [104] system where the OLAP technology is used in combination with data mining techniques for prediction, classification, and time-series analysis for Web log data. Another cube model, proposed in [55], explicitly identifies Web access sessions, maintains the order of the session's components and uses multiple attributes to describe the Web pages visited.

Given a set of transactions such as page-views generated by a user in a Web site, a variety of unsupervised knowledge discovery techniques can be applied to obtain patterns in usage. Techniques such as clustering of sessions can lead to the discovery of important user or visitor segments. Other techniques such as item (e.g., page-view) clustering, association rule mining or sequential pattern discovery can be used to find important relationships among items based on the navigational patterns of users in the site. The notion of uncertainty in Web usage mining, discovering clusters of user session profiles using robust fuzzy algorithms was explored by Nasraoui et al. in [71]. In their approach a user or a page can be assigned to more than one cluster. Nasraoui et al. also define new subjective similarity measure between two Web sessions, that captures the organization of a Web site, presented as a new mathematical model for ”robust” Web user profiles and quantitative evaluation means for their validation [72].

In [40, 91] they define Web usage mining as a three step process, consisting of preprocessing, pattern discovery, and pattern analysis. Their prototype system, WebSIFT, preforms intelligent cleansing and preprocessing for identifying users, server sessions, and inferring cached page references through the use of the referer field, and also performs content and structure preprocessing. Pattern discovery is done by using general statistic algorithms and data mining techniques such as association rules, sequential pattern analysis, clustering and classification. Results are analyzed with a simple knowledge query mechanism, a visualization tool or the information filter.

Association rules are used to capture the relationships among items based on their patterns of co-occurrence across transactions (without considering the ordering of items). In the case of Web transactions, association rules capture relationships among page-views based on the navigational patterns of users. The result of association rule mining can be used in order to produce a model for recommendation or personalization systems [69].

Sequential patterns in usage data capture the page trails that are frequently visited by users, in the order that they were visited. Sequential patterns are those sequences of items that frequently occur in a sufficiently large proportion of transactions. In the context of Web usage data, contiguous sequential patterns can be used to capture frequent navigational paths among user trails [90]. Frequent item sets, discovered as part of association rule mining, represent the least restrictive type of navigational patterns, since they focus on the presence of items rather than the order in which they occur within user session. Markov models are especially suited for predictive modeling based on contiguous sequences of events. In the context of Web transactions, Markov chains can be used to model transition probabilities between page-views. In Web usage analysis, they have been proposed as the underlying modeling machinery for Web prefetching applications or to minimize system latencies [45, 77]. Such systems are designed to predict the next user action based on a user’s previous navigational behavior. Markov models can also be used to discover high probability user navigational trails in a Web site. In [66] they apply association rules and sequential pattern discovery on Web log files and then use this to customize the server hypertext organization dynamically, their prototype system is WebTool, which also provides a visual query language in order to improve the mining process.

Another way of efficiently representing navigational trails is by inserting each trail into a trie structure [90]. It is also possible to insert frequent sequences (after or during sequential pattern mining) into a trie structure [73]. A well-known example of this approach is the notion of aggregate tree introduced as part of the WUM (Web Utilization Miner) system [90]. The aggregation service of WUM extracts the transactions from a collection of Web logs, transforms them into sequences, and merges those sequences with the same prefix into the aggregate tree (a trie structure). A mining language MINT was developed for the implementation of WUM, which allows the extraction of navigation patterns of interest obtained with the mining system. STRATDYN [27] is an add-on module for WUM that extends its capabilities by identifying the differences between navigation patterns and exploiting the site’s semantics in the visualization of the results. This approach uses concept hierarchies as the basic method of grouping Web pages together.

One of the most advanced systems is WebPersonalizer [68], and it provides a framework for mining log files to provide recommendations to users based on similar browsing patterns presented by other users. It performs several data mining techniques such as association rules, sequential pattern discovery, clustering, and classification, to find interesting usage patterns. The recommendation engine matches each user's behavior to provide a list of recommended hypertext links.

2.2.4. Query Mining

Past query mining research has focused in improving technical aspects of search engines. Nevertheless, query analysis can impact Web search and Web design from two different points of view [19]: Web findability and information scent. Web findability is a measure of how easy it is to find a site on the Web using search engines. One example, of improving a Web site's findability using queries, is by including very popular search strings inside the main contents of the site. On the other hand, the most common queries are usually the ones with more information scent, so analyzing queries can help find the words that describe better the contents of the site.

Queries provide a very precise insight into what are the users' motivations for visiting certain documents. In this area most of the work has been directed at using queries to enhance Web site search [102] and to make more effective global Web search engines [20, 21, 59, 87]. In particular, in [82] chains (or sequences) of queries with similar information needs are studied to learn ranked retrieval functions for improving Web search. Additionally, queries have also been studied to improve the quality of a Web site. Previous work on this subject include [44] which proposed a method for analyzing similar queries on Web search engines. The idea of [44] is to find new queries that are similar to the ones that directed traffic to a Web site, and later use this information to improve the Web site.

2.2.5. Document Modeling

Web usage mining has also been applied to discover information about Web documents, based on the information provided implicitly by users. This helps improve tasks such as, automatically clustering, labeling and classifying Web documents.

Traditionally Web documents were analyzed using their structure and their contents: HTML formatting, sometimes allows to identify important parts of a document, such as title and headings, and link information between pages [52]. Nevertheless, 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 Web site, 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 Web sites 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 the site. Also, structural information does not reflect what users find more interesting in a Web page. It only reflects what webmasters find interesting. On the other hand, the straightforward analysis of the contents of Web documents do not always convey what users find interesting or why they do visit certain Web pages. This is why the study of usage data brings new information about Web documents. Some related work in this area is [107] which combines many data sources to solve navigation problems in Web sites. Their approach consists in clustering Web documents based on their link similarities by using visitor's navigation paths as weights to measure semantic relationships between Web pages. Also, in [85] 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. There are several issues which make clustering of Web documents extremely challenging, mostly due to the large size of collections and dimensionality of the data. In general, most clustering and classifications methods for documents are based on the *vector space* document model. In this context, 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 [54] 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 [26] 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 [79] 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* [100] provided by WordNet[2]. In [95] 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 [47] also uses extraction of keywords based on frequency and techniques such as using *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* into their models.

Also, user queries have been studied to improve clustering of Web documents. This idea, to the best of our knowledge, has only been considered previously in [25], [35] and [80]. In [25] 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. On the other hand, the works presented in [35] and [80] are quite similar. They only vary in the source of the query log

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

that they use and the weight assigned to the features in their vectors. In [35] they present the *query view* model for Web site document representation. This model uses queries from a site's internal search engine as features to model documents, the weight of each feature is the frequency of the query. In [80] they introduce the *query vector model*, which uses search engine queries to model the documents in a collection to improve document selection algorithms for parallel information retrieval systems. The model represents each document as a vector of queries weighted by the search engine rank of the document for each particular query in the feature space. The approach of [80] *only uses the query log to extract queries* and does not use the click through information of the documents clicked for each query. By doing so [80] only takes information from the search engine rank algorithm, and not from the users click through data.

Chapter 3

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

3.1. Introduction

As the amount of contents in the Web grow, it becomes increasingly difficult to manage and classify its information. Optimal organization of Web documents is important for Web sites as well as for heterogeneous sets of documents. For example, in a Web site, the classification of documents into cohesive and relevant topics is essential to make the site easier to navigate and more intuitive to its visitors. Also, the results presented by search engines can be enhanced by grouping documents into significant topics. These topics can allow users to disambiguate or specify their searches quickly, improving Web IR [99]. 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 benefit from automatic topic discovery and classification are human edited directories, such as DMOZ [3] or Yahoo! [4]. Directories such as these, are increasingly hard to maintain. Furthermore, automatic organization of Web documents is very important from the point of view of *discovering new interesting topics*. This allows Web content providers to keep up with user's trends and changing interests.

It is important to point out that an automatic approach to the problem of organizing Web documents is highly relevant. Mostly because of the high competition in the Web, which makes it necessary for content providers to improve in a fast, effective and scalable fashion. As mentioned in Chapter 2, the task of automatically clustering, labeling and classifying documents in the Web is not easy. 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.

Since neither content or structural data seem to provide all the necessary information for the task of clustering, labeling and classification of Web documents, we analyze the incorporation of *usage data*, obtained from the logs of a Web site or a search engine. In particular, we suggest to use the information provided by user queries. 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 the motivation of improving general Web document clustering and labeling, we propose a new document representation model, which 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*. 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 Web site.

The main contributions of this chapter 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* [35] 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 Web site, general Web documents and search engine results.
- We also present an initial experimental evaluation to corroborate our models.

Our work is based on the idea that search queries and their clicked results provide an indication of the relevance of documents to queries. From this point of view, our approach is similar to several related works discussed in Chapter 2. In particular, to that presented in [25], but it differs in that we consider the *queries from which documents are clicked* are *good summaries* of user's intent when viewing a document. We also use global search engine queries (and not only internal searches as in [35]) as surrogate text for documents and choose the document's features from the terms of the queries from which it was clicked from. Furthermore, in our model we extend [35] by using as features frequent sets of queries, in a similar way to [26] and [79], with the difference that they extract patterns from the original full text of the document (and not considering user feedback from queries).

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

3.2. Document Clustering and Labeling

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 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 Web site can be completely different of what user's would expect. This would make navigation difficult and documents hard to find for visitors, see Chapter 5. 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 judgments of the real users and not the search engines judgments (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 Web site's access logs. This situation is most likely when organizing documents belonging to a particular Web site. 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 Web site (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.

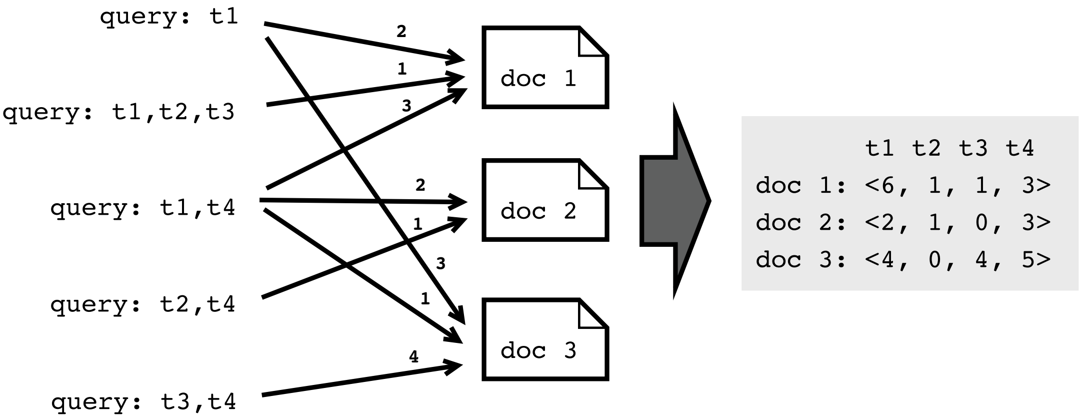


Figure 3.1: Example of the query document representation, without normalization.

3.2.1. 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 [35]. We extend [35] 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.

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 Web site 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 3.1 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 $C_{ij} = 0, \forall j$.

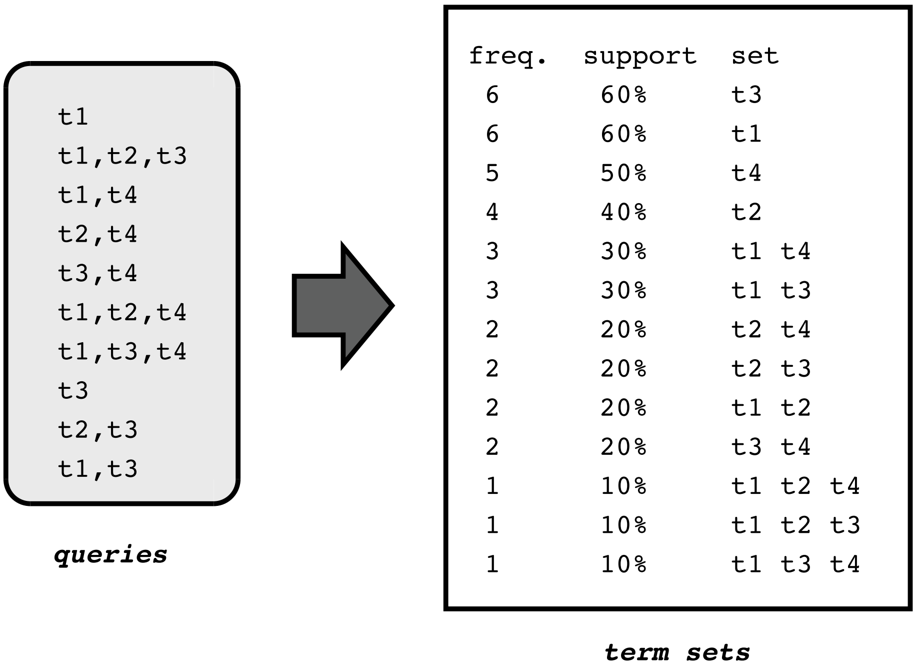
Although we use queries for modeling documents, this approach differs from [80] in the following way: the queries considered for document features are *only* those from which users clicked on the document as a result of the query in the search engine. This way, implicit user feedback is used to relate 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.

3.2.2. Query-Set Document Model

The main drawback of 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 [26, 79] 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



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

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

Figure 3.2: Example of all the term sets found for a group of queries and their supports.

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 threshold level, for the support of each pattern size, avoids the creation of very large feature spaces. 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 3.2 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. A real world example is shown in Figure 3.5 of Section 3.3, in which the support threshold is set by discarding any *trivial* patterns, appreciated in the first bins of each histogram. In this example the result is a 90% reduction of the feature space in comparison with the one generated by the vector space model.

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.

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 term 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 $Q'(d_i)$.

3.2.3. 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 [79]), 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).

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 Web site. In addition, there are many techniques available to cluster and label text documents.

3.3. Evaluation and Discussion

As a first evaluation for our query-based models we chose to use a Web site with its access logs. This decision was based on the fact that by using a Web site'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 Web site 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 Web site gathers a great amount of visits and contents from a broad spectrum of educational institutions. Table 3.1 shows some general statistics of the one month log used for this evaluation. Table 3.2 describes how the different documents in the Web site were *found* by users (i.e., clicked on for the first time in a session). Documents in Table 3.2 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),

General Log Statistics

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

Table 3.1: General log statistics for the Web site.

	% 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 3.2: General statistics on how users reached the documents of the Web site.

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 3.3 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 [15]. 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 3.4 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 3.4. 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 and they capture a large fraction of the queries to the site (see Table 3.1). 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.

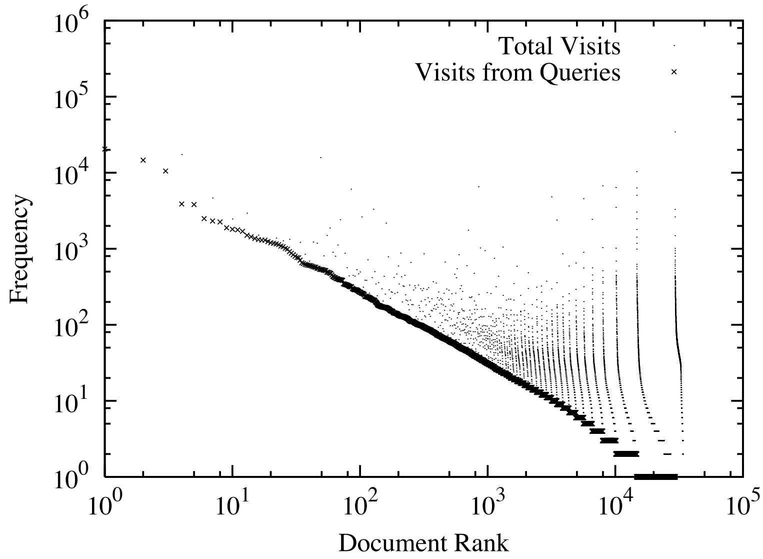


Figure 3.3: Distribution of query clicks and of visits to documents of a Web site (documents ranked by # of queries then by # of visits).

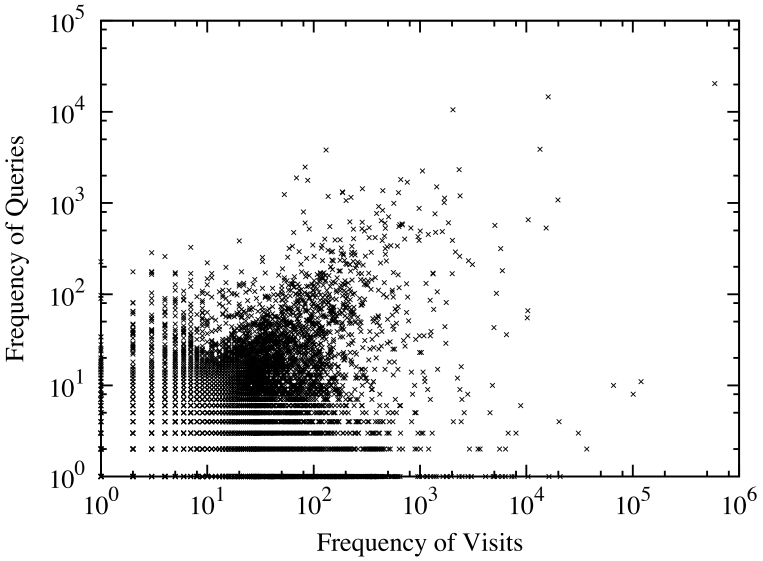


Figure 3.4: Scatter plot of the frequency of queries and the frequency of navigational visits in a Web site (each dot represents a document from the site).

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 [37], 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.

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

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 [60]. 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 3.3. 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 $\mathcal{I}_2$. This function was experimentally found appropriate for document clustering, as discussed in [60]. 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.

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 [84]. This was done for patterns with 1, 2, 3, 4 and 5 terms, to obtain the support level for each case. Figure 3.5 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 3.4 shows a summary of the resulting support table.

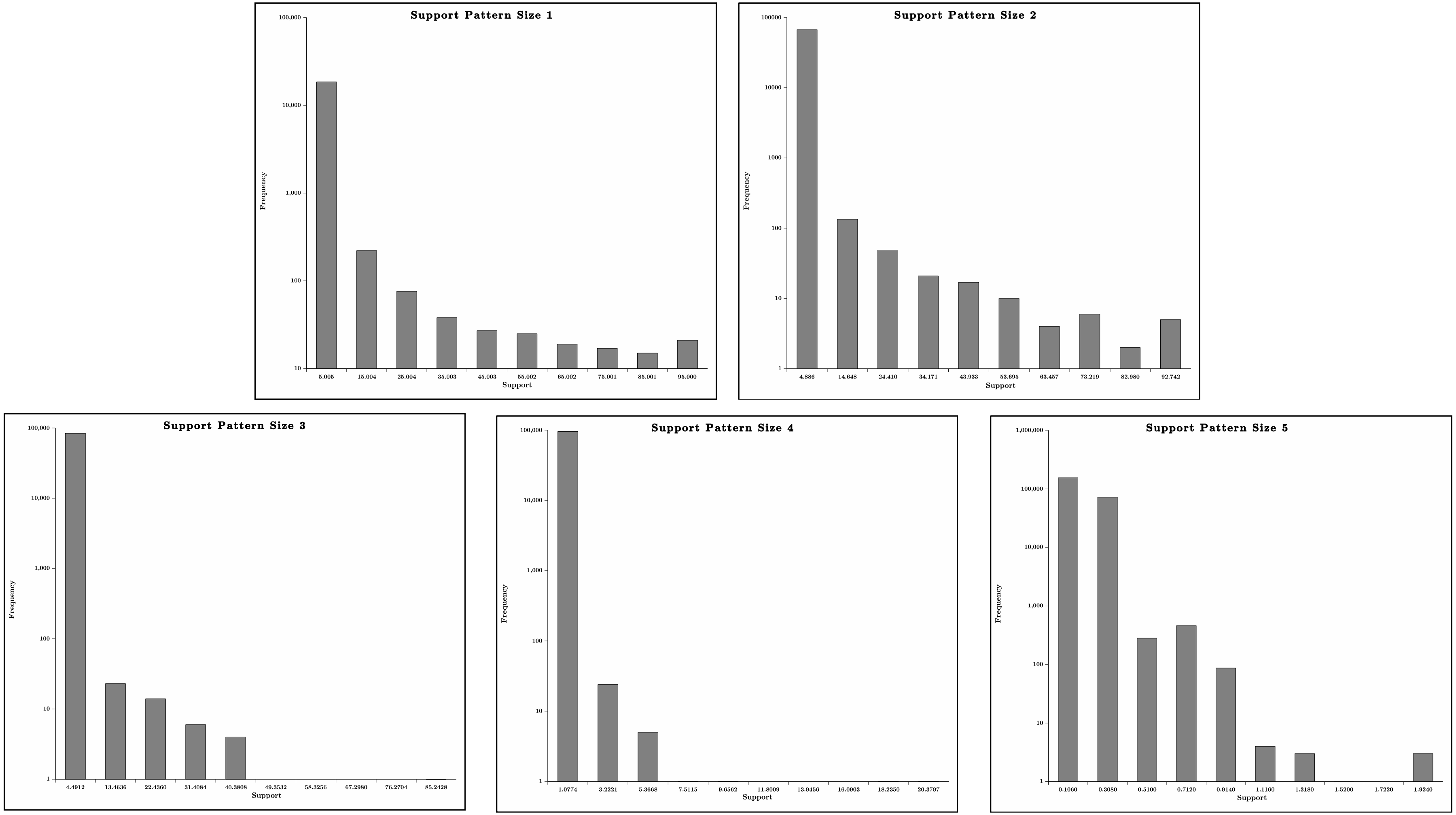


Figure 3.5: Support graphs for different pattern sizes.

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

Table 3.4: 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 3.5: Experimental results for each document model.

In Table 3.5 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 3.7 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).

In Table 3.5 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:

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

where

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

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 3.6 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 3.6: Probability of random inter-judge agreement, with $j = 3$ and $w = 0.5$.

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

3.4. Chapter Conclusions and Future Work

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 Web site. 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 dimensionality 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 the quality of 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 Web site 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 Web sites, such as the one presented in Chapter 5. 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 [22, 46]. 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.

Chapter 4

Discovering Similar Web Sites Using Search Engine Queries

4.1. Introduction

In this chapter we present an application and extension of Chapter 3. To do this we shift the focus of Web IR from the traditional retrieval of Web documents that satisfy a certain query, towards the retrieval of complete Web sites.

In general terms, a Web site is a network of related pages which together aim at providing a specific service or covering a particular topic. Web site retrieval encompasses several applications (which are not satisfied fully by document retrieval), such as: (1) retrieving sites that are relevant to a specific query, (2) retrieving sites that are similar to a given site, (3) grouping similar sites, and (4) building groups of sites, where each group is representative of a specific topic.

The first application refers to the retrieval of Web sites (rather than individual documents) which are representative of a topic expressed in a user's query. This is very useful in the case of *informational queries* [34], according to which a user searches for a good collection of links on a topic, rather than only one good document. The collection of documents that satisfies the query may be a dedicated Web site or a relevance-ranked list of Web sites. Examples of informational queries are "Web data mining", "Barcelona housing", "cats" and "exotic travel". Even in the case of *transactional queries* [34], which aim to satisfy a transactional need, such as e.g. "on-line bookstore" or "on-line travel agency", a whole Web site can be a more appropriate result than a single document. The challenge is to express how relevant a site is for a subject.

The second application refers to using a Web site as a *query* to search for Web sites that are similar to it. This is relevant, for example, in the context of *competitive intelligence*, where companies are interested in finding their (possibly unknown) competitors and how they present themselves on-line. However, Web sites that cover the same topic (e.g. "on-line bookstores") may differ greatly in size, structure, vocabulary and content. Therefore, similarity between Web sites needs to be modeled in a way that is independent of these differences. Most important, similarity in this case does not include mirror sites or near duplicate sites, it rather focuses on sites that fulfill similar user needs. This is also useful for search engines since they could offer a "similar Web sites" next to each result.

The last two applications mentioned, involve the identification of groups of Web sites on a same topic. The topics may be given *a priori*, as in (4), or derived *a posteriori* from the resulting groups of similar sites, as in (3). Currently, this type of problem is solved through human-intensive initiatives such as the DMOZ [3] and Yahoo! [5] directories: Which, for a given topic hierarchy, manually find and assign Web sites that fit best each category. This is usually done by associating one document from the site to the matching category. Obviously, this approach does not scale well in a rapidly growing and heterogeneous environment like the Web, since it requires knowledge of all possible topics and regular inspection of all of the top Web sites for each topic. Furthermore, human editors may leave out relevant Web sites, or fail to see the similarity among sites when analyzing only one or two documents in each candidate site.

We focus on these last two issues, i.e. creating an automatic approach for building groups of similar Web sites. Which, as mentioned before, encompasses the challenge of defining site similarity in a way that is independent of the structure, size and vocabulary of Web sites. Additionally, we believe that these two problems are a basis for solving (1) and (2).

Our solution to the Web site similarity problem is based on the *wisdom of the crowds* concept, conveyed concretely through queries of search engine users. We consider *two Web sites as being similar if users access both of them to satisfy similar information needs*. This approach allows us to identify Web sites that are similar according to their *perceived information content*, independent of their structure, size, vocabulary and specific data content.

Our methodology for modeling and discovering similar Web sites can be summarized as follows: For each document in a Web site, we identify the queries used to access it. The set of all of the queries used to access the documents constitutes a giant *query-based feature space*, in which each Web site is expressed as a vector. Query frequency and association among queries can be used for feature space reduction to avoid the *curse of dimensionality* problem. We study different techniques, extended from our earlier work on modeling individual documents as vectors over a query-based feature space, described in Chapter 3. We then use clustering to build groups of similar vectors/sites.

Our contribution is twofold: First, we address the problem of finding similar Web sites as an extension of conventional Web IR. For this problem, we propose a solution that is not sensitive to structural and linguistic discrepancies among Web sites. Second, we deal with the curse of dimensionality problem that emerges when modeling Web sites on the basis of all possible terms that can be associated with them. Our approach is based on association rule discovery and results in much smaller feature spaces that allow for high-quality clustering of similar Web sites.

This chapter is organized as follows: In Section 4.2 we discuss related work. Section 4.3 describes our model of a Web site as a vector over feature spaces of queries. Section 4.4 describes how we cluster similar site-vectors. The experimental section follows, in which we evaluate clusters of Web sites built with different feature spaces on the basis of external and internal cluster quality criteria. The last section concludes our study with a summary of the findings and an outlook.

4.2. Related Work

In this section we want to make the distinction between *similar* document and *duplicate* or *near-duplicate* document detection. Although there is extensive work in the latter area, such as [33, 86], it pursues a different goal than similarity research. Most of the work on duplicate or near-duplicate detection, models documents based on their contents. This enables the detection of documents with *very high resemblance* to one another, i.e. almost identical documents. On the other hand, similarity research as we understand it in this work, seeks to find documents that satisfy similar information needs from users, but are not necessarily alike in contents, extension or vocabulary. From now on we will only discuss similarity related research.

The work presented in [49] is more related to that of finding similar Web sites. They present two methodologies to classify sites based on the topics of their documents. In one, they view a Web site as one large HTML-document and create a feature space of topics, to this they apply a well-known classification technique. In the second, they model Web sites as trees of topics and use a Markov tree model for classification. The second method results in better classification.

Many document clustering and classification methods are based on the *vector space* document model, this is discussed in detail in Chapter 3. This work takes as a starting point the query-set model presented in Chapter 3, extending it to model Web sites for clustering. It should be noted that in our work, we do not compare our Web site model to that presented in [49], since they were designed for different tasks, one for supervised classification and the other for non-supervised classification. Nevertheless, our model differs from [49] in the use of queries as surrogate text for website modeling. Ideally in future work a comparison of these two models for clustering, as well as for classification will be performed.

4.3. Modeling Web Sites

We first discuss different ways of defining the notion of "Web site" and then go ahead to model Web sites as vectors, by extending the traditional vector space model for documents. Our framework is generic and allows different Web site models, but our focus is on modeling a site as a vector over a query-based feature space. In the last portion of this section, we elaborate on methods to build and reduce the size of query-based feature spaces.

4.3.1. Defining the Concept of "Web Site"

What constitutes a Web site? Providing a formal and unambiguous definition is still an open problem [29]. Intuitively, a site is expected to cover a broad topic. Sometimes, a site is implicitly expected to deal with exactly *one* topic, although this might not always be satisfied in practice. Even for sites that address a single broad topic, an association between the informal concept "Web site" and a formal entity, as e.g. webserver, is not straightforward.

For example, consider the site of a faculty of Computer Science. We expect that this site addresses the topic "computer science" as an area of research and as an educational subject. Although humans can recognize the coherence among the documents offered by the CS faculty site, defining a formal border of what constitutes the site is more difficult: Some faculties have their own webserver, while others may have their Web site hosted at the university's server. On the other hand, larger faculties may use multiple servers, e.g. have one Web site per department or accommodate different contents on different hosts.

In this work, we adhere to a simple heuristic definition of *Web site* as the set of documents that appear under the same host name; this agrees with the Web site definition in [29]. In terms of our example above, this definition means that the CS faculty documents are part of a Web site, only if they are located under a single host name that does not contain other documents. If the CS faculty rather organizes its documents in a subdirectory of the university's webserver, then the faculty is not a Web site on its own, but a subsite within the Web site of the university. This last situation is heterogeneous; it highlights the fact that many sites refer to multiple topics without being truly representative for any of them. Nevertheless, this heuristic gives an approximation of the concept of Web site, which is adequate for the purpose for our study on site similarity.

4.3.2. A Framework for Web Site Vectorization

We extend the traditional *vector space model* for documents to a model that represents a Web site as a vector. This vector aggregates the information of the site's documents. However, this information does not necessarily consist of the terms in the documents' contents. Our approach allows for an arbitrary feature space and we concentrate on features that are *sets of terms from queries*, rather than terms from text.

More formally, let $\mathcal{D} = \{d_1, d_2, \ldots, d_n\}$ be a collection of documents and let $\mathcal{F} = \{f_1, f_2, \ldots, f_m\}$ be the set of features that describe those documents. Further, let $w_{i,j}$ the *weight* of feature f_j for document d_i. Then, d_i is represented as a vector:

$$\overrightarrow{d_i} =< w_{i,1}, w_{i,2}, \ldots, w_{i,m} > \tag{4.1}$$

Equation 4.1 is a generalization of the vector space document model (or bag-of-words), towards the incorporation of an arbitrary feature space. In the bag-of-words representation, $\mathcal{F}$ would be the set of terms in $\mathcal{D}$ and $w_{i,j}$ would be the weight of the j^{th}-term in the i^{th}-document - possibly the term frequency in the document, normalized by the inverse document frequency in $\mathcal{D}$.

We use the vector representation of documents as a basis for a vector representation of a Web site. In particular, let $SITES = \{s_1, s_2, \ldots, s_N\}$ be a set of Web sites and let $\mathcal{D}$ be the union of all documents in all sites, i.e. $s_k \subseteq \mathcal{D}$ for $k = 1, \ldots, N$. Then, a Web site s_k is a vector:

$$\overrightarrow{s_k} =< c_{k,1}, c_{k,2}, \ldots, c_{k,m} > \tag{4.2}$$

where $c_{k,j}$ is the normalized weight of feature $f_j \in \mathcal{F}$ for site s_k and is computed as follows: Let $w'_{k,j}$ be the sum of the weights of all documents in s_k for feature f_j, i.e. $w'_{k,j} = \sum_{d_i \in s_k} w_{i,j}$. Then, $c_{k,j}$ is the normalized counterpart of $w'_{k,j}$ according to a specific *tf-idf* scaling scheme proposed in [17, 60]:

$$c_{k,j} = \left(0.5 + \frac{0.5w'_{k,j}}{\max_{f_l \in \mathcal{F}}(w'_{k,l})}\right) \times \left(-log_2 \frac{n_j}{N}\right) \tag{4.3}$$

where $\max_{f_l \in \mathcal{F}}(w'_{k,l})$ is the feature with the largest weight in site s_k and n_j is the number of sites where feature f_j appears.

In our framework, the vectorization of a Web site $s_k \in SITES$ requires:

1. The specification of the *feature space* $\mathcal{F}$ over the documents constituting all the sites in $SITES$.
2. The *weighting scheme* for the features over the documents needs to be specified.

Then, the sites' vectors are computed according to Equation 4.2 and Equation 4.3. Next, we derive "query-based Web site models", i.e. we model Web sites as vectors over feature spaces of sets of query terms.

4.3.3. Query-Based Web Site Models

We model Web sites using feature spaces that reflect how the sites are *perceived* by users. For this, we derive features from the queries registered in search engine logs. We concentrate on *successful* queries, i.e. on queries that resulted in a click to a document from the sites under study (the $SITES$ set defined in the previous section). Even though not all queries that produce a click on a document are actually successful, we reduce the noise produced by error in this process by considering the total volume of clicks in the log for each (query, document) pair.

Our approach for query-based Web site modeling is an extension of the work presented Chapter 3 on query-sets document modeling. We extract queries from usage logs in a search engine and apply itemset discovery to queries to create feature spaces of sets of *query-terms* (i.e. keywords in queries).

Query-Set Mining

As mentioned in Chapter 3, we refer to *query-set mining* as the discovery of *query-sets*, which are *sets of query-terms* extracted from individual queries. More formally, let $\mathcal{L}$ be the query log of the search engine and let $\mathcal{Q}$ be the set of *unique* queries in $\mathcal{L}$. Therefore, each query $q \in \mathcal{Q}$ can be repeated one or more times in the query log $\mathcal{L}$. For a document $d \in \mathcal{D}$, we denote as $\mathcal{Q}(d)$ the set of unique queries in $\mathcal{Q}$ that resulted in a request for d, and as $\mathcal{L}(d)$ the subset of the query log $\mathcal{L}$ that contains requests for d. Further, we denote as $QT(d)$ the set of individual query-terms in $\mathcal{Q}(d)$. Now, we distinguish among the following mining tasks:

1. *Extraction of query-sets:* The input to this mining task are $\mathcal{Q}(d)$ and the set of query-terms $QT(d)$ for document d. The output is the set of all frequent query-sets, subject to a support threshold τ. We denote the output set as $QS(d, \tau)$.

2. *Extraction of maximal query-sets:* This mining task is similar to the above, except that only maximal length query-sets for d are retained; their frequent subsets are discarded. We denote the set maximal query sets for d as as $\widehat{QS}(d, \tau)$.

According to the principles of itemset discovery, the (absolute) support of an itemset x is the number of transactions containing all of the items in x. Similarly, the support of a query-set qs for document d is the number of queries in $\mathcal{L}(d)$ that contain the terms in qs. Equivalently, the support of qs for document d is the sum of the clicks of each distinct query $q \in \mathcal{Q}(d)$ such that $qs \subseteq q$. We denote this support as $clicks(qs, d)$.

Alternatively and for completeness, we use a similar notation $clicks(q, d)$ to refer to the total number of occurrences of a *query* q within $\mathcal{L}(d)$ (i.e., the total number of clicks from query q to document d). Next, we use the queries, query-terms and query-sets extracted per document to build a feature space for documents and Web sites in $SITES$.

Query-Based Feature Spaces for Web Sites

We model each site s in a given set of Web sites $SITES$, as a vector over a feature space consisting of elements that are either *queries*, *query-terms* or *query-sets*. We consider the following alternative feature spaces:

- "QUERYTERMS Model": The feature space $\mathcal{F}$ consists of all individual *query-terms* that constitute the queries leading to documents in the $SITES$:

$$\mathcal{F} = \cup_{s \in SITES} \left(\cup_{d \in s} QT(d) \right).$$

- "FULLQUERIES Model": The feature space $\mathcal{F}$ consists of complete *queries*, namely the queries used to access the documents in $SITES$:

$$\mathcal{F} = \cup_{s \in SITES} \left(\cup_{d \in s} \mathcal{Q}(d) \right).$$

- "FREQPATTERNS Model": The feature space $\mathcal{F}$ consists of the frequent *query-sets* for each document in $SITES$, subject to threshold τ:

$$\mathcal{F} = \cup_{s \in SITES} \left(\cup_{d \in s} QS(d, \tau) \right).$$

  The support thresholds for our study are discussed in Section 4.5.2.

- "FULLPATTERNS Model": The feature space $\mathcal{F}$ consists of all *query-set* elements for all documents, i.e. the support threshold τ is zero:

$$\mathcal{F} = \cup_{s \in SITES} \left(\cup_{d \in s} QS(d, 0) \right).$$

- "MAXPATTERNS Model": The feature space $\mathcal{F}$ consists of all *maximal query-sets* for the documents in $SITES$, i.e. the frequency threshold to zero as above:

$$\mathcal{F} = \cup_{s \in SITES} \left(\cup_{d \in s} \widehat{QS}(d, 0) \right).$$

- "FULLQUERIESPLUS Model": The feature space $\mathcal{F}$ contains for each document d those *query-sets* for which there is a query in $\mathcal{Q}$ (not necessarily in $\mathcal{Q}(d)$), i.e. independently of whether this query resulted in a request for d:

$$\mathcal{F} = \cup_{s \in SITES} \left(\cup_{d \in s} (QS(d, 0) \cap \mathcal{Q}) \right).$$

The intuition behind the FULLQUERIESPLUS Model is to keep only the query-sets that actually represent a query formulated by a user. This selects the relevant query-sets used to model documents from the users' point of view.

For the vectorization of a Web site over the alternative feature spaces, we also need the weights of the features for the individual documents. Let f_j be a feature, i.e. a query-term, a query-set or a complete query, depending on the feature space we use. The weight of f_j for a document $d \in \mathcal{D}$ is either: a) the number of queries in $\mathcal{L}(d)$ that *contain* f_j, in the case that f_j is a query-terms or query-set, or b) the number of queries in $\mathcal{L}(d)$ that *match exactly* f_j, in the case that f_j is a query. In other words, the weight of each f_j for a document d is $clicks(f_j, d)$ as defined in Section 4.3.3. Then, the unnormalized weight of feature f_j for the site $s_k \in SITES$ is the sum:

$$w'_{k,j} = \sum_{d \in s_k} clicks(f_j, d)$$

where the normalized weight $c_{k,j}$ can be computed according to Equation 4.3.

Using the alternative feature spaces and the common vectorization scheme of Section 4.3.2, we model Web sites as vectors over query-sets.

4.4. Clustering Web Sites

With the purpose of evaluating our approach for Web site modeling to find similar Web sites, we apply existing clustering techniques, using the vector representation generated by each model. In particular we use *bisecting k-means*, described in Chapter 2, as a clustering method. This follows the experimental evaluation used in Chapter 3, but is extended to generate different solutions, for the global clustering functions I_1, I_2, H_1 and H_2. In this case k is set in agreement with the dataset used (see Section 4.5.3).

To evaluate the quality of each clustering solution we use the well-known measures of *entropy* and *purity*, as described in [106]. It should be noted that good clustering solutions are those that generate high purity and low entropy values. The formal definitions for these metrics are:

Given a particular cluster S_r of size n_r, the entropy of this cluster is defined to be

$$E(S_r) = -\frac{1}{\log q} \sum_{i=1}^{q} \frac{n_r^i}{n_r} \log \frac{n_r^i}{n_r}$$

where q is the number of classes in the dataset, and n_r^i is the number of clusters in the ith class that were assigned to the rth cluster. The entropy of the clustering solution is then defined as

$$Entropy = \sum_{r=1}^{k} \frac{n_r}{n_r} E(S_r).$$

Similarly, the purity of a cluster is defined as

$$P(S_r) = \frac{1}{n_r} \max_i (n_r^i)$$

and the overall purity of the clustering solution is given by

$$Purity = \sum_{r=1}^{k} \frac{n_r}{n} P(S_r).$$

4.5. Experimental Evaluation

For the experimental evaluation of our approach on clustering similar Web sites, we study the different query-based feature spaces and vectorization models presented in Section 4.3.3. We compare the clustering results they achieve to those yielded by different usage-based feature selection methods and also to the baseline *bag-of-words* model. We denote this baseline as the "TEXT model" hereafter. To evaluate the quality of the clusters, we use internal quality indicators and external measures. For the latter, we use the DMOZ directory as a "gold standard".

For the query log processing, implementation of the Web site models and clustering solutions, we use only publicly available software. The tools fastutils [30] and WebGraph [31] are used to extract the query related information for each document from the query log. These are extremely fast tools, that reduce computational costs considerably. The tool LPMINER [84] is used to extract the query-sets, and for generating the clusters we use CLUTO [60].

4.5.1. Dataset

As a data source we use a sample of the Yahoo! UK query log, which corresponds to a series of contiguous request during a certain period of time.

Since our models are based on usage data, we use only the URLs and Web sites which are registered in the query log. This is, URLs and Web sites that were clicked by a user as a result to a query. Additionally, in order for the dataset to be appropriate to evaluate our models we keep only the *URLs* that match all of the following restrictions (in this order):

1. Have been clicked at least two times.
2. Belong to a Web site which is listed in only *one* DMOZ category.
3. Belong to a Web site which has at least three other URLs in the dataset.

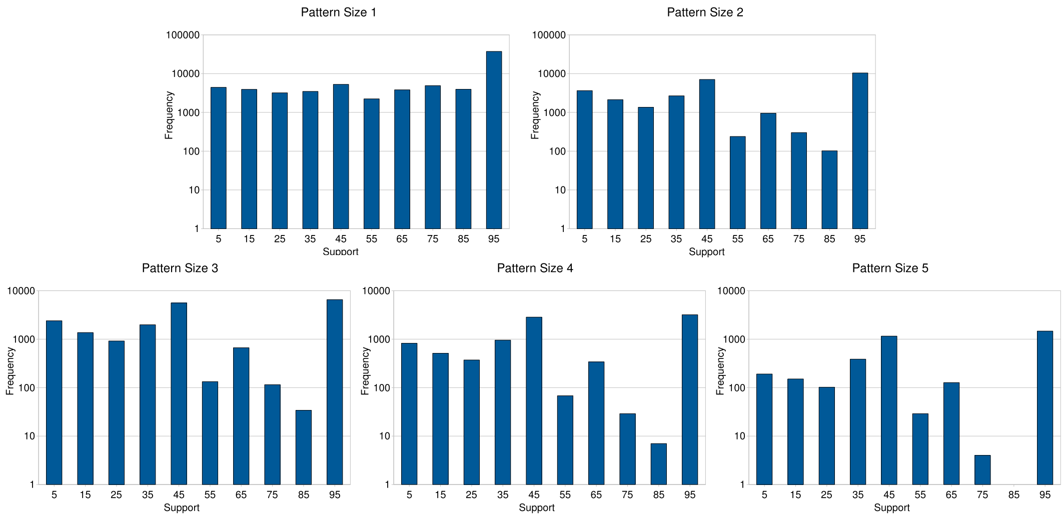


Figure 4.1: Histogram for the support distributions of the Web documents in the query log sample.

4. Belong to a DMOZ category that contains URLs (in the dataset) that belong to least three other Web sites.

The previous restrictions on the URLs used are set mainly to assure that there is enough usage information to model and cluster sites. Restriction 1) is set to remove URLs which do not have much usage information. Restriction 2) is set to remove ambiguity in the DMOZ evaluation and work only with Web sites that are clearly classified into one DMOZ category. Restriction 3) makes sure that each Web site has at least three URLs to build the model. In restriction 4) we make sure that each category has at least three elements Web sites in it. Note that this does not mean that the models are not appropriate to represent URLs and Web sites that do not meet these criteria. This is only used to simplify this initial evaluation. In our log sample we worked with 977 Web sites that contained $5,070$ URLs that met these restrictions, classified into 216 DMOZ categories.

4.5.2. Building the Feature Spaces

To build the feature spaces according to the models in Section 4.3.3, we have performed query-set mining. In Figure 4.1, we present the distribution of the support values for each pattern size in the log. In this figure we observe a heterogeneous distribution of the supports per pattern size. Additionally, all of the histograms show a *peak* in the 95% support bin. We believe that this occurs because the current dataset shows an aggregation of an heterogeneous set of Web sites. This includes Web sites that have a *long-tail* in their query distribution (i.e., patterns with low support and high frequency), and others with the opposite situation (i.e., frequent patterns with high support). For this reason, setting a global support threshold for each pattern size is not appropriate in this case. This is does not occur in Chapter 3 since only one Web site was used in the dataset. Therefore, we have set the threshold to zero and skipped the FREQPATTERNS model from now on.

4.5.3. Evaluation with External Cluster Quality Measures

Next, we present the results obtained for the different clustering solutions regarding an external cluster quality indicator, in this case DMOZ categories. In this study, we consider the DMOZ categories to be the *real* categories of the Web sites. Therefore, we measure the quality of the clustering solutions against this "gold standard".

Categories in DMOZ follow a tree hierarchy (see example in Figure 4.2), this allows to establish different levels of granularity in the classification of Web sites. Categories can thus range from general to very specific. In this evaluation we considered several different levels of the DMOZ hierarchy. We evaluated by truncating the tree at levels 3 (45 categories), 4 (75 categories), 5 (104 categories) in the hierarchy tree and also using the complete category branches for each site (216 categories). The quality of each clustering solution is measured using the solution's entropy and purity. The methodology used for the evaluation was the following:

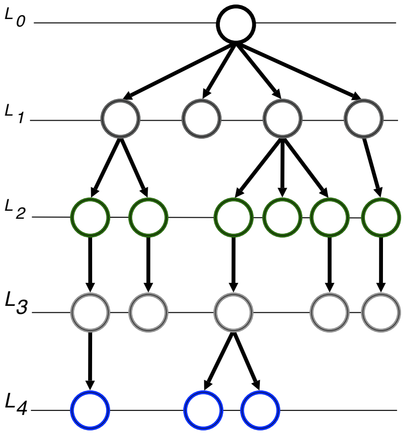


Figure 4.2: Example of directory hierarchy levels.

1. For one of the Web site models described in Section 4.3.3, generate the corresponding representation for all the sites in the dataset.

2. Label each of the Web site representations with the DMOZ category it belongs to.

3. Cluster the Web sites into as many clusters as DMOZ categories exist in the dataset, i.e. *select k according to the number of categories* .

4. Obtain the *entropy* and *purity* measures of the clustering solution.

5. Repeat for each different Web site model in Section 4.3.3.

The global clustering functions considered in the evaluation were I_1, I_2, H_1 and H_2, described in Section 4.4. We evaluate the solutions generated by each one of these functions, to see which one produces the best overall results for this type of data. Then we use only this function to compare our Web site models.

The results obtained from this clustering process are shown in Tables 4.1, 4.2 (the best values are in bold). For selecting the best clustering function, we prioritize the purity values as a first selection criteria and then use the entropy values, as the second criteria. Our interest is focused on grouping together documents from the same class. Table 4.1 shows that the best overall clustering solutions are given by I_1 and I_2 and that the best results according to these functions are obtained for the FULLQUERIESPLUS clustering model. It is important to observe that the FULLQUERIESPLUS model outperforms the TEXT model using only the 5% of the number of features (see Table 4.4).

Table 4.2 shows that the best solutions according to the average entropy values are those produced by I_2 and H_1. Therefore, combining the results from Tables 4.1 and 4.2, we select I_2 as the *most adequate global clustering function for our evaluation*. The values shown in Table 4.2 show that the best solution according to the entropy values for I_2 are also those of the FULLQUERIESPLUS Web site model.

Figures 4.3 and 4.4 show the purity and entropy values of the Web site models using I_2, while varying the number of clusters. Note that we compare all of the models, excluding FULLQUERIES and MAXPATTERNS. These models were excluded because they did not produce solutions in which most of the Web sites were clustered, therefore their results were not comparable with the rest. This occurs because the FULLQUERIES and MAXPATTERNS models produced, in some cases, representations that did not share any features with other sites. This is probably due to the long-tail distribution of queries in some Web sites. Nevertheless, to still compare these models with the rest, we repeat the evaluation *only* on the set of Web sites that could be clustered using all models. Then, we re-evaluate using the baseline model, TEXT, and the winner of the previous evaluation FULLQUERIESPLUS. The results are shown in Table 4.3 and show that the FULLQUERIESPLUS model preforms better than the baseline model, using a significant amount of less features (see Table 4.5). Figures 4.5 and 4.6 show the behavior of the entropy and purity measures as we vary the number of clusters and classes using the different DMOZ levels.

Purity(I_1)

No. of Clusters	*46*	*75*	*104*	*216*	*Avg.*
FULLPATTERNS	0.732	0.711	0.600	0.514	0.639
FULLQUERIESPLUS	0.757	0.718	0.591	0.545	**0.653**
QUERYTERMS	0.744	0.710	0.596	0.526	0.644
TEXT	0.720	0.699	0.582	0.571	0.643

Purity(I_2)

No. of Clusters	*46*	*75*	*104*	*216*	*Avg.*
FULLPATTERNS	0.734	0.697	0.566	0.516	0.628
FULLQUERIESPLUS	0.749	0.716	0.594	0.550	**0.652**
QUERYTERMS	0.739	0.686	0.581	0.528	0.634
TEXT	0.724	0.700	0.580	0.548	0.638

Purity(H_1)

No. of Clusters	*46*	*75*	*104*	*216*	*Avg.*
FULLPATTERNS	0.703	0.683	0.577	0.508	0.618
FULLQUERIESPLUS	0.731	0.722	0.588	0.554	**0.649**
QUERYTERMS	0.733	0.671	0.576	0.519	0.625
TEXT	0.724	0.696	0.601	0.554	0.644

Purity(H_2)

No. of Clusters	*46*	*75*	*104*	*216*	*Avg.*
FULLPATTERNS	0.718	0.674	0.550	0.498	0.610
FULLQUERIESPLUS	0.723	0.691	0.571	0.527	0.628
QUERYTERMS	0.716	0.679	0.561	0.514	0.618
TEXT	0.728	0.701	0.580	0.532	**0.635**

Table 4.1: Purity values for I_1, I_2, H_1 and H_1.

4.5.4. Evaluation with Internal Cluster Quality Measures

In this section we analyze and compare the internal cluster quality measures produced by the different Web site models. The first measure used is the *overall value of the global clustering function*, in this case I_2, of the resulting solution. The second set of measures are: the *average similarity between objects in each cluster*, i.e. internal similarities (*ISim*), and the *average similarity of the objects of each cluster and the rest of the objects*, i.e. external similarities (*ESim*).

Overall the results obtained according to these metrics indicate that the lowest performance is always that of the TEXT model and that the FULLQUERIESPLUS model, on the other hand, always generates better solutions.

Figure 4.7 show the values for I_2. Since the clustering method attempts to optimize the global clustering function, the best solutions are those that achieve greater values for I_2.

Table 4.6 and Figures 4.8 and 4.9 show the ISim and ESim values obtained for the solutions generated using I_2. Here we can observe that the ISim values for the FULLQUERIESPLUS model are quite higher than the values obtained for the other models. Which shows that the FULLQUERIESPLUS model leads to clusters in which elements

Entropy(I_1)

No. of Clusters	*46*	*75*	*104*	*216*	*Avg.*
FULLPATTERNS	0.269	0.235	0.277	0.210	0.248
FULLQUERIESPLUS	0.205	0.183	0.238	0.186	**0.203**
QUERYTERMS	0.256	0.222	0.256	0.188	0.231
TEXT	0.257	0.211	0.239	0.167	0.219

Entropy(I_2)

No. of Clusters	*46*	*75*	*104*	*216*	*Avg.*
FULLPATTERNS	0.214	0.194	0.234	0.189	0.208
FULLQUERIESPLUS	0.187	0.176	0.220	0.182	**0.191**
QUERYTERMS	0.214	0.195	0.226	0.181	0.204
TEXT	0.206	0.182	0.214	0.167	**0.192**

Entropy(H_1)

No. of Clusters	*46*	*75*	*104*	*216*	*Avg.*
FULLPATTERNS	0.234	0.205	0.232	0.189	0.215
FULLQUERIESPLUS	0.192	0.174	0.221	0.180	**0.192**
QUERYTERMS	0.215	0.202	0.230	0.184	0.208
TEXT	0.206	0.185	0.210	0.164	**0.191**

Entropy(H_2)

No. of Clusters	*46*	*75*	*104*	*216*	*Avg.*
FULLPATTERNS	0.223	0.199	0.238	0.188	0.212
FULLQUERIESPLUS	0.201	0.188	0.226	0.187	0.201
QUERYTERMS	0.221	0.199	0.228	0.183	0.208
TEXT	0.202	0.177	0.217	0.172	**0.192**

Table 4.2: Entropy values for I_1, I_2, H_1 and H_1.

Purity(I_2)

No. of Clusters	46	75	104	216	Avg.
FULLQUERIESPLUS	0.794	0.773	0.718	0.773	**0.765**
FULLQUERIES	0.813	0.770	0.723	0.746	0.763
MAXPATTERNS	0.784	0.763	0.694	0.739	0.745
TEXT	0.790	0.803	0.691	0.731	0.754

Entropy(I_2)

No. of Clusters	46	75	104	216	Avg.
FULLQUERIESPLUS	0.138	0.124	0.132	0.067	**0.115**
FULLQUERIES	0.159	0.137	0.133	0.072	0.125
MAXPATTERNS	0.189	0.148	0.149	0.075	0.140
TEXT	0.140	0.107	0.137	0.074	**0.115**

Table 4.3: Purity and entropy values for the reduced set of Web sites using I_2.

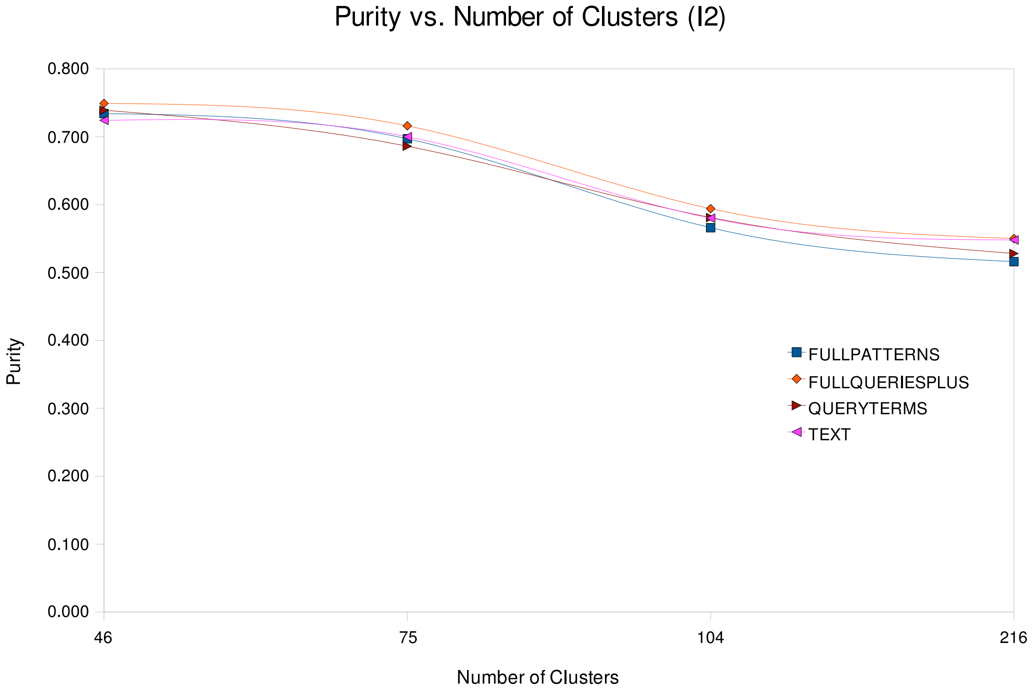


Figure 4.3: Purity vs. the number of clusters for I_2.

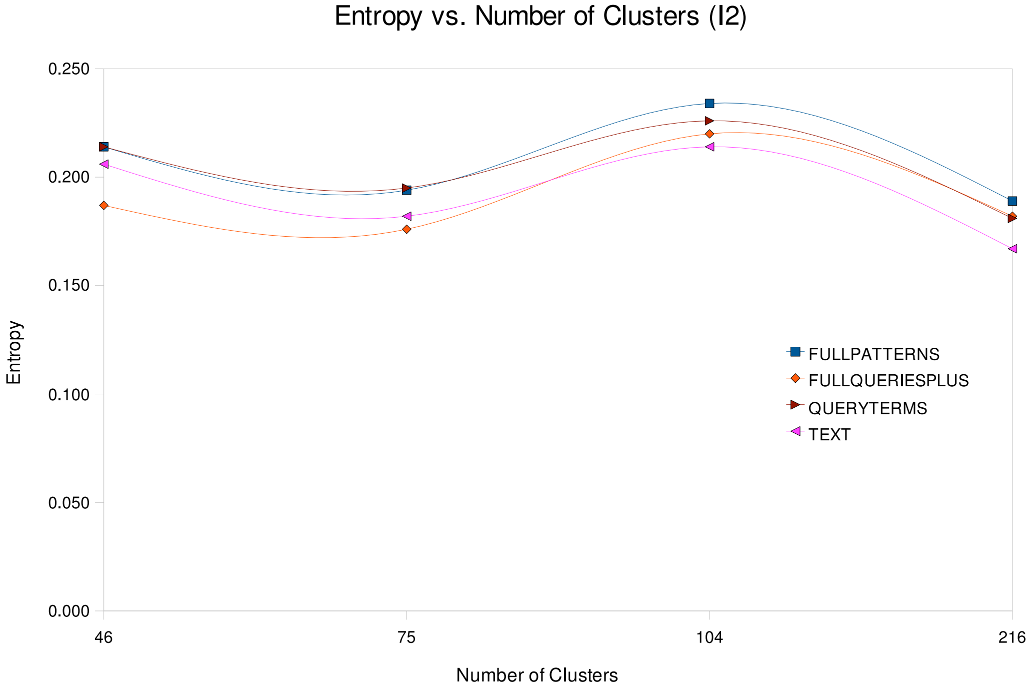


Figure 4.4: Entropy vs. the number of clusters for I_2.

are more similar to each other. The ESim values on the other hand favor the FULLPATTERNS model, followed closely by the rest of the query-based Web site representations.

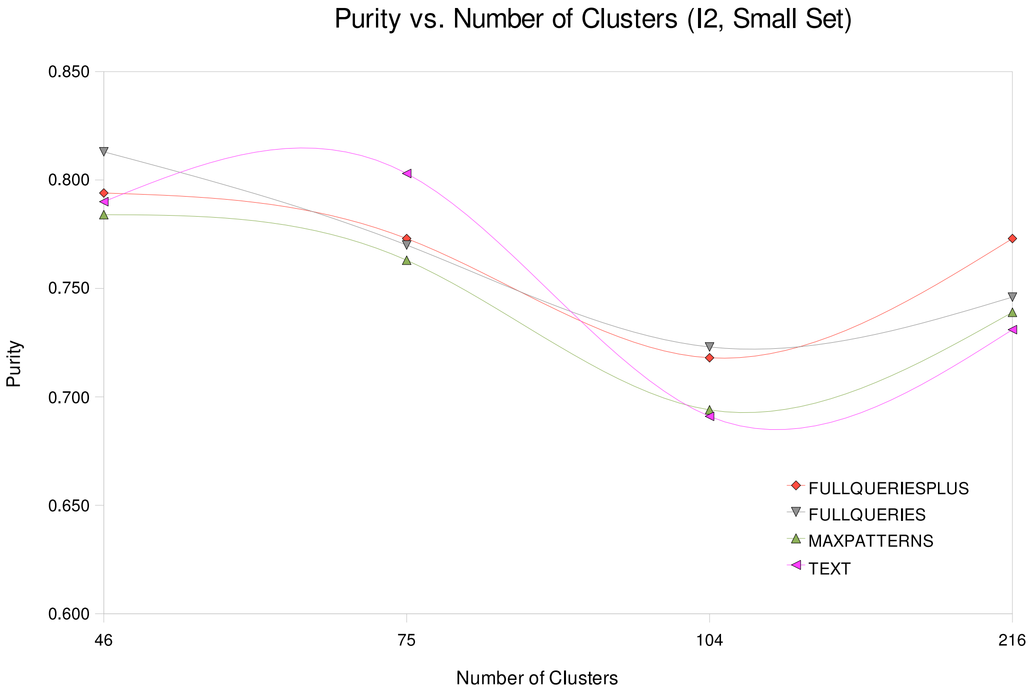


Figure 4.5: Purity vs. the number of clusters for I_2.

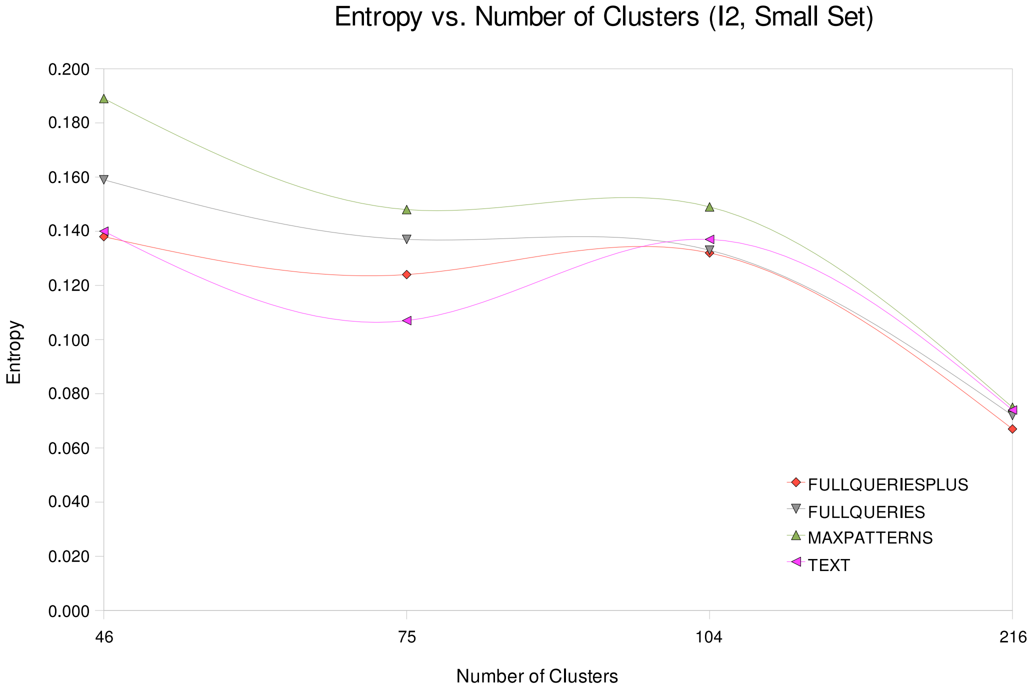


Figure 4.6: Entropy vs. the number of clusters for I_2.

Statistical testing was performed for the clustering solutions generated for $k = 216$ and I_2. The results are statistically significant, using a two tailed t-test, at the $p < 0.001$ level for ISim and ESim, and at the $p < 0.05$ level for Purity, and at the $p < 0.01$ level for

Model	No. of Features
FULLPATTERNS	56,929
FULLQUERIESPLUS	**8,957**
QUERYTERMS	6,763
TEXT	178,449

Table 4.4: Number of features of each model.

Model	No. of Features
FULLQUERIESPLUS	5,674
FULLQUERIES	5,673
MAXPATTERNS	6,913
TEXT	109,973

Table 4.5: Number of features of each model (reduced set of documents).

the Entropy measures. In this estimation we compared the baseline case, TEXT, with the FULLQUERIESPLUS model.

4.6. Chapter Conclusions and Future Work

In this work we focus on generating document and Web site models which convey users' information needs when visiting Web documents from search engines. The purpose of this is to find similarities between Web sites independently of discrepancies such as content, structure and size. To achieve this we analyze the usage data registered in queries submitted to search engine query logs.

In particular, we present a general framework to extend the vector space document model to arbitrary feature spaces. Then we adapt this framework to model Web sites with the specific purpose of using query-based feature spaces. Specifically, we generate several different Web site models which use as features queries, query-terms and query-sets.

Our experimental evaluation shows that the query-sets obtained from a search engine query log, cannot be trimmed in the same way as for those sets obtained from a particular Web site. The reason is that the approach of using supports to reduce the number of query-sets does not provide a solution for an heterogeneous set of Web sites. This indicates that different Web sites have different support distributions depending on the characteristics of each site. Therefore we seek alternative space-reduction techniques for query-sets, shown in our query-based Web site models.

The usefulness of our Web site models was measured applying clustering to the Web site vectors, with the objective of discovering groups of similar sites. Our experimental evaluation shows that the query-based approaches use significantly less features than the full text approach obtaining better results. Secondly, the FULLQUERIESPLUS Web site model is overall the Web site representation that produces the best clusters according to the internal and external quality measures used in the evaluation. This is an impor-

ISim(I_2)					
No. of Clusters	*46*	*75*	*104*	*216*	*Avg.*
FULLPATTERNS	0.133	0.196	0.255	0.438	0.256
FULLQUERIESPLUS	0.266	0.373	0.469	0.674	**0.445**
QUERYTERMS	0.156	0.218	0.269	0.448	0.273
TEXT	0.146	0.191	0.235	0.375	0.237

ESim(I_2)					
No. of Clusters	*46*	*75*	*104*	*216*	*Avg.*
FULLPATTERNS	0.005	0.005	0.005	0.005	**0.005**
FULLQUERIESPLUS	0.007	0.007	0.007	0.008	0.007
QUERYTERMS	0.007	0.007	0.007	0.008	0.007
TEXT	0.027	0.027	0.027	0.028	0.027

Table 4.6: ISim and ESim values for I_2.

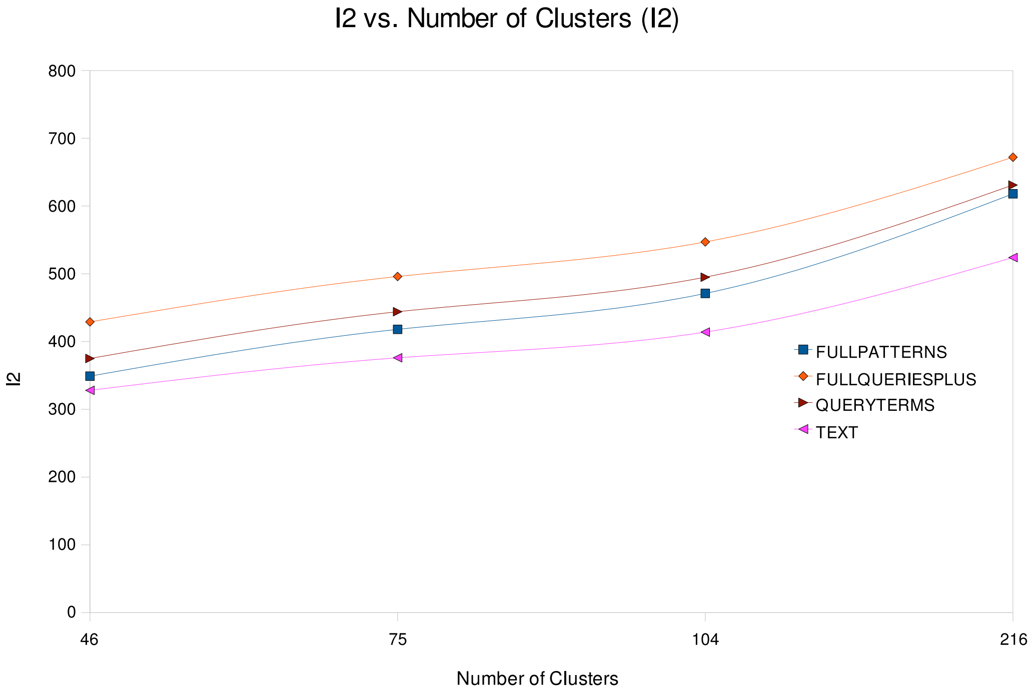


Figure 4.7: I_2 values vs. the number of clusters.

tant result, especially considering that the FULLQUERIESPLUS model uses only 5% of number of features than the full-text approach.

Future work will be focused on the different frequency distributions for query-sets found in Web sites. With the purpose to analyze interesting information that this data can provide about each Web site.

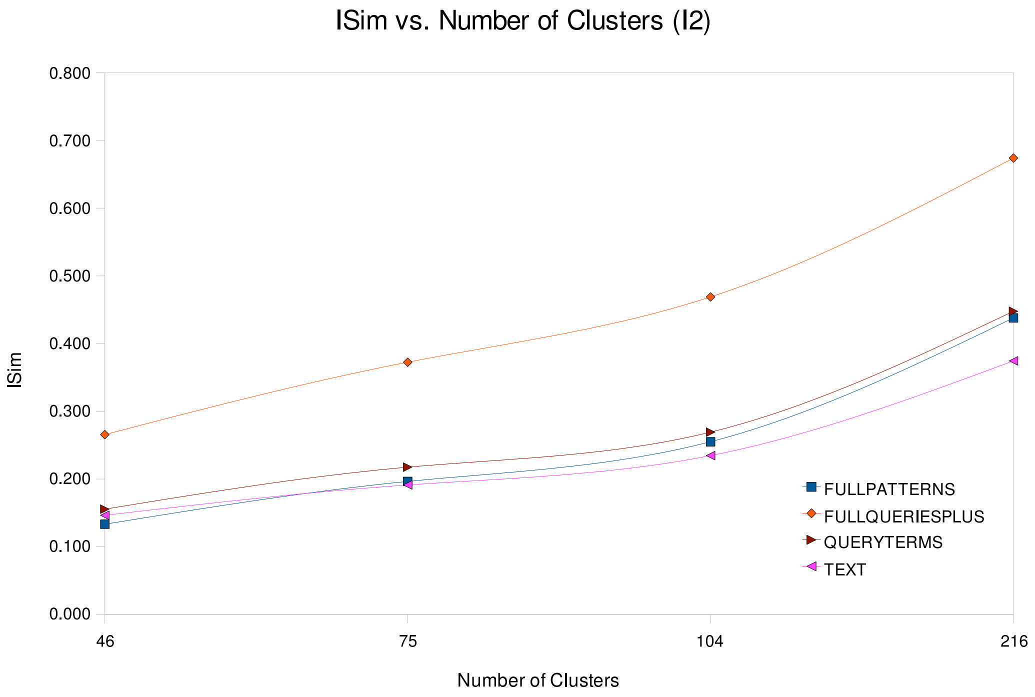


Figure 4.8: ISim vs. the number of clusters for I_2.

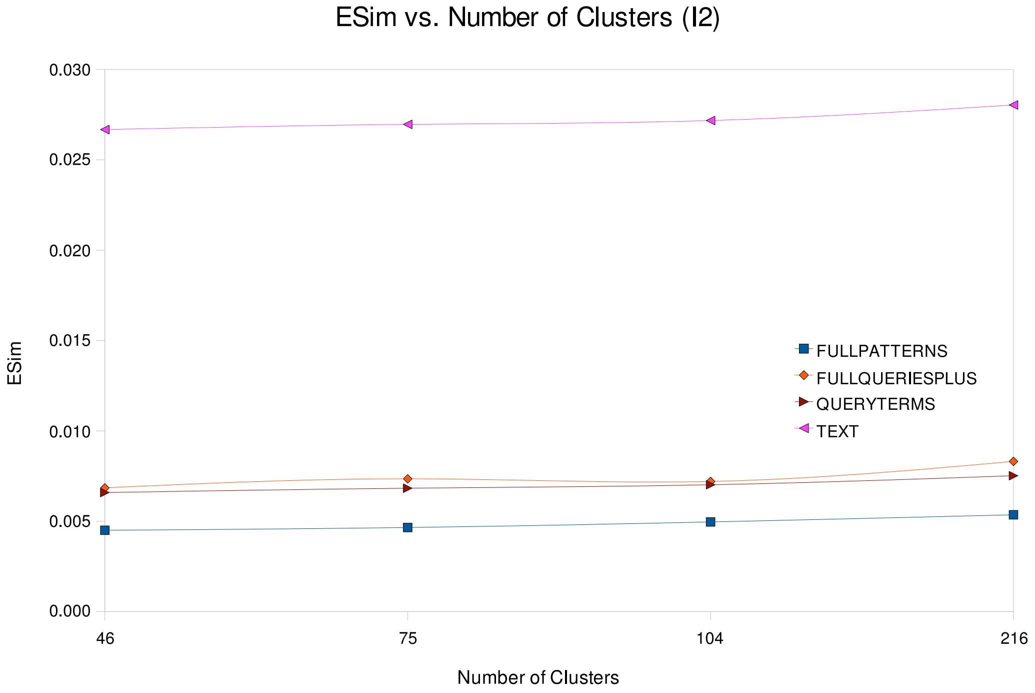


Figure 4.9: ESim vs. the number of clusters for I_2.

Chapter 5

A Web Site Mining Model Centered on User Queries

5.1. Introduction

In previous chapters we have studied the effectiveness of search engine queries to enhance general Web document representation and Web site models. In this chapter we look into the usefulness of queries to improve a particular Web site and the documents it contains.

In general, queries in Web mining have been studied with the purpose of enhancing Web site search, and not with the intention of discovering new data to increase the quality of the Web site. In this chapter we present a novel model that mines queries found in the usage logs of a Web site, classifying them into different categories based on navigational information. These categories differ according to their importance for discovering new and interesting information about ways to improve the site. Our model also generates a visualization of the site's content distribution in relation to the link organization between documents, as well as the URLs selected due to queries. This model was mostly designed for Web sites that register traffic from internal and/or external search engines, even if this is not the main mechanism of navigation in the site. The output of the model consists of several reports from which improvements can be made to the Web site.

The main contributions of our model for improving a Web site are: *to mine user queries within a Web site's usage logs*, *obtain new interesting contents to broaden the current coverage of certain topics in the site*, *suggest changes or additions to words in the hyperlink descriptions*, and at a smaller scale *suggest to add new links between related documents and revise links between unrelated documents in a site*.

We have implemented this model and applied it to different types of Web sites, ranging from small to large, and in all cases the model helps to point out ways to improve the site, even if this site does not have an internal search engine. We have found our model specially useful on large sites, in which the contents have become hard to manage for the site's administrator.

Queries found in a site's usage logs can be studied to improve the quality of a Web site. Queries made to the Web site's internal search engine (if one is available) and also the queries on external search engines that resulted in requests of documents from the Web site, queries account for a large portion of the visits of most sites on the Web. This kind of analysis, is presented in [14] and consists of studying queries submitted to a site's internal search engine, and indicating that valuable information can be discovered by analyzing the behavior of users in the Web site after submitting a query. This is the starting point of our work in this chapter.

This chapter is organized as follows. Section 5.2 presents our Web site mining model, Section 5.3 gives an overview of our evaluation and results. Section 5.4 presents our chapter conclusions and future work.

5.2. Model Description

In this section we will present the description of our model for mining Web site usage, content and structure, centered on queries. This model performs different mining tasks, using as input the Web site's access logs, its structure and the content of its pages.

In our model the structure of the Web site is obtained from the links between documents and the content is the text extracted from each document. The aim of this model is to generate information that will allow to improve the structure and contents of a Web site, and also to evaluate the interconnections amongst documents with similar content. Our model analyzes two different types of user queries, that can be found in a Web site's access registries. These queries are:

External queries: These are queries submitted on Web search engines, from which users selected and visited documents in a particular Web site. They can be discovered from the log's `referer` field.

Internal queries: These are queries submitted to a Web site's internal search box. Additionally, external queries that are specified by users for a particular site, will be considered as internal queries for that site. For example, Google.com queries that include `site:example.com` are internal queries for the Web site `example.com`. In this case we can have queries without clicked results.

Figure 5.1 (left) shows the description of the model, which gathers information about internal and external queries, navigational patterns and links in the Web site to discover IS that can be used to improve the site's contents. Also the link and content data from the Web site is analyzed using clustering of similar documents and connected components. These procedures will be explained in more detail in the following sections.

5.2.1. Navigational Model

By analyzing the navigational behaviors of users within a Web site, during a period of time, the model can classify documents into different types, such as: *documents reached without a search*, *documents reached from internal queries* and *documents reached from external queries*. We define these types of documents as follows:

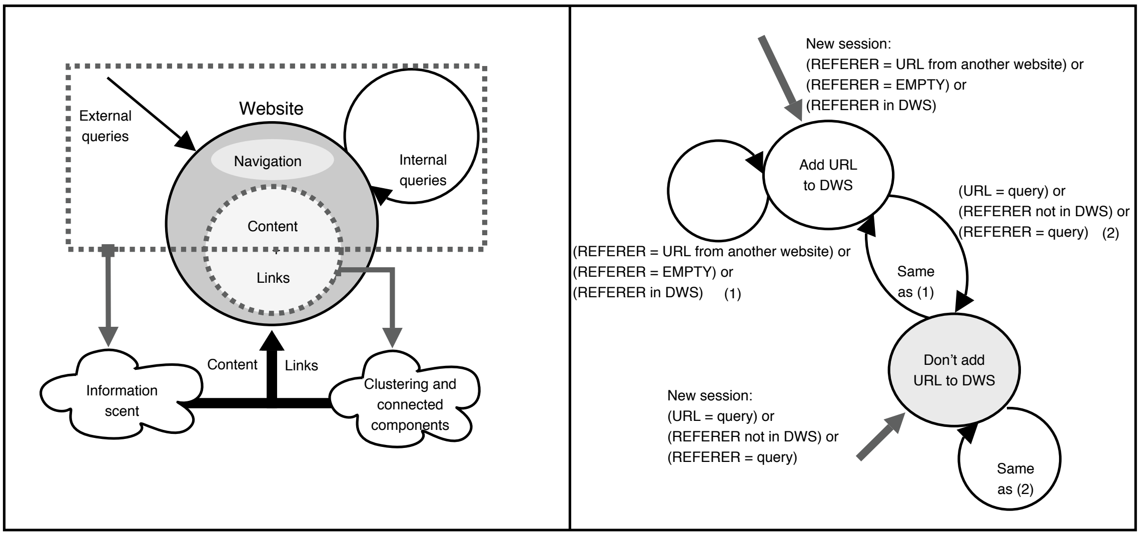


Figure 5.1: Model description *(left)* and heuristic for DWS *(right)*.

Documents reached Without a Search (DWS): These are documents that, throughout the course of a session, were reached by browsing and without the interference of a search (in a search engine internal or external to the Web site). In other words, documents reached from the results page of a search engine and documents attained from those results, are *not* considered in this category. Any document reached from documents visited previously to the use of a search engine will be considered in this category.

Documents reached from Internal Queries (DQ$_i$): These are documents that, throughout the course of a session, were reached by the user as a direct result of an *internal query*.

Documents reached from External Queries (DQ$_e$): These are documents that, throughout the course of a session, were reached by the user as a direct result of an *external query*.

For future references we will drop the subscript for DQ$_i$ and DQ$_e$ and will refer to these documents as *DQ*.

It is important to observe that DWS and DQ are *not disjoint sets of documents*, because in one session a document can be reached using a search engine (therefore belonging to DQ) and in a different session it can also be reached without using a search engine. The important issue then, is to register *how many times* each of these different events occur for each document. We will consider the frequency of each event directly proportional to that event's significance for improving a Web site. The classification of documents into these three categories will be essential in our model for discovering useful information from queries in a Web site.

Heuristic to Classify Documents.

Documents belonging to DQ sets can be discovered directly by analyzing the referer URL in an HTTP request to see if it is equal to the results page of a search engine (internal or external). In these cases only the *first occurrence* of each requested document in a session is classified. On the other hand, documents in DWS are more difficult to classify, due to the fact that backward and forward navigation in the browser's cached history of previously visited documents is not registered in web servers usage logs. To deal with this issue we created the heuristic shown in Figure 5.1, which is supported by our empirical results. Figure 5.1 (right) shows a state diagram that starts a new classification at the beginning of each session and then processes sequentially each request from the session made to the Web site's server. At the beginning of the classification the set DWS is initialized to the value of the Web site's start page (or pages) and any document requested from a document in the DWS set, from another Web site or from an empty referer (the case of bookmarked documents) are added to the DWS set.

5.2.2. Query Classification

We define different types of queries according to the outcome observed in the user's navigational behavior within the Web site. In other words, we classify queries in relation to: if the user chooses to visit the generated results and if the query had results in the Web site. Our classification can be divided into two main groups: *successful queries* and *unsuccessful queries*. Successful queries can be found both in internal and external queries, but unsuccessful queries can only be found for internal queries since all external queries in the Web site's usage logs were successful for that site.

Successful Queries.

If a query submitted during a session had visited results in that same session, we will consider it as a successful query. There are two types of successful queries, which we will call A and B. We define formally classes A and B queries as follows (see Figure 5.2):

Class A queries: Queries for which the session visited one or more results in *AD*, where AD contains documents found in the DWS set. In other words, the documents in AD have also been reached, in at least one other session, browsing without using a search engine.

Class B queries: Queries for which the session visited one or more results in *BD*, where BD contains documents that are only classified as DQ and not in DWS. In other words documents in BD have *only* been reached using a search in all of the analyzed session.

The purpose of defining these two classes of queries, is that A and B queries *contain keywords that can help describe the documents that were reached as a result of these queries*. In the case of A queries, these keywords can be used in the text that describes links to documents in AD, contributing additional IS for the existing link descriptions to these documents. The case of B queries is even more interesting, because the words used for B queries describe documents in BD better than the current words used in link descriptions to these documents, contributing with new IS for BD documents. Also, the most frequent documents in BD should be considered by the site's administrator as good suggestions of documents that should be reachable from the top levels in the Web site (this is also true in minor extent for AD documents). That is, we suggest hotlinks based on queries and not on navigation, as is usual. It is important to consider that the same query can co-occur in class A and class B (what cannot co-occur is the same document in AD and BD!), so the relevance associated to each type of query is proportional to its frequency in each one of the classes in relation to the frequency of the document in AD or BD.

Unsuccessful Queries.

If a query submitted to the internal search engine did not have visited results in the session that generated it, we will consider it as an unsuccessful query. There are two main causes for this behavior:

1. The search engine displayed zero documents in the results page, because there were no appropriate documents for the query in the Web site.

2. The search engine displayed one or more results, but none of them seemed appropriate from the user's point of view. This can happen when there is poor content or with queries that have polysemic words.

There are four types of unsuccessful queries, which we will call C, C', D and E. We define formally these classes of queries as follows (see Figure 5.2):

Class C queries: Queries for which the internal search engine displayed results, but the user choose not no visit them, probably because there were no appropriate documents for the user's needs at that moment. This can happen for queries that have ambiguous meanings and for which the site has documents that reflect the words used in the query, but not the concept that the user was looking for. It can also happen when the contents of the site do not have the specificity that the user is looking for. Class C queries represent concepts that should be developed in depth in the contents of the Web site with the meaning that users intended, focused on the keywords of the query.

Class C' queries: Queries for which the internal search engine did not display results. This type of query requires a manual classification by the webmaster of the site. If this manual classification establishes that the concept represented by the query *exists* in the Web site, but described with different words, then this is a class C' query. These queries represent words that should be used in the text that describes links and documents that share the same meaning as these queries.

Class D queries: As in class C' queries, the internal search engine did not display results and manual classification is required. However, if in this case, the manual classification establishes that the concept represented by the query does *not exist* in the Web site, but we believe that it should appear in the Web site, then the query is classified as class D. Class D queries represent concepts that should be included in documents in the Web site, because they represent new topics that are of interest to users of the Web site.

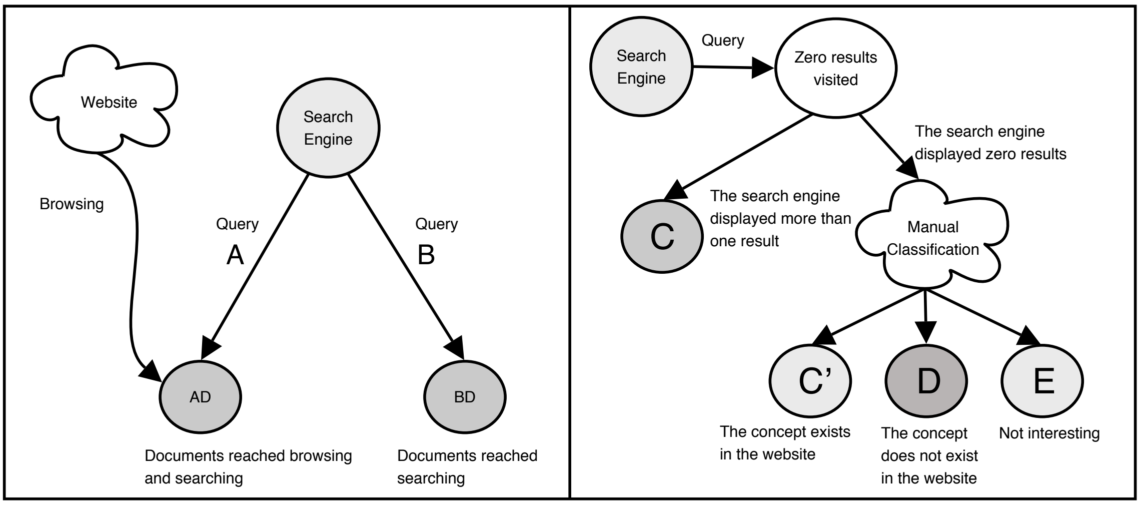

Website
Browsing
Search Engine
Query
A
Query
B
AD
BD
Documents reached browsing and searching
Documents reached searching
Search Engine
Query
Zero results visited
The search engine displayed zero results
C
The search engine displayed more than one result
Manual Classification
C'
D
E
The concept exists in the website
The concept does not exist in the website
Not interesting

Figure 5.2: Successful queries *(right)* and unsuccessful queries *(left)*.

Class	Concept exists	Results displayed	Visited documents	Significance	Contribution	Affected component
A	yes	yes	DQ ∩ DWS	low	additional IS	anchor text
B	yes	yes	DQ \ DWS	high	new IS, add hotlinks	anchor text, links
C	yes	yes	∅	medium	new content	documents
C'	yes	no	—	medium	new IS	anchor text, documents
D	no, but it should	no	—	high	new content	anchor text, documents
E	no	no	—	none	—	—

Table 5.1: Classes of queries and their contribution to the improvement of a Web site.

Class E queries: Queries that are not interesting for the Web site, as there are no results, but it's not a class C' or class D query, and should be omitted in the classification[1].

Each query class is useful in a different way for improving the Web site's content and structure. The importance of each query will be considered proportional to that query's frequency in the usage logs, and each type of query is only counted once for every session. Table 5.1 shows a review of the different classes of queries.

Manual classification is assisted by a special interface in our prototype implementation. The classification is with memory (that is, an already classified query does not need to be classified in a subsequent usage of the tool) and we can also use a simple thesaurus that relates main keywords with its synonymous. In fact, with time, the tool helps to build an ad-hoc thesaurus for each Web site.

[1] this includes typographical errors in queries, which could be used for a hub page with the right spelling and the most appropriate link to each word.

5.2.3. Supplementary Task 1: Content and Structure Mining

A Web site *"is not simply a collection of pages, it is a network of related pages"* [89] and as an interconnected network it can be viewed and studied as a graph. The information provided by the graph organization, along with the contents of the site and the usage data collected by the server, can be used to significantly optimize and enhance a Web site, thus improving the quality of that site.

Most of the existing Web site mining models are focused on analyzing the usage information of the Web site, but they generally do not relate this information with the contents and structure of the site. To fill this void we include in our model a preliminary method to validate the Web site organization and interconnectivity. This method suggests improvements to the Web site's contents and structure, with the intention of making the contents and link structure more coherent and straightforward for users in general. This is achieved by: *(1) suggesting the addition of links between similar documents*, *(2) revising links between unrelated documents*, *(3) pointing out the most visited sets of documents so they can be linked from the top levels of the Web site*, and *(4) establishing relevant topics in the Web site, based on the most visited clusters of documents, so these topics can be improved in the site*.

Our method analyzes the contents of a Web site, based on the text found in each document and the structure of the site, reflected by the links between pages. This information then is processed to validate if the content distribution in the Web site agrees with its link structure and the usage data registered in the Web site's access logs. Using this information several reports are generated by the model's prototype, which help the site's administrator visualize possible "problem areas" in the Web site and ways to solve them.

Text Clustering and Link Analysis

The model clusters the documents in the Web site according to their similarity (the number of clusters is determined experimentally for each site). In the clustering phase each document in the site is represented internally using the vector space model, and the similarity is measured using the cosine function between vectors, scaled using the *tf-idf* scheme. The clustering methodology used in this case is a first approximation to this problem, since for future versions we will substitute the bag-of-words approach for the query-sets document model (presented in Chapter 3).

The clustering process is achieved using sequential bisections, optimizing in each iteration the global clustering function: $\max(\sum_{i=1}^{k} \sqrt{\sum_{v,u \in S_i} sim(u,v)})$. This function was experimentally found appropriate for the process, and is discussed in further in [60]. Sequential bisections, generate a hierarchical tree, called a dendogram. The root node of this tree contains all of the documents in the Web site. Every time a bisection is performed the tree branches into smaller sets until it has as many leafs as desired clusters. The resulting structure is analyzed by our model to validate if documents that are similar, such as documents in the same cluster and documents from clusters that descend from the same parent node, have links between them. We sustain the theory that a user interested in one document is very likely to be interested in other similar documents. In our model we propose that these documents should be connected by links to improve the structure of the Web site by helping users reach more easily the contents they are looking for.

Correlation of Content and Queries

The clustering results are then compared with the information from the usage logs of the Web site. These logs indicate which clusters were the most visited by users. We propose that the most visited clusters represent "topics" that interest users, and these documents should be linked from the top levels of the Web site. The usage logs also provide information on which documents were reached by visitors from global search engines and/or from the site's internal search engine. From this we can establish a ratio of documents in a cluster visited from a search engine, compared to the number of documents visited by navigation. This model shows possible problems, such as clusters that are very visited from search engines but not by navigation, which indicates that the links pointing to these clusters are not visible from the top pages of the site or that they do not describe correctly the contents of these documents. In this way the model helps the site's administrator to view these issues and find possible solutions improving the Web site.

5.2.4. Supplementary Task 2: Frequent Query Patterns

All of the user queries are analyzed to discover frequent item sets (or frequent query patterns). Every keyword in a query is considered as an item. The discovered patterns contribute general information about the most frequent word sets used in queries. The patterns are then compared to the number of results given in each case by the internal search engine, to indicate if they are answered in the Web site or not. If the most frequent patterns don't have answers in the Web site, then it is necessary to review these topics to improve these contents more in depth.

5.3. Evaluation

To test our model we used our prototype on several Web sites that had an internal search engine, the details of the prototype can be found in [78]. We will present some results from two of those sites: the first one, the Web site of a company dedicated to providing domain name registrations and services, and the second one, a portal targeted at university students and future applicants.

First Use Case. In Table 5.2 we present some results from the different query classes obtained for the first use case. This site does not have a large amount of documents (approximately 1,130 documents) and its content, rather technical, seems quite straightforward. We believe this was the reason for finding only class A, B, C, D and E queries, but no class C' queries in its reports.

In Table 5.2 we have several suggestions for additional IS obtained from class A queries. Class B queries shown in this sample are very interesting, since they indicate which terms provide new IS for anchor text of documents about "nslookup", "CIDR", "trademarks"

Class A	Class B	Class C	Class D
domains	nslookup	hosting	ASN
Internet providers	CIDR	DNS	
syntax	trademarks	server	
electronic invoice	lottery	prices	
diagnosis tools	Web domain	web hosting	

Table 5.2: Sample of class A, B, C and D queries for the first use case.

and “Web domains”, which were topics not found by browsing in the site. Another interesting query in class B is “lottery”, which shows a current popular topic within the registered domains in the site and makes a good suggestion for a hotlink in the top pages of the Web site. On the other hand, class C queries show that documents related mainly to topics on “Web hosting services” should be developed more in depth in the Web site. The only class D query found for this site, was “ASN”, which stands for a unique number assigned by the InterNIC that identifies an autonomous system in the Internet. This is a new topic that was not present in the contents of the site at the moment of our study.

Second Use Case. The second use case, the portal targeted at university students and future applicants, was the primary site used for our evaluation in this chapter. This site has approximately $8,000$ documents, $310,000$ sessions, $130,000$ external and $14,000$ internal queries per month. Additionally there were $194,525$ links interconnecting the documents in this site. Using our model reports were generated for four months, two months apart from each other. The first two reports were used to evaluate the Web site without any changes, and show very similar results amongst each other. For the following reports, improvements suggested from the evaluation were incorporated to the site’s content and structure. In this approach, the 20 most significant suggestions from the particular areas of: “university admission test” and “new student application”, were used. This was done to target an important area in the site and measure the impact of the model’s suggestions. A sample of frequent query patterns found in the Web site is shown in Table 5.3 and a sample of class A, B, C, C’ and D queries is presented in Table 5.4.

Additionally, we present the visualization using colors (see Figure 5.3) which shows the dendogram built during the text clustering phase of this use case. This includes the number of links interconnecting documents inside of the different nodes. Each color represents different levels of interconnectivity amongst documents. The prototype also shows a detailed report with all the usage information regarding each cluster, such as: how many user sessions visited the cluster by navigation, how many user sessions visited the cluster using the internal search engine, or an external search engine, and the most visited clusters in the Web site. The reports also show the number of connected components inside the cluster, which give a measure of inter-cluster link connectivity.

The improvements to the Web site were made mainly to the top pages of the site, and included adding IS to link descriptions, adding new relevant links, suggestions extracted from frequent query patterns, class A and B queries. Other improvements consisted of broadening the contents on certain topics using class C queries, and adding new contents to the site using class D queries. For example the site was improved to include more

Percent(%)	Frequent pattern
3.55	admission test results
2.33	admission test scores
1.26	application results
1.14	scholarships
1.10	tuition fees
1.05	private universities
0.86	institutes
0.84	law school
0.80	career
0.74	courses
0.64	admission score
0.61	student loan
0.58	admission score
0.57	nursing
0.55	practice test *(only 2 results)*
0.54	engineering
0.53	psychology
0.53	credit
0.51	registration
0.51	grades
0.51	admission results *(only 2 results)*
0.49	architecture
0.44	student bus pass *(only one answer)*

Table 5.3: Sample of frequent query patterns for the second use case (indicating which ones had few answers).

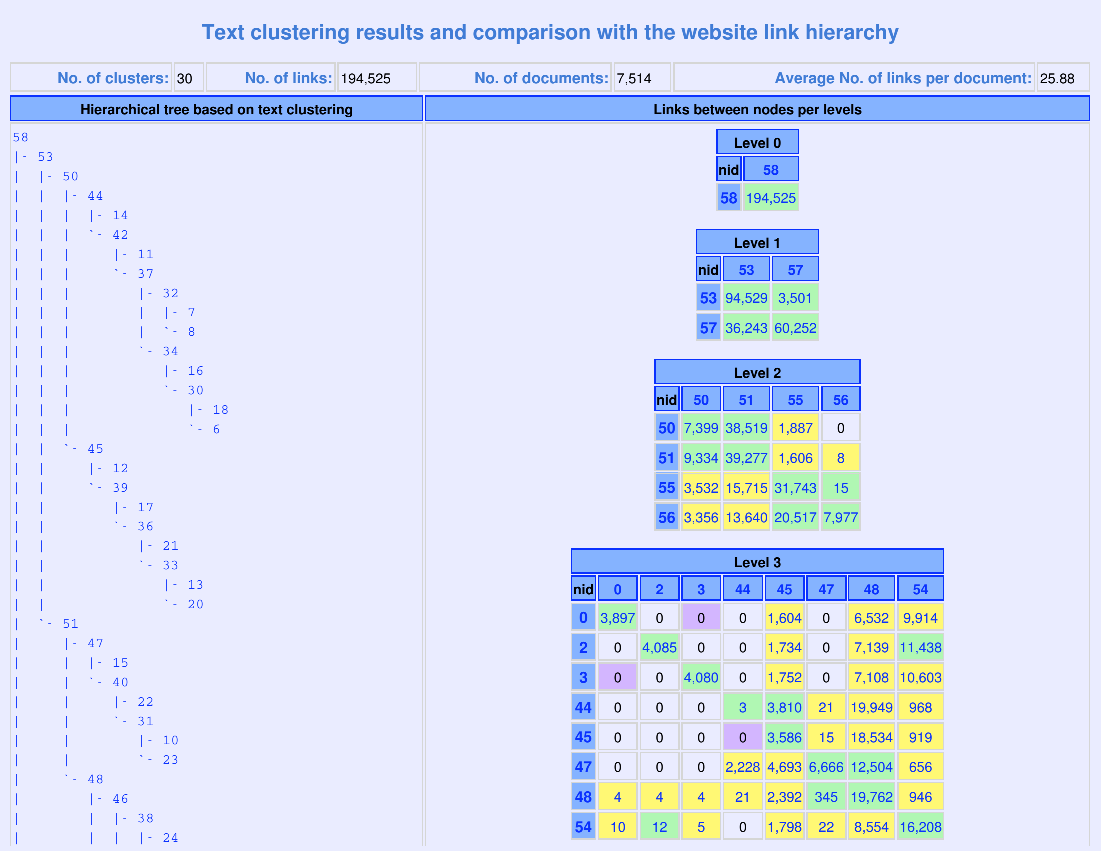
admission test examples, admission test scores and more detailed information on scholarships, because these where issues constantly showing in class C and D queries. To illustrate our results we will show a comparison between the second and third report. Figures 5.4, 5.5 and 5.6 show the changes in the Web site after applying the suggestions. For Figure 5.6 the queries studied are only the ones that were used for improvements.

In Figure 5.4 we present the variation in the general statistics of the site. After the improvements were made, an important increase in the amount traffic from external search engines is observed (more than 30% in two months), which contributes to an increase in the average number of page views per session per day, and also in the number of sessions per day. The increase in visits from external search engines is due to the improvements in the contents and link descriptions in the Web site, validated by the keywords used on external queries. After the improvements were made to the site, we can appreciate a slight decrease in the number of internal queries and clicked documents from those queries. This agrees with our theory that contents are being found more easily in the Web site and that now less documents are accessible only through the internal search engine. All of these improvements continue to show in the next months of analysis.

Class A	Class B
practice test	university scholarships
thesis	admission test
admission test preparation	admission test inscription
university ranking	curriculum vitae
private universities	presentation letter
employment	bookstores

Class C	Class C’	Class D
admission test	government scholarships	Spain scholarships
admission test results	diploma	waiting lists
practice test	evening school	vocational test
scholarships	mobility scholarship	compute test score
careers	humanities studies	salary

Table 5.4: Sample of class A, B, C, C’ and D queries for the second use case.



Text clustering results and comparison with the website link hierarchy

No. of clusters:	30	No. of links:	194,525	No. of documents:	7,514	Average No. of links per document:	25.88

Hierarchical tree based on text clustering

```
58
|- 53
|   |- 50
|   |   |- 44
|   |   |   |- 14
|   |   |   `- 42
|   |   |       |- 11
|   |   |       `- 37
|   |   |           |- 32
|   |   |           |   |- 7
|   |   |           |   `- 8
|   |   |           `- 34
|   |   |               |- 16
|   |   |               `- 30
|   |   |                   |- 18
|   |   |                   `- 6
|   |   `- 45
|   |       |- 12
|   |       `- 39
|   |           |- 17
|   |           `- 36
|   |               |- 21
|   |               `- 33
|   |                   |- 13
|   |                   `- 20
|   `- 51
|       |- 47
|       |   |- 15
|       |   `- 40
|       |       |- 22
|       |       `- 31
|       |           |- 10
|       |           `- 23
|       `- 48
|           |- 46
|           |   |- 38
|           |   |   |- 24
```

Links between nodes per levels

Level 0

nid	58
58	194,525

Level 1

nid	53	57
53	94,529	3,501
57	36,243	60,252

Level 2

nid	50	51	55	56
50	7,399	38,519	1,887	0
51	9,334	39,277	1,606	8
55	3,532	15,715	31,743	15
56	3,356	13,640	20,517	7,977

Level 3

nid	0	2	3	44	45	47	48	54
0	3,897	0	0	0	1,604	0	6,532	9,914
2	0	4,085	0	0	1,734	0	7,139	11,438
3	0	0	4,080	0	1,752	0	7,108	10,603
44	0	0	0	3	3,810	21	19,949	968
45	0	0	0	0	3,586	15	18,534	919
47	0	0	0	2,228	4,693	6,666	12,504	656
48	4	4	4	21	2,392	345	19,762	946
54	10	12	5	0	1,798	22	8,554	16,208

Figure 5.3: Hierarchical tree and interconnectivity.

Figure 5.5 shows the comparison between the number of documents (results) clicked from each query class, this number is relative to the numbers of queries in each class. External and internal AD documents present an important increase, showing that more external queries are reaching documents in the Web site, and that those documents now belong

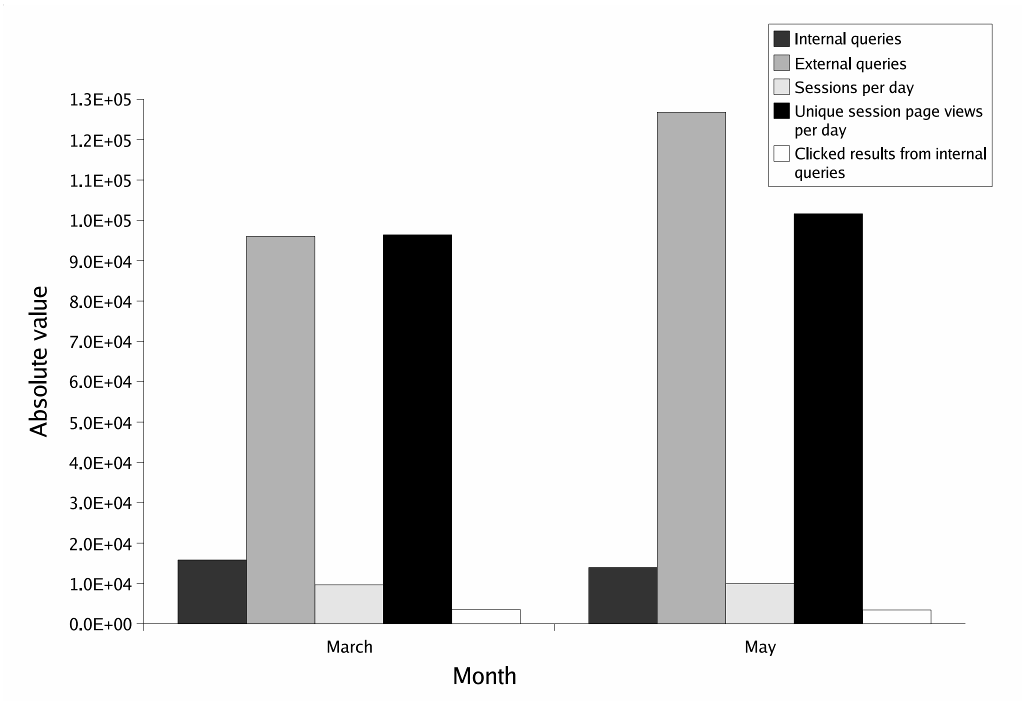


Figure 5.4: General results.

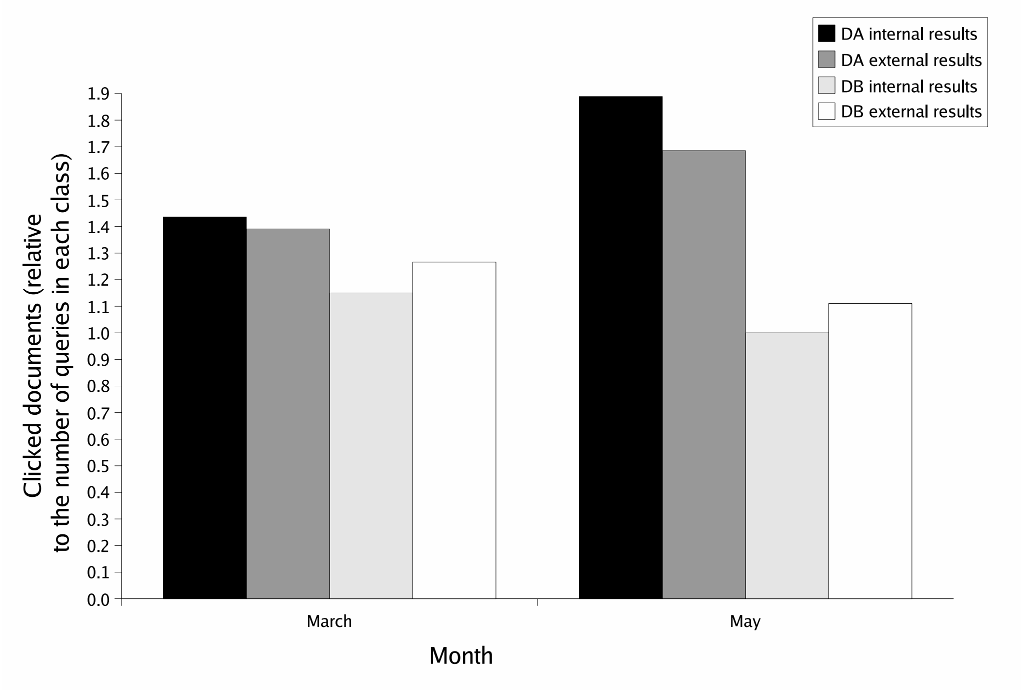


Figure 5.5: Clicked results.

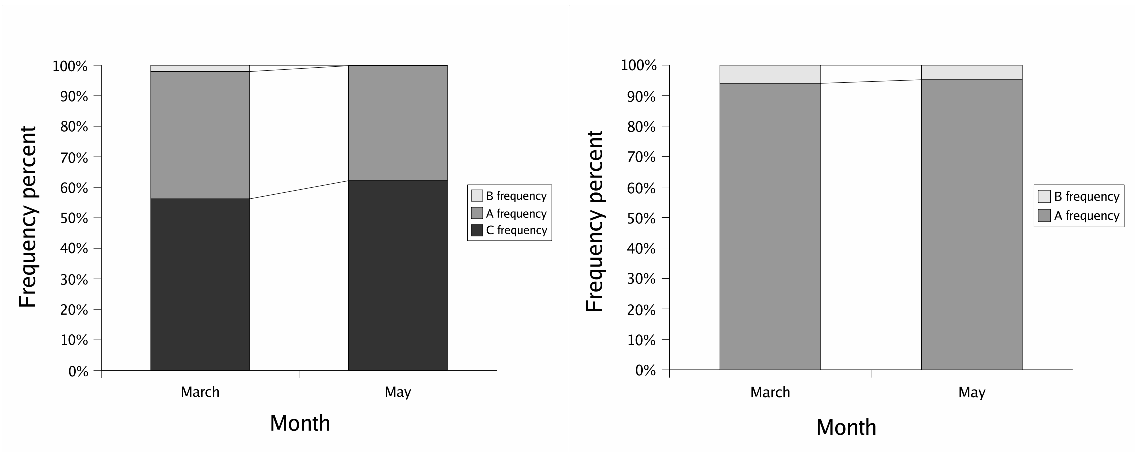


Figure 5.6: Internal *(left)* and external *(right)* query frequency.

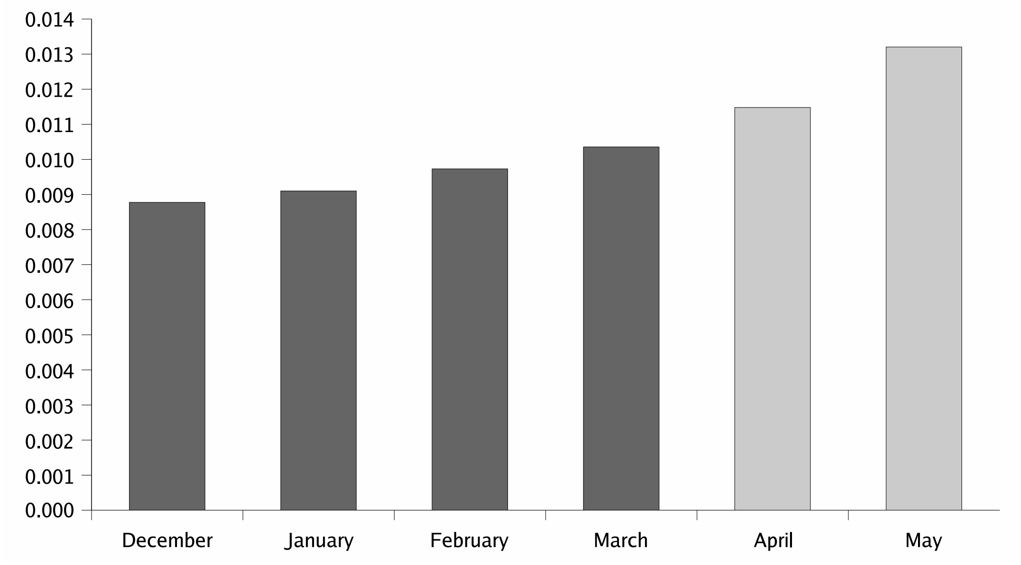


Figure 5.7: Daily average number of external queries per month (normalized by the number of sessions).

to documents that are being increasingly reached by browsing also. On the other hand BD documents continue to decrease in every report, validating the hypothesis that the suggested improvements cause less documents to be only reached by searching. In Figure 5.6 the distribution of A, B and C queries can be appreciated for internal and external queries. Internal queries show a decrease in the proportion of A and B queries, and an increase in queries class C. For external queries, class A queries have increased and class B queries have decreased, as external queries have become more directed at AD documents.

Figure 5.7 and Figure 5.8 show statistics related to the amount of external queries in the Web site in months previous to the application of the model's suggestions and for the two months during and after they were applied (April and May). Usage data for the month of February was incomplete in Figure 5.7 (due to circumstances external to the authors) and had to be generated using linear interpolation with the months unaffected by our study. The data presented in Figures 5.7 and 5.8 show a clear increase above average in the

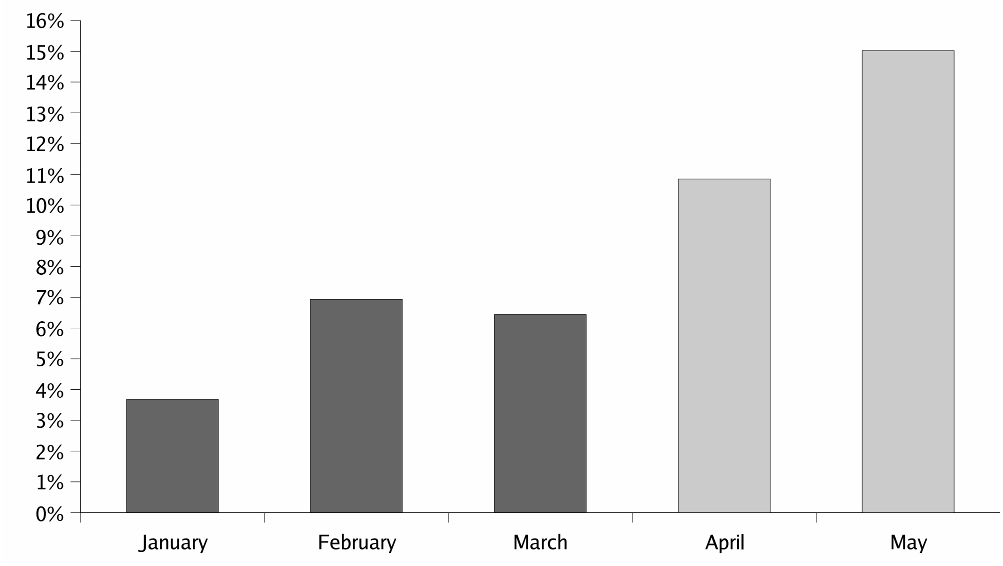


Figure 5.8: Month to month percent variation of the daily average number of external queries (normalized by the number of sessions).

volume of external queries that reached the Web site during April and May, specially in the month of May when the increase was in 15% compared to April, which is coherent with the fact that the results from the prototype where applied at the end of March.

5.4. Chapter Conclusions and Future Work

In this chapter we presented the first Web site mining model that is focused on query classification. The aim of this model is to find better IS, contents and link structure for a Web site. Our tool discovers, in a very simple and straight forward way, interesting information. For example, class D queries may represent relevant missing topics, products or services in a Web site. Even if the classification phase can be a drawback at the beginning, in our experience, on the long run it is almost insignificant, as new frequent queries rarely appear. The analysis performed by our model is done offline, and does not interfere with Web site personalization. The negative impact is very low, as it does not make drastic changes to the Web site. Another advantage is that our model can be applied to almost any type of Web site, without significant previous requirements, and it can still generate suggestions if there is no internal search engine in the Web site.

The evaluation of our model shows that the variation in the usage of the Web site, after the incorporation of a sample of suggestions, is consistent with the theory we have just presented. Even though these suggestions are a small sample, they have made a significant increase in the traffic of the Web site, which has become permanent in the next few reports. The most relevant results that are concluded from the evaluation are: *an important increase in traffic generated from external search engines*, *a decrease in internal queries*, *also more documents are reached by browsing and by external queries*. Therefore the site has become more findable in the Web and the targeted contents can be reached more easily by users.

Future work involves the development and application of different query ranking algorithms, improving the visualizations of the clustering analysis and extending our model to include the origin of internal queries (from which page the query was issued). Also, adding information from the classification and/or a thesaurus, as well as the anchor text of links, to improve the text clustering phase. Our work could also be improved in the future by analyzing query chains as discussed in [82] with the objective of using these sequences to classify unsuccessful queries, specifically class C' and E queries. Furthermore, we would like to change the clustering algorithm to automatically establish the appropriate number of clusters and do a deeper analysis of most visited clusters. The text clustering phase could possibly be extended to include stemming. Another feature our model will include is an incremental quantification of the evolution of a Web site and the different query classes. Finally, more evaluation is needed specially in the text clustering area.

Chapter 6

A Unified Hyperlink-Click Graph

6.1. Introduction

In this chapter, we take a different approach than in previous chapters and we look into the graph interpretation of queries and their clicked documents. In recent years, significant amount of research has been devoted to studying the *Web graph* (which we refer to as *hyperlink graph* to avoid ambiguity) and the *click graph*. The hyperlink graph is the directed graph among Web pages in which edges represent hyperlinks. The click graph is a view of the information contained in query logs, i.e., a bipartite graph between queries and Web pages, in which edges connect a query with the documents that were clicked by users as a result.

At an intuitive level, these two graphs capture two of the most common tasks of users on the Web: *browsing* and *searching*. A user who browses the Web essentially follows edges on the hyperlink graph, while a user who searches and consequently clicks on the result pages, is following edges on the click graph. Searching and browsing together are equivalent to the two prototypical actions of information seeking and exploration.

The edges of these two graphs can capture certain semantic relations between the objects they represent. An example of such a relation is *similarity*: two pages connected together by a hyperlink, or a query and a page connected together by a click, are more likely to be similar than two non-connected objects [43]. Another presumed semantic relation is *authority endorsement*: a hyperlink from a page u to a page v, or a click from a query q to the page v, can both be viewed as implicit "votes" for page v [64]. These hypotheses provide a foundation for the research of several Web information retrieval problems, for instance clustering of Web pages, queries and users, on-line community discovery. Similarity search exploits the similarity hypotheses, while ranking leverages the authority-endorsement theories.

Unfortunately both the hyperlink graph and the click graph have certain disadvantages. For example, Google's PageRank [32] uses links in the hyperlink graph to compute importance scores for Web pages. As a result substantial adversarial effort has been put into artificially increasing the PageRank score of Web pages. This adversarial effort takes the form of *spam* pages or *link farms* [53, 50].

Similarly, the click graph has its own disadvantages. One of those disadvantages is its sparsity: a page that is clicked for a certain query must first appear in the list of results for that query. This may not be trivial considering the vast number of pages available for each query. Also, there is an issue of an inherent bias in any rankings produced by this graph, favoring already highly ranked Web pages. Another related problem is its large dependency on textual matching: typically search engines emphasize precision at the expense of recall, and display only results which match exactly all the query terms, causing many relevant pages not to be connected with queries if they are not exact matches. Furthermore, the click graph is also prone to spam, but in this case *click spam* which aims towards taking advantage of usage mining algorithms to improve search ranking.

Contributions of this work

In this chapter we propose a new type hybrid Web graph, which combines the existing hyperlink and the click graphs, and we apply Web mining and link-analysis algorithms to it. This new graph, which we call the *hyperlink-click graph*, is a simple graph union: it has two types of nodes, *pages* and *queries*, with directed edges between pages according to the hyperlink graph, and undirected edges between queries and pages according to the click graph.

The union of these two graphs combines the traditional hyperlink graph, based on connectivity structure, and the click graph, based on search engine usage information. The purpose of this graph is to extend the traditional hyperlink graph into a graph which reflects more accurately users' natural behavior in the Web.

In particular we define and study random walks on the unified graph. We show that ranking according to the scores obtained from the hyperlink-click graph is similar to ranking using the score of the non-combined graph with the highest performance. The unified graph compensates where either the hyperlink or click graph execute poorly, being overall more robust and fail-safe. It is important to note that in modern Web search engines, link analysis scores in the style of PageRank might be only small components of the overall ranking function. Nevertheless, we compare directly to those scores in order to isolate the effect of the hyperlink-click graph.

Combining usage and content information in one structure can improve the quality of many Web mining algorithms. From our point of view, the two graph structures are complementary and each of them can be used to alleviate the shortcomings of the other. For example, using clicks to include user feedback on the Web graph improves its resistance against link-spam. On the other hand, by considering hyperlinks and browsing patterns we increase the density and connectivity of the click graph, and we can account for pages that users might visit *after* issuing particular queries.

Applications of the hyperlink-click graph

There are several Web mining tasks in which the hyperlink-click graph can be used:

- **Ranking of documents.** A random walk on the hyperlink-click graph can be used to obtain importance scores for documents, which can be used to enhance document ranking. This particular application is in the focus of this chapter.
- **Query ranking and query recommendation.** As a by-product of the random walk on the hyperlink-click graph, importance scores are obtained not only for documents but also for queries. Such query scores can be used for query recommendation: given a query, we can use the graph to find other similar queries, and then use the importance scores to rank those queries and provide alternative query recommendations to the user.
- **Similarity search.** There have been many notions of distances among documents and among queries, which have been based on the topology of the hyperlink graph (e.g. SimRank [56]) and the click graph (e.g. [41]). Such distance functions provide building boxes for designing meaningful similarity-search algorithms. We believe that refining such graph-based distance measures for the hyperlink-click graph can lead to better notions of similarity, since the hyperlink-click graph provides richer information about the objects that it relates. These similarity metrics can be used to find communities on the Web.
- **Spam detection.** Link-based features extracted from the hyperlink graph, can be used to improve content-based spam detection algorithms [24]. It is reasonable to hypothesize that link features extracted from the hyperlink-click graph can be useful to further improve spam detection.

We plan to investigate some of these applications in future work. The main focus of this chapter is the first application: enhancing the ranking of Web documents.

The rest of the chapter is organized as follows. In Section 6.2 we present the related work. In Section 6.3 we introduce our notation and provide a formal description of the graphs used in this chapter. Section 6.4 discusses the random walk model, which is mainly used for ranking. In Section 6.5 we discuss our experimental results, and finally, in Section 6.6 summarizes our results and conclusions.

6.2. Related Work

There are several models for representing the information on the Web. The most popular view is the one based on structure. This approach sees the Web as a graph in which documents are nodes that are connected to each other when there is at least one hyperlink from one document to the other. This graph structure has been exploited by link-based ranking algorithms such as [32] and [61]. Both methods rank pages according to their *importance* and *authority*, estimated by analyzing the endorsements or links from other documents.

In the work presented in [16] there is an overview of many other possible graph-based representations based on the content and usage data found on the Web. The focus is on the analysis of queries from search engines and their semantic relations, as well as their relations given by the clicks on common documents. Relations between queries can be inferred from common keywords or common clicked documents. In a similar way, relations between documents can be found by looking at shared links or words. The incorporation of document contents into these types of graphs is introduced from the words in queries, their selected documents, and also by the relations induced among documents with similar words.

With respect to usage data, a common model for query logs from search engines is in the form of a bipartite undirected graph. This graph includes two types of nodes: queries and documents. Links between the two types of nodes are generated by user clicks from queries to documents in the process of selecting a search result. This type of representation was presented in [25] and used for agglomerative clustering to find related queries and documents. Later, this view was expanded in [41] where weights were added to the undirected edges, based on the number of clicks from the query to a document. This graph is referred to as *click graph*. They study the effect of forward and backward random walks on this model for document ranking. They discuss that queries should be considered as *soft* relevance judgments, and that query logs give noisy and sparse data. The work of [41] suggest that an effective method is a backward random walk.

On the other hand, the notion of *unification* of different Web data sources is not a new one. In [101] a framework is proposed for link analysis. This framework allows to model inter-type and intra-type links between different Web objects. They discuss that any link-based model can be studied within their framework and they focus their work on users and their browsing behavior. In particular they apply this to extend the HITS algorithm by incorporating users browsing patterns.

Noise and malicious manipulation of Web content affect both the click graph and hyperlink graphs. The most typical type of manipulation is link spam on the hyperlink graph [53, 50]. In this approach artificial links are created to induce higher link-based ranks on documents. In a similar way, click graph manipulation can be produced from artificial clicks on search engine results [81, 50]. The aim of this attack is to manipulate learned ranking functions that are based on click through information. Another type of *noise* that can be found in click through data is the bias of clicks due to the position of the search result. This bias has been studied and modeled, e.g. by [46, 42].

Another perspective on query logs is to avoid considering queries individually, but use them as sequences of actions. This is explored in [82] and serves a dual purpose: it reduces the noise due to single queries, and it allows the connection of different actions of users over time.

6.3. Web Graphs

In this section we describe three types of Web graphs: the hyperlink graph, the click graph, and the hyperlink-click graph. We introduce the notation that is used in the chapter, and describe the random walks that are performed over the graphs.

The hyperlink graph: Given a set of N Web documents D we consider the *hyperlink graph* $G_{\mathrm{H}} = (D, H)$ as a directed graph, where there is an edge $(u, v) \in H$ if and only if document u has a hyperlink to document v, for $u, v \in D$.

For a document $u \in D$, the set of *in-neighbors* of u (the documents that point to u) and the set of *out-neighbors* of u (the documents that are pointed to by u) are denoted by $N_{\mathrm{IN}}(u)$ and $N_{\mathrm{OUT}}(u)$, respectively. That is, $N_{\mathrm{IN}}(u) = \{v \in D \mid (v, u) \in H\}$ and $N_{\mathrm{OUT}}(u) = \{v \in D \mid (u, v) \in H\}$. For $u \in D$, $d_{\mathrm{IN}}(u) = |N_{\mathrm{IN}}(u)|$ is the *in-degree* of document u, and $d_{\mathrm{OUT}}(u) = |N_{\mathrm{OUT}}(u)|$ is its *out-degree*.

The click graph: Let $Q = \{q_1, \ldots, q_M\}$ be the set of M unique queries submitted to a search engine during a specific period of time. In practice, in order to construct the set of unique queries we assume some simple normalization, such as normalizing for space, letter case, and ordering of the query terms. For a query $q \in Q$ we denote by $f(q)$ the *frequency* of the query q, that is, how many times the query was submitted to the search engine.

In a large-scale search engine query log, in addition to the information about which queries have been submitted, there is information about which documents are clicked by the users who submit those queries. Let $D = \{d_1, \ldots, d_N\}$ be the set of N Web documents clicked for those queries.

The click graph $G_{\mathrm{C}} = (Q \cup D, C)$ is an undirected bipartite graph that involves the set of queries Q, the set of documents D, and a set of edges C. For $q \in Q$ and $d \in D$, the pair (q, d) is an edge of C if and only if there is a user who clicked on document d after submitting the query q. The obvious prerequisite is that the document d is in the set of results computed by the search engine for the query q. To each edge $(q, d) \in C$ we associate a numeric weight $c(q, d)$ that measures the number of times the document d was clicked when shown in response to the query q.

As before, we define $N(q) = \{a \mid (q, a) \in C\}$ the set of neighboring documents of a query $q \in Q$, and $N(a) = \{q \mid (q, a) \in C\}$ the set of neighboring queries of a document $a \in D$. We then define the weighted degree of a query $q \in Q$ as $d(q) = \sum_{a \in N(q)} c(q, a)$, and similarly, the weighted degree of a document $a \in D$ as $d(a) = \sum_{q \in N(a)} c(q, a)$.

The hyperlink-click graph: Quite simply, the hyperlink-click graph G_{HC} can be seen as the *union* of the hyperlink graph and the click graph. There is a directed edge of weight 1 between documents u and v if there is a hyperlink from u to v, and there is an undirected weighted edge between query q and document d if there are clicks from q to d, and the weight of the edge is equal to the number of clicks $c(q, d)$.

6.4. Random Walks on Web Graphs

Given a graph $G = (V, E)$ a *random walk* on G is a process that starts at a node $v_0 \in V$ and proceeds in discrete steps by selecting randomly a node of the neighbor set of the node at the current step. A random walk on a graph of N nodes can be fully described by an $N \times N$ matrix $\mathbf{P}$ of *transition probabilities*. The i-th row and the i-th column of $\mathbf{P}$ correspond both to the i-th node of the graph, $i = 1, \ldots, N$. The $\mathbf{P}_{ij}$ entry of $\mathbf{P}$ is the probability that the next node will be the node j given that the current node is the node i. Thus, all rows of $\mathbf{P}$ sum to 1, and $\mathbf{P}$ is called *row-stochastic* matrix.

Under certain conditions (irreducibility, finiteness, and aperiodicity, see [70] for definitions and more details) a random walk is characterized by a steady-state behavior, which is known as the *stationary distribution* of the random walk. Formally, the stationary distribution is described by an N-dimensional vector $\boldsymbol{\pi}$ that satisfies the equation $\boldsymbol{\pi}\mathbf{P} = \boldsymbol{\pi}$. Alternatively, the i-th coordinate $\boldsymbol{\pi}_i$ of the stationary-distribution vector π measures the frequency in which the i-th node of the graph is visited during the random walk, and thus, it has been used as an intuitive measure of the *importance* of each node in the graph.

Next we will consider random walks in the three different graphs we have introduced: the hyperlink graph, the click graph, and the hyperlink-click graph. We will denote the stationary distributions in those three graphs by $\boldsymbol{\pi}_{\mathrm{H}}$, $\boldsymbol{\pi}_{\mathrm{C}}$, and $\boldsymbol{\pi}_{\mathrm{HC}}$, respectively. We will refer to the values of the stationary distribution vectors as *scores*.

Random walk on the hyperlink graph. The random walk on the hyperlink graph corresponds to surfing the Web by following hyperlinks at random from the current Web page. The concept has been popularized through the seminal paper by Brin and Page [32], and its application to the Google search engine. The stationary distribution is also known as the *PageRank* vector. In the PageRank model, a step of following a random hyperlink is performed with probability α, while the walk “jumps” (“teleports” or “resets”) to a random page with probability $1 - \alpha$. Additionally, special care is taken when reaching a *dangling node*, a node with no outgoing edges. A common assumption is that upon reaching to a dangling node the random walk continues by selecting a target node uniformly at random. Consequently, if $\mathbf{A}_{\mathrm{H}}$ is the adjacency matrix of the Web graph G_{H}, define $\mathbf{N}_{\mathrm{H}}$ to be the normalized version $\mathbf{A}_{\mathrm{H}}$ so that all rows sum to 1. Assume that $\mathbf{N}_{\mathrm{H}}$ is defined to take care of the dangling nodes, so that if a row of $\mathbf{A}_{\mathrm{H}}$ has all 0s, then the corresponding row of $\mathbf{N}_{\mathrm{H}}$ has all values equal to $1/N$. Finally, let $\mathbf{1}_{\mathrm{H}}$ be a matrix that has the value $1/N$ in all of its entries. Then the transition-probability matrix $\mathbf{P}_{\mathrm{H}}$ of the random walk on the Web graph is given by $\mathbf{P}_{\mathrm{H}} = \alpha\mathbf{N}_{\mathrm{H}} + (1 - \alpha)\mathbf{1}_{\mathrm{H}}$.

In addition to yielding a better model of surfing the Web graph, performing the random jumps with probability $(1 - \alpha) \neq 0$ ensures the sufficient conditions for the stationary distribution to be defined.

Random walk on the click graph. Random walk on the click graph is similar, except for the fact that the click graph is bipartite and undirected. Being bipartite creates periodicity in the random walk, while being undirected has the consequence that the stationary distribution is proportional to the degree of each node. However, assuming that we also perform random jumps with probability $(1 - \alpha)$, then the random walk is aperiodic and irreducible (every node can be reached from every other node), and also the stationary distribution at each node is not a direct function of its degree.

Formally, the random walk on the click graph is described as follows. Let $\mathbf{A}_{\mathrm{C}}$ be an $M \times N$ matrix, whose M rows correspond to the queries of Q and the N columns correspond to the documents of D, and whose (q, d) entry has value $c(q, d)$, the number of clicks between query $q \in Q$ and document $d \in D$. Let $\mathbf{A}'_{\mathrm{C}}$ be an $(M+N) \times (M+N)$ matrix defined by

$$\mathbf{A}'_{\mathrm{C}} = \begin{pmatrix} \mathbf{A}_{\mathrm{C}} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{\mathrm{C}}^{T} \end{pmatrix},$$

and let $\mathbf{N}_{\mathrm{C}}$ be the row-stochastic version of $\mathbf{A}'_{\mathrm{C}}$. Here again we assume that $\mathbf{N}_{\mathrm{C}}$ is defined to take care of the dangling nodes, so that if a row of $\mathbf{A}_{\mathrm{C}}$ has all 0s, then the corresponding row of $\mathbf{N}_{\mathrm{C}}$ has all values equal to $1/(M+N)$. Finally, let $\mathbf{1}_{\mathrm{C}}$ be an $(M+N) \times (M+N)$ matrix that has value $1/(M+N)$ in all its entries. Then the transition-probability matrix that describes the random walk on the click graph is $\mathbf{P}_{\mathrm{C}} = \alpha \mathbf{N}_{\mathrm{C}} + (1-\alpha)\mathbf{1}_{\mathrm{C}}$.

Note that in [41] a backward random walk is used, while we consider instead a forward random walk.

Random walk on the hyperlink-click graph. Using the notation that we introduced in the previous paragraphs, the random walk on the hyperlink-click graph is defined as follows: First overwrite $\mathbf{A}_{\mathrm{H}}$ to be an $(M+N) \times (M+N)$ matrix, including also the M queries and assuming that all rows that correspond to queries are 0s. Then let $\mathbf{N}_{\mathrm{H}}$ be the row-stochastic version of $\mathbf{A}_{\mathrm{H}}$, normalizing for dangling nodes—note that all newly introduced queries correspond to dangling nodes—while let $\mathbf{N}_{\mathrm{C}}$ be as before. Finally, let $\mathbf{1} = \mathbf{1}_{\mathrm{C}}$.

For combining the graphs introduce a *querying probability* β, which determines the rate at which a user switches between querying a surfing behavior. The transition-probability matrix for the random walk on the hyperlink-click graph is then given by

$$\mathbf{P}_{\mathrm{HC}} = \alpha\beta \mathbf{N}_{\mathrm{C}} + \alpha(1-\beta)\mathbf{N}_{\mathrm{H}} + (1-\alpha)\mathbf{1}. \tag{6.1}$$

Let us also describe at a high-level the random walk defined by the above equation. First, with probability $(1-\alpha)$ the walk goes to a random query or to a random document. With probability α, the walk follows a link in the hyperlink-click graph. The exact action depends on whether the current state is a document or a query. If the current state is a document u, then with probability β the next state is a query q for which there are clicks to u, while with probability $1-\beta$ the next state is a document v pointed by u. If the current state is a query, then with probability β the next state is document for which there are clicks from the query, while with probability $1-\beta$ the next state is any random document.

For our experiments, while we investigate the effects of the value of the parameter β to the results, we fix the value of α to be 0.85, since it is a value widely used for PageRank computation.

6.5. Experimental Evaluation

In this section we present the experiments performed in order to validate the utility of the scores produced by random walks on the hyperlink-click graph. We compare these scores to those generated by the hyperlink and click graphs independently. The objective of this section is to discover new information for improving the ranking of Web documents.

For the comparison of the different random walk scores, we focus on two *tasks* in which a good ranking method should perform well. These tasks are: *ranking high-quality documents* and *ranking pairs of documents*. The evaluation is centered on analyzing the dissimilarities among the different models.

We begin by describing the datasets used.

6.5.1. Dataset

As a data source we use an in-house query log. Due to the enormous size of the Web, we use only a small sample of documents and queries. Thus, we use only partial graphs instead of the full graphs. No publicly available Web document collections are used, because there are no collections with query log information associated to them, which is a fundamental constraint for our experiments.

We create the graph data by using the query log as the starting point. First let us denote by D_{QL} the set of all documents contained the query log. We parse the query log and we find all the documents that have 10 or more clicks. There are about $9\,000$ such documents in our sample, and we refer to them as *seed* documents D_{S}.

We then use a Web crawl to find all documents that point to and are pointed to by the seed documents. Let $D_{\mathrm{IN}} = \bigcup_{u \in D_{\mathrm{S}}} N_{\mathrm{IN}}(u)$ and $D_{\mathrm{OUT}} = \bigcup_{u \in D_{\mathrm{S}}} N_{\mathrm{OUT}}(u)$ be the sets of documents with outlinks to and inlinks from D_{S}, respectively, and let $D_{\mathrm{ALL}} = D_{\mathrm{S}} \cup D_{\mathrm{IN}} \cup D_{\mathrm{OUT}}$ be the set of all documents encountered. The above expansion process increases the number of total documents (documents in D_{ALL}) to approximately 144 million.

It should be noted that documents gathered through this expansion process might also exist in D_{QL}. We then define D_{C} to be the documents in the intersection of D_{ALL} and D_{QL}, that is $D_{\mathrm{C}} = D_{\mathrm{ALL}} \cap D_{\mathrm{QL}}$.

Finally, the set of queries Q_{C} that we consider are the queries that have at least one click in the set of documents D_{C}. In total, there are about $61\,000$ such queries. The dataset construction described above is shown in Figure 6.1. Given the above sets, we then define the three graphs we consider, the hyperlink graph, the click graph, and the hyperlink-click graph as follows:

Hyperlink graph: The nodes of the hyperlink graph G_{H} are all the documents in the set D_{ALL}. The edges are all the induced hyperlinks between this set of documents. We also note here that, due to the popularity of the documents in the seed set, the set D_{IN} is considerably larger than the set D_{OUT}.

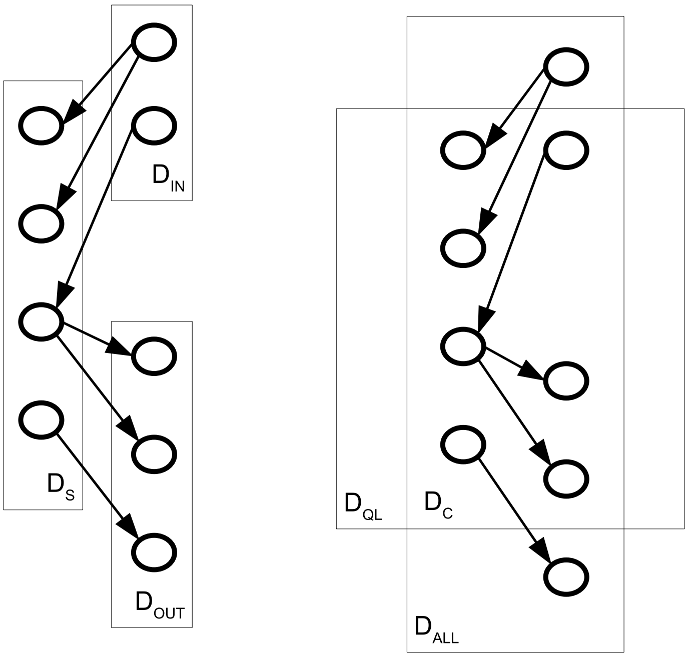


Figure 6.1: Construction of our dataset.

Click graph: The nodes of the click graph G_C are the documents in D_C and the queries in Q_C. The edges are induced by the clicks in the query log, and the number of clicks serve also as weights for the edges.

Hyperlink-click graph: The hyperlink-click graph G_HC is the union of the hyperlink graph G_H and and click graph G_C. Thus the document set for the hyperlink-click is again the set D_ALL. The weights on the edges of G_HC depend on the querying probability β, as in Equation 6.1. We use 5 values of the parameter β ($\beta = \{0.25, 0.50, 0.75, 0.85, 0.95\}$), and we denote the resulting graph by $G_\mathrm{HC}(\beta)$.

The selected dataset reflects a consistent sample of the Web graph, although highly popular documents are chosen as a seed set, this is further expanded to include most of the neighboring documents. This expansion allows to include in the dataset an heterogeneous sample of documents which are connected to the initial set. Additionally, the query log data is processed very quickly using the MG4J[1] and fastutil[2] tools available on-line. This computational cost is almost negligible compared to that of processing the hyperlink graph.

6.5.2. Random-walk Evaluation

As described in Section 6.5.1, our experimental datasets are partial and they only represent a sample of the whole Web. Hence, to make the obtained results comparable, we analyze only the results for the documents contained in the intersection of the click, hyperlink and

[1] http://mg4j.dsi.unimi.it

[2] http://fastutil.dsi.unimi.it

combined graphs (which we refer to as D_{C}). However, it is important to note that we use all of the nodes in each graph to compute the random walk results, and not only the ones contained in D_{C}.

We compute $\boldsymbol{\pi}_{\mathrm{H}}$, $\boldsymbol{\pi}_{\mathrm{C}}$ and $\boldsymbol{\pi}_{\mathrm{HC}}$ for the values of $\beta = \{0.25, 0.50, 0.75, 0.85, 0.95\}$. It is important to take into account that even for very large values of β, random walks on G_{HC} are quite different from those on G_{C}. This is due to the high influence of G_{H} on the combined graph and is observed in throughout the evaluation.

Task: ranking high-quality documents

To compare the random walk results, we decided to focus on high-quality Web documents and how they score within the different models. The hypothesis we sustain is that it is desirable for a good model to score high-quality documents above other documents. To measure this, we use documents from the DMOZ document directory.[3] Our working hypothesis is that since DMOZ is editorially maintained, on average, documents in this directory are of higher quality than documents not in the directory. Consequently, we use D_{Z} to denote the set of documents in the evaluation set D_{C} that belong also to the DMOZ directory. Following our working hypothesis, we postulate that the graph that produces the best ranking results is the graph that ranks documents in D_{Z} higher than the rest of the documents in D_{C}.

To quantitatively measure the agreement of the rankings produced from the different graphs with the DMOZ directory, we use two measures:

Π_{Z}: Our first measure is the normalized sum of the $\boldsymbol{\pi}$ scores of D_{Z} documents. This is,

$$\Pi_{\mathrm{Z}} = (\sum_{d \in D_{\mathrm{Z}}} \boldsymbol{\pi}(d))/(\sum_{d \in D_{\mathrm{C}}} \boldsymbol{\pi}(d)).$$

The intuition of this measure is that we want a large amount of probability mass of the stationary distribution of the random walk to be accumulated with documents in D_{Z}. Thus the value of the measure should be as high as possible.

Γ_{Z}: The second measure we use is inspired by the *Goodman-Kruskal Gamma* measure[62], which is a descriptive rank-order correlation statistic, often used in psychology. Given two rankings on a set of items, on which the two rankings disagree on D pairs of items and agree in A pairs, the Γ measure between the rankings is defined to be $\Gamma = (D - A)/(D + A)$. In our case, even though membership in the DMOZ category does not induce a complete ranking, we can still consider a weak ranking in which all documents in DMOZ are ranked before all documents that are not in DMOZ, and the definition of Γ can still be applied: we just do not include pairs of documents that are either both in DMOZ or none in DMOZ. The measure Γ takes values between -1 and 1, where -1 means that the two rankings are completely discordant, while 1 means that the two rankings are concordant. Again the value of the measure should be as high as possible.

[3] http://dmoz.org

Algorithm 1 Micro-evaluation

```
1. Define a set of queries $Q \subset Q_{\mathrm{C}} \in G_{\mathrm{C}}$ that have at least 1 edge to a document in $D_{\mathrm{Z}}$ and 1 edge to a document in $D_{\mathrm{C}} - D_{\mathrm{Z}}$.
    a) for each $q \in Q$ find all the adjacent documents $D_q$ that belong to $D_{\mathrm{C}}$.
    b) compute $\Pi_{\mathrm{Z}}$ and $\Gamma_{\mathrm{Z}}$ replacing $D_{\mathrm{C}}$ with the documents in $D_q$ and $D_{\mathrm{Z}}$ with $D_q \cap D_{\mathrm{Z}}$.
2. Compute the average values of $\Pi_{\mathrm{Z}}$ and $\Gamma_{\mathrm{Z}}$.
```

Table 6.1: Macro-evaluation results

	Π_{Z}	Γ_{Z}
G_{C}	0.275	**0.643**
G_{H}	**0.600**	0.458
$G_{\mathrm{HC}}(0.95)$	**0.597**	**0.558**
$G_{\mathrm{HC}}(0.85)$	0.591	0.552
$G_{\mathrm{HC}}(0.75)$	0.587	0.551
$G_{\mathrm{HC}}(0.50)$	0.580	0.544
$G_{\mathrm{HC}}(0.25)$	0.574	0.540

Table 6.2: Micro-evaluation results

	Π_{Z}	Γ_{Z}
G_{C}	**0.738**	**0.604**
G_{H}	0.664	0.273
$G_{\mathrm{HC}}(0.95)$	**0.752**	**0.563**
$G_{\mathrm{HC}}(0.85)$	0.749	0.546
$G_{\mathrm{HC}}(0.75)$	0.745	0.534
$G_{\mathrm{HC}}(0.50)$	0.738	0.501
$G_{\mathrm{HC}}(0.25)$	0.730	0.483

We evaluate the proposed measures Π_{Z} and Γ_{Z} in two levels of granularity, which are are defined as follows:

Macro-evaluation: This evaluation intends to capture the overall scores of high-quality documents for the complete D_{C} document set. The quality measures Π_{Z} and Γ_{Z} are computed considering all the documents in D_{C} and D_{Z}.

Micro-evaluation: This evaluation is performed at *query level*. This means that to compute Π_{Z} and Γ_{Z} the sets D_{C} and D_{Z} are reduced to only those documents clicked from a particular query. This is repeated for each query in Q_{C} that has at least one document in DMOZ and at least one document that is not in DMOZ. In the end the results of are averaged over the total number of queries processed. Formally the procedure for the micro-evaluation is defined in Algorithm 1.

The results obtained in the macro and micro evaluations are shown in Table 6.1 and Table 6.2, respectively. The macro-evaluation results show that for the Π_{Z} the best value is

Table 6.3: Top 10 documents for 3 of the random walk scores

G_{H}	G_{C}	$G_{\mathrm{HC}}(85)$
www.mp3lyrics.org	www.yahoo.com	www.gmail.com
www.gratka.pl	cams.com	www.quizilla.com
www.pimpmyspacepages.com	uk.yahoo.com	www.gratka.pl
www.dpreview.com	www.google.com	www.ebay.com.my
www.mtv.com/...	www.theaa.com/...	www.veoh.com
www.ebay.com.my	www.ebay.co.uk	www.livejournal.com
www.veoh.com	www.nationalrail.co.uk	www.google.pl
www.xe.com	www.cineworld.co.uk	spaces.live.com
www.livevideo.com	games.yahoo.com	www.flixster.com
www.music.com	www.streetmap.co.uk	mail.yahoo.co.uk

obtained for G_{H} and the worst for G_{C}. On the other hand, in the Γ_{Z} the roles are reversed with G_{C} being the overall graph with less inverted elements and G_{H} the one with the most number of inverted elements. The results of the G_{HC} follow closely the best performing scores with less than 0.003 difference for Π_{Z} and 0.085 difference for Γ_{Z}.

The micro-evaluation results, in Table 6.2, shows that for the Π_{Z} metric, G_{HC} obtains the best value followed by G_{HC}. For the Γ_{Z} metric G_{C} is the best, and G_{H} is the worst in both Π_{Z} and Γ_{Z}.

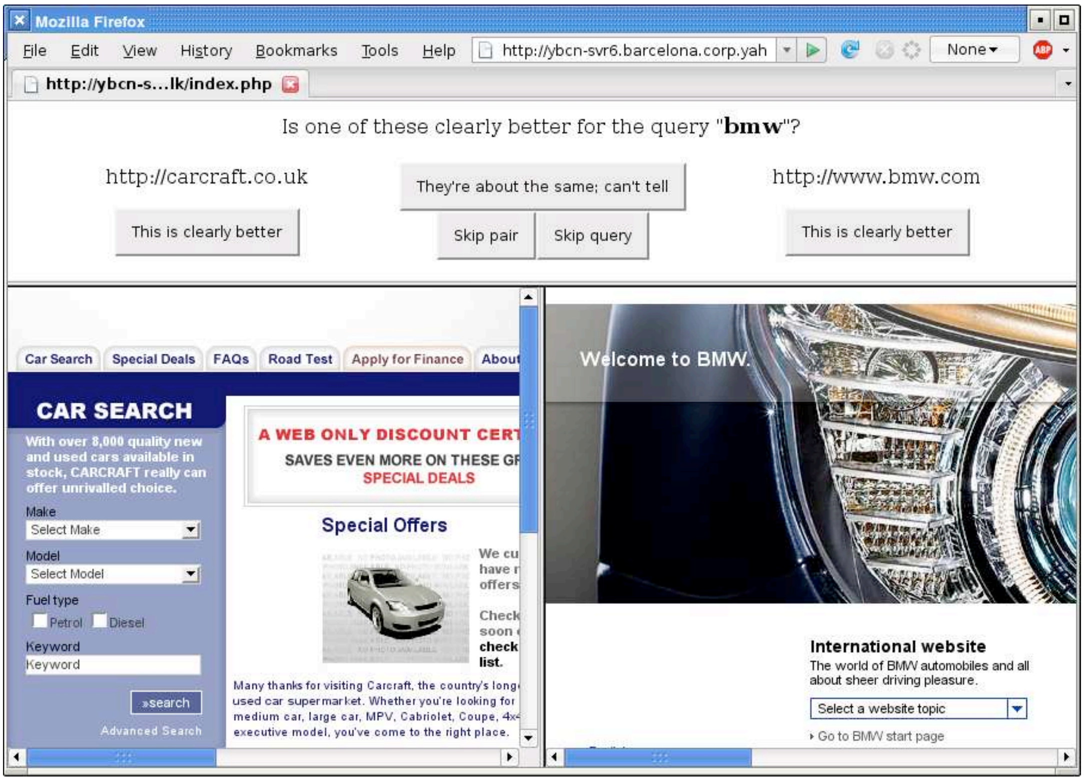
These metrics observe the performance of the random walk scores using different perspectives. From our point of view a good scoring method should perform well both at macro and micro level. The results obtained show that the random walk scores on the G_{HC} follow closely the best scores generated by the non-combined graphs.

Task: ranking pairs of documents

In addition to evaluating our rankings using the measures Π_{Z} and Γ_{Z}, which are based on the assumption that documents in DMOZ are on average of high quality, we also perform a user study.

We evaluated a set of triples of the form (q, d_1, d_2) where q is a query with at least 10 clicks in total, d_1 and d_2 are two distinct documents returned by the search engine for that query. Also, we limited the evaluation to cases in which the ordering of d_1 and d_2 was different according to at least two scoring methods in $\{\ \boldsymbol{\pi}_{\mathrm{H}}, \boldsymbol{\pi}_{\mathrm{C}}, \boldsymbol{\pi}_{\mathrm{HC}}\ \}$. The evaluation interface is shown in Figure 6.2. Users where presented a randomly selected triple and asked: "*Is one of these pages clearly better for the query q?*". They were also given the option to say that the two documents were about the same, or that they could not be compared.

A group of 13 human assessors participated in the evaluation. A total of $1,710$ assessments were collected, from which 515 (32%) expressed preference for one of the two documents. There were 82 cases in which more than one evaluator assessed the same triple and expressed a preference for one of the two documents. In those cases, the agreement among the evaluators was 70%. Still, the assessment process proved to be very



Is one of these clearly better for the query "**bmw**"?

http://carcraft.co.uk

http://www.bmw.com

Mozilla Firefox
File
Edit
View
History
Bookmarks
Tools
Help
http://ybcn-svr6.barcelona.corp.yah
None
http://ybcn-s...lk/index.php
This is clearly better
They're about the same; can't tell
Skip pair
Skip query
This is clearly better
Car Search
Special Deals
FAQs
Road Test
Apply for Finance
About
CAR SEARCH
With over 8,000 quality new and used cars available in stock, CARCRAFT really can offer unrivalled choice.
Make
Select Make
Model
Select Model
Fuel type
Petrol
Diesel
Keyword
Keyword
»search
Advanced Search
A WEB ONLY DISCOUNT CERT
SAVES EVEN MORE ON THESE GR
SPECIAL DEALS
Special Offers
We cu
have r
offers
Check
soon
check
list.
Many thanks for visiting Carcraft, the country's long
used car supermarket. Whether you're looking for
medium car, large car, MPV, Cabriolet, Coupe, 4x
executive model, you've come to the right place.
Welcome to BMW.
International website
The world of BMW automobiles and all about sheer driving pleasure.
Select a website topic
Go to BMW start page

Figure 6.2: Classification interface.

difficult since many of the selected pairs of documents have only a marginal difference in their scores.

The results of the user study are shown in Table 6.4, using again the Γ statistic to measure the agreement between the rankings of the algorithms and the rankings induced by the human evaluators.

Table 6.4: Γ of ranking functions with human preferences

Method	Overall	Average per query
π_{C}	**0.197**	**0.195**
π_{HC}	0.063	0.042
π_{H}	-0.122	-0.141

Table 6.5: Γ of ranking functions with human preferences for $\delta \geq 4.5 \cdot 10^{-7}$ (38% of unique pairs)

Method	Overall	Average per query
π_{HC}	**0.156**	**0.132**
π_{C}	0.111	0.124
π_{H}	-0.078	-0.082

Due to the marginal difference in scores between many pairs of documents, we study the behavior of Γ for the pairs of documents which have a *greater* difference between their scores. For this we evaluate only the pairs of documents (d_i, d_j) for which all scoring methods have a minimum $\delta = |\boldsymbol{\pi}(d_i) - \boldsymbol{\pi}(d_j)|$. This allows to evaluate pairs that are less ambiguous to assess for humans. As a result we found that for values of $\delta \geq 4.5 \cdot 10^{-7}$ the Γ values of $\boldsymbol{\pi}_{\mathrm{C}}$ and $\boldsymbol{\pi}_{\mathrm{HC}}$ are reversed and that $\boldsymbol{\pi}_{\mathrm{HC}}$ produces the best performance at this point (shown in Table 6.5).

If we continue to increase the minimum value of δ we obtain the results shown in Figure 6.3.

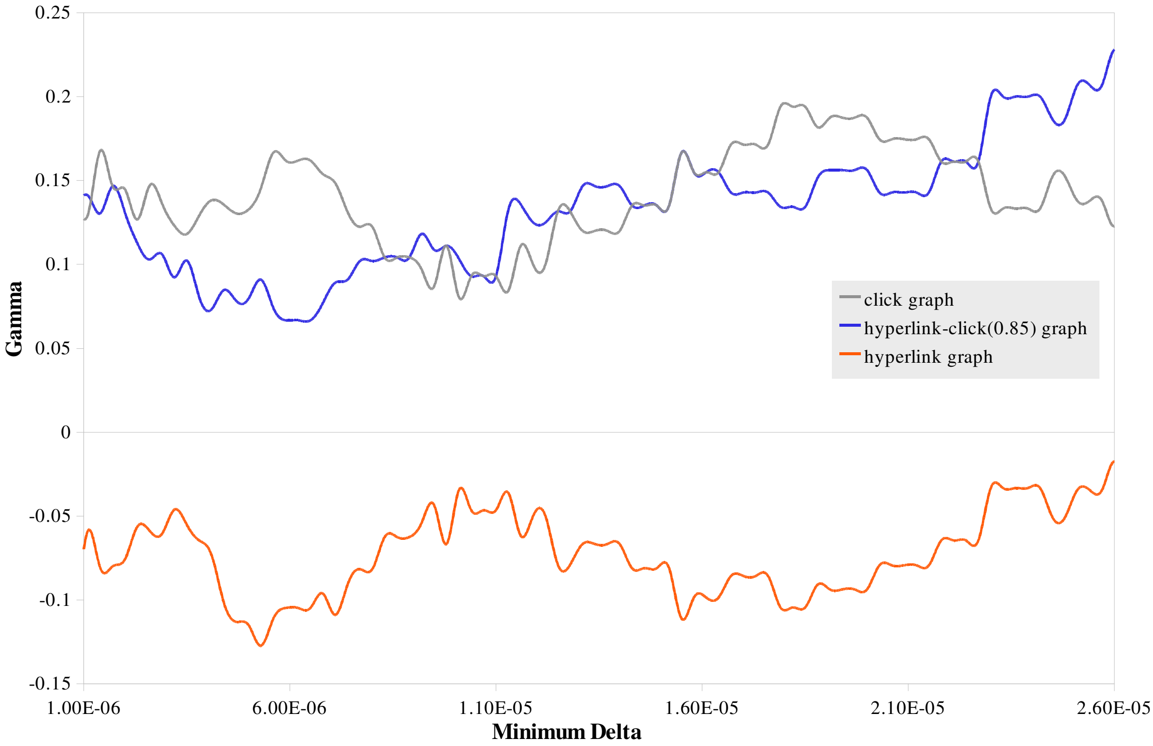


Figure 6.3: Behavior of Γ in the user study when restricting the minimum allowed value of δ.

Introducing “variations” into the click-graph

The click graph, just as the hyperlink graph can be prone to induced variations, which can affect the scores of the random walk. For the hyperlink graph it is well know that typical variations are produced by link-spam. In the case of the click graph, undesirable modifications in the random walk scores can be the consequence of different methods that increase the number of clicks, such as click-spam. Other variations on the click-through data can occur from sponsored placement of search engine results, in general these do not represent a practical problem, since in general they can be filtered from a query log. Nevertheless we will study the effects of induced variations by using clicks on sponsored results to simulate click-spam.

In the previous part of the evaluation sponsored clicks were filtered from the G_{C} and G_{HC}. We repeat this evaluation introducing sponsored click-through data into G_{C} and G_{HC}. Tables 6.6 and 6.7 show the results of the high-quality document evaluation with this variation. We can observe that in the macro-evaluation the order prevails with respect

Table 6.6: Macro-evaluation results

	$\Pi_{\rm Z}$	$\Gamma_{\rm Z}$
$G_{\rm C}$	0.2151	**0.7912**
$G_{\rm H}$	**0.5851**	0.2103
$G_{\rm HC}(0.95)$	0.5584	0.6429

Table 6.7: Micro-evaluation results

	$\Pi_{\rm Z}$	$\Gamma_{\rm Z}$
$G_{\rm HC}(0.95)$	**0.5772**	**0.2361**
$G_{\rm H}$	0.5713	-0.1495
$G_{\rm C}$	0.5356	0.1677

to the original results. On the other hand, in the micro-evaluation $G_{\rm HC}$ performs better for both metrics.

The user study was repeated with 5 judges, which did $1,576$ assessments in total, from which 588 (37%) expressed a preference for one of the two documents. Unlike the results of the user study without sponsored clicks, in this case users agreed more with $\boldsymbol{\pi}_{\rm H}$ and less $\boldsymbol{\pi}_{\rm C}$, i.e.: results were reversed. Nevertheless, the results for $\boldsymbol{\pi}_{\rm HC}$ remained in the middle (see Table 6.8).

Table 6.8: Γ of ranking functions with human preferences using click-through data with sponsored clicks

Method	Overall	Average per query
$\boldsymbol{\pi}_{\rm H}$	**0.098**	**0.088**
$\boldsymbol{\pi}_{\rm HC}$	-0.091	-0.137
$\boldsymbol{\pi}_{\rm C}$	-0.244	-0.186

Summary of the experimental evaluation

In Tables 6.9 and 6.10 we provide a concise summary of the metrics and types of evaluations used to measure the quality of the different random walk scores. The convention that we use is that $G_A > G_B$ means that the ranking generated using the graph G_A is better than the ranking generated using the graph G_B (according to our measures), while $G_A \approx G_B$ means that the difference between the two rankings is less than 0.1.

In Figures 6.4 and 6.5 we show a comparison of metrics $\Gamma_{\rm Z}$ and $\Pi_{\rm Z}$ with click variations and without variations. In this Figures we can observe that the values for G_{HC} are always very close or better than the best result from the non-combined graphs. This result is independent on whether or not click variations where induced into the data.

Overall the different tasks evaluated reflect consistency in the results. The values obtained for the study performed with DMOZ documents are coherent for the variations in the value of β, and furthermore, they agree with the results obtained from the user evaluation. We consider this as an indicator of the usefulness of the evaluation and its metrics.

Table 6.9: Summary of the evaluation for the task of finding high-quality documents

metric	macro		micro	
	without click variations	with click variation	without click variations	with click variations
Γ_{Z}	$G_C \approx G_{HC} > G_H$	$G_C > G_{HC} > G_H$	$G_C \approx G_{HC} > G_H$	$G_{HC} \approx G_C > G_H$
Π_{Z}	$G_H \approx G_{HC} > G_C$	$G_H \approx G_{HC} > G_C$	$G_{HC} \approx G_C > G_H$	$G_{HC} \approx G_H \approx G_C$

Table 6.10: Summary of the evaluation for the task of ranking pairs of documents

metric	without click variation	with click variation
Γ	$G_C > G_{HC} > G_H$	$G_H > G_{HC} > G_C$
$\Gamma(\delta \geq 4.5 \cdot 10^{-7})$	$G_{HC} \approx G_C > G_H$	–

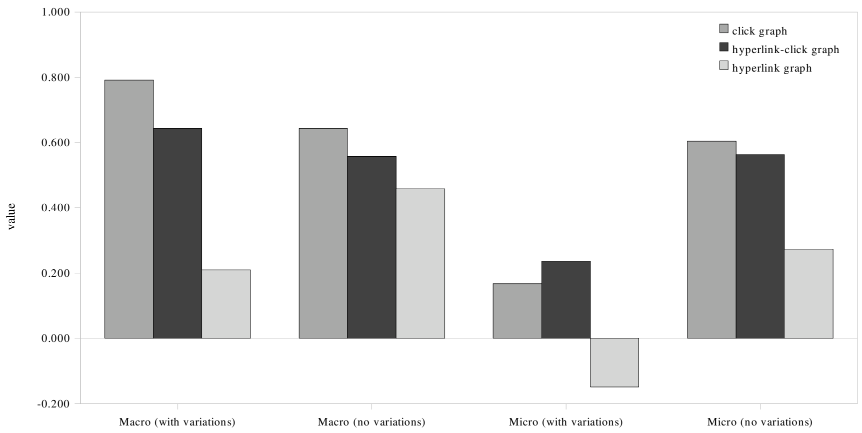


Figure 6.4: Summary of Γ_{Z} values with and without click variations for high-quality document evaluation

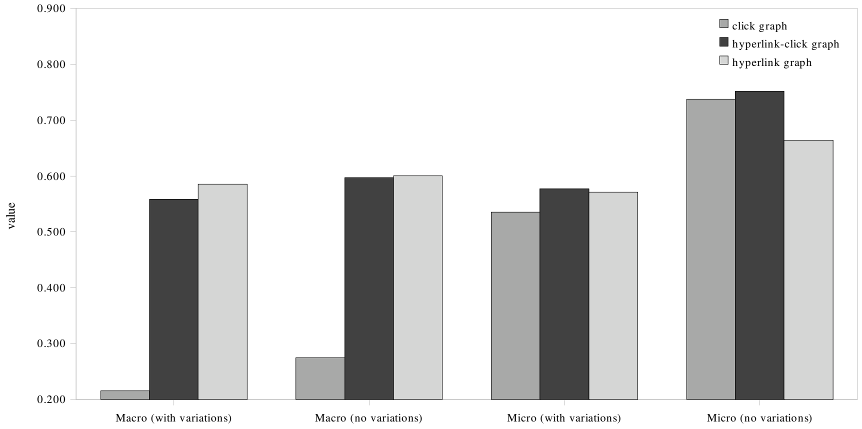


Figure 6.5: Summary of Π_{Z} values with and without click variations for high-quality document evaluation

6.6. Chapter Conclusions and Future Work

In this chapter we studied the effects of a random walk on a unified Web graph. This Web graph combines both hyperlinks between documents and clicks from queries to documents, and was created to capture more completely users' searching and browsing behavior in the Web.

Our main motivation for studying this unified graph is to analyze the new information that it can provide. As a first approach, we focus on the task of using this model to enhance Web document ranking. For this we used a number of different evaluation metrics in order to assess the ranking produced on tour graph with respect to rankings produced by the hyperlink and click graphs. We evaluated by analyzing useful tasks for ranking, such as, ranking high-quality documents and also ranking pairs of documents. For the later, we conducted a user study which provided consistent results with the rest of the evaluation. On the other hand, we also tested the tolerance of our model to click variations or *noisy* data.

Our experimental evaluation shows that the scores generated by random walks on the combined Web graph have several useful properties for document ranking. Overall these scores produce good quality results which are very stable and tolerant to noisy click-through data. Additionally, our results show that the unified graph is always close to the best performance of either the click or hyperlink graph. Furthermore the results on the combined graph never approximate the lower bound according to any metric, while the non-combined graphs do not generate good results in all cases.

It is our belief that these properties of the unified graph are useful for improving current ranking techniques. Partly as an indicator of how reliable link-based ranks and click-based ranks are for different tasks. As well as an independent indicator of document quality.

As part of future work we would like to analyze how to deal with the inherent bias that exists in any ranking technique based on usage mining. This is, that pages with already high stationary distribution scores are presented to users more often as a query result. Thus, high ranked pages tend to be more clicked. In long term use, this could create a self-reinforcing ranking. This is not an easy problem to solve and it depends mainly on other underlying ranking techniques. Therefore we recommend any click-based ranking as as a complement to other independent ranking methods.

Also in the future we would like to analyze other Web mining applications for the hyperlink-click graph, such as: link and click spam detection and similarity search.

The code used for the weighted random walk described in this chapter is available at `http://law.dsi.unimi.it/satellite-software/`

Chapter 7

Business Privacy Protection in Query Log Mining

7.1. Introduction

As discussed in previous chapters of this thesis, Web query logs contain rich information about users' activities, personal preferences and interests. But at the same time, query log analysis can also reveal information that users consider private and therefore inappropriate for publication. Due to this, a large amount of research has been directed towards protecting *user* information through log anonymization and privacy-preserving data mining. In this chapter, we look into the privacy related implications of query log analysis and in particular, we analyze a new concern, the privacy of *businesses*. This includes both institutions (such as companies) and people in the public eye (such as political leaders). Our definitions, observations and findings are focused on company privacy protection, but extend to people who stand in the public (Internet) light.

Privacy preservation for businesses seems like a contradiction in itself. What one institution considers private others might decide to publish without any concerns. This is analogous to the case of Web users. Some users choose to publish their personal information, such as pictures, videos and phone number, while others prefer to keep this information private, and under no circumstances want it to become public. In the same way as users' private information should not be disclosed, business confidential data is also worth protecting. In this study, we give a definition of the type of information that we consider *confidential* and we analyze attacks that lead to confidentiality breaches, including methods to prevent information disclosure.

A key issue in our research is to *prevent the disclosure of previously unknown data about a particular institution* through the publication of an anonymized query log. The same holds for any processed data published by query log mining applications. Therefore, the scope of our work is confidentiality protection for query log publishing and query log mining. This applies to services offered by third-party search engine log owners, including the information released by keyword suggestion tools for search engine ad placement, or online Web site usage reports. We believe that entities that provide content for these type of services should be aware of the *real amount of data* that they are disclosing. This is also an important issue for query log owners, such as search engine providers, who depend on Web sites to deliver high quality content. For example, many search engines play a *neutral* role in the assessment of Web sites and would be setback if important sites decide to forbid access to their crawlers due to privacy concerns.

To illustrate the need to protect business confidential information, we provide the following made-up, but realistic example.

Example 1 (Disclosing confidential company data). We consider two car selling companies X, Y that use their site as a promotional channel. Company X's product, $car(X)$, is a direct competitor of company Y's product $car(Y)$: The two cars have comparable functionalities, prices and are designed for the same target group. Now assume that Y wants to know the impact of X's Web site on the sales of $car(X)$ and assume that X considers this information confidential; this supposition is valid since X has chosen *not* to make this information public.

Company Y applies Web mining on their own site $site(Y)$ and obtains a fairly complete characterization of the user groups accessing it. However, they cannot extrapolate this information on X's Web site, $site(X)$, because they do not know anything about the traffic on that site. Moreover, $site(X)$ is designed differently and many of its contents are irrelevant to Y.

Web analysis also returns statistics about accesses from search engines. Y has learned that $p\%$ of the pageviews on $car(Y)$ lead to a purchase, and $s\%$ of these pageviews come from search engine S. Y can thus compute the contribution of S to the sales of $car(Y)$. This is valuable for the relationship of Y to S. But it still cannot be used to compute the number of sales of $car(X)$.

Finally, Y knows the queries or keywords used in S to access $site(Y)$. This set, $queries(Y)$, reflects the *position* of $car(Y)$ in the Web, according to the perception of users. Company Y can use these queries to enhance their marketing campaign and their Web site. By running each of these queries, they can also find whether $car(X)$ is ranked higher or lower than $car(Y)$ in S. But they cannot make conclusions about how many users have accessed $car(X)$ via S – mainly because $site(X)$ contains words that are different from those used in $site(Y)$.

Now, assume that S decides to publish its query log periodically, using a similar anonymization as AOL, discussed in [13]. Thanks to this, company Y can acquire new information about $car(X)$: They simply have to extract all of the entries in the query log for $site(X)$, this is straightforward. Then, Y can collect $queries(X)$ by filtering the subset of requests that show clicks on pages featuring $car(X)$. Even in the case that the URLs in the query log are truncated at the Web site name, it is not a problem to find the full URLs, as explained in [11]. To find the relevant URLs for $car(X)$, the URLs extracted from the query log for $site(X)$ are inspected on-line, this gives the number of pageviews from S to $car(X)$. Since Y knows the contribution of pageviews from S to the sales of $car(Y)$ and since X, Y address the same target group, Y can therefore make an approximation of the sales of $car(X)$ induced by S. From this approximation, Y can estimate the the total number of sales of $car(X)$, which allows Y to have a point of comparison with its competitor which is a good indicator of its performance in the Web.

Additionally, Y can compare the queries from $queries(Y)$ and $queries(X)$ to learn new things about their own on-line presence. For example, they can see which queries exist in $queries(X)$ but not in $queries(Y)$ (i.e., $queries(X) - queries(Y)$) and decide to purchase search engine advertising for these queries (*keyword stealing*). Also, they can use other Web site optimization techniques based on these new queries, e.g.: placing them as keywords and anchor text on their Web site.

Furthermore, if the query log has been additionally protected by fully anonymizing URLs, then Y can use their own $queries(Y)$ in the published log to find the anonymized identifier for $site(X)$. This is a more complex de-anonymization process which is described in detail in the following sections.

□

The example above, although simple, differs in two ways from previously known privacy breaches in query logs [23]: First, information disclosure is achieved by *combining* data sources, i.e. a published query log, a private Web server log and publicly accessible Web sites. As we will see in Section 7.2, privacy preservation methods concentrate in protecting a single type of data source, the query log, rather than a combination of data from multiple *independent* sources. Second, the background knowledge used to disclose confidential company data is not of arbitrary nature: The adversary uses data that is very similar in content and format to that to be disclosed. This makes the confidentiality preservation slightly easier to define but no less challenging to achieve.

The threat of confidential information disclosure is not limited to business institutions that use the Web for marketing and sales. In Example 1, one may replace companies X and Y with politicians who are candidates for the same region and use their Web sites to inform, conduct polls and to discuss with citizens.

This chapter is organized as follows. In Section 7.2, we discuss related work on privacy and privacy preservation with anonymization methods. In Section 7.3, we specify the types of adversaries expected in a business privacy breach scenario, we introduce the general setting for adversarial activities against a business or a public person's Web site and illustrate with three concrete attacks. Section 7.4 presents a query log anonymization method that is based on the removal of information disclosing queries. In the same Section, we estimate the information loss produced by our method. Section 7.5 describes our experiments on the anonymization of a real query log. Finally in Section 7.6, we summarize our findings and provide an agenda for further research on this subject.

7.2. Related Work

Regardless of the extensive research on privacy preservation, the term "privacy" itself is not unanimously defined. An overview of different privacy definitions can be found in [36]. For the purpose of our study, we make a distinction between two broad categories of privacy preservation methods in the context of data analysis. The first category involves "anonymization" methods, which prevent the identification of single individuals (or of some of their features) in a database. The second category involves "cryptographic" methods or protocols for learning a statistical model in a collaborative way, *without* disclosing data that is private to the agents involved. Our work adheres to the first category.

A seminal solution to the *anonymity preservation* challenge has been proposed by Sweeney in [93] and studied intensively since. Sweeney introduced the concept of *k-anonymity*, which ensures that each information request contains at least k (anonymized) individuals with the same values, so that it is not possible to identify one individual in particular.

User privacy in search engine query logs has become a subject of research very recently, among other things, in response to the privacy breaches detected in the anonymized query log published by AOL. Kumar *et al* [63] propose query tokenization for query log anonymization and apply a secure hash function upon each token. However, they show that even this anonymization does not guarantee privacy and explain how statistical techniques on a reference log can still be used to disclose private information.

Jones *et al* [58] provide a detailed description of a data analysis process that leads to information disclosure in a query log. They show how the combination of simple classifiers can be used to map a series of user queries into a gender, age and location, showing that this approach remains very accurate even after personally identifying information has been removed from the log. They emphasize that a user can be identified by a real-life acquaintance; this type of person has background knowledge on the user (e.g. location, age, gender or even access the user's browser) and can use it to disclose the activities of the user in the log.

Adar [11] elaborates on vulnerabilities in the AOL log and shows that traditional privacy preservation methods do not transfer directly to query logs. Adar also points out that k-anonymity is too costly for query log anonymization, because this type of dataset changes very rapidly. Two user anonymization methods are proposed, whose goal is to balance the achieved privacy and the retained *utility*, i.e. the usability of the anonymized log for statistical analysis.

The term *utility* refers to the *data utility* of the anonymized log for the purposes of non-adversarial information acquisition. Verykios *et al* [97] count utility as one of the important features for the evaluation of privacy preserving algorithms, next to the *performance* of the algorithm, the *level of uncertainty* with which the sensitive information can be predicted and the *resistance* to different data mining techniques. In particular, we model the utility of our log anonymization approach based on the information loss, as explained in Section 7.5.

All of the previously mentioned studies on query log privacy concentrate on the protection of user privacy. To the best of our knowledge, we are the first to study the dangers of a privacy breach in an independent area. The disclosure of confidential information about companies can be achieved by exploiting query keywords, clicked URLs and their rank positions in combination with the contents of the companies' Web sites. This information is not necessarily related to the disclosure of data about users who issued queries and visited URLs. Therefore, advances towards protecting user privacy do not guarantee business privacy. From now on, we will prefer the term *confidential*, over the term *private*, to refer to the protection of non-public information about an institution.

7.3. Framework for Breaches on Confidential Information

Our model for confidentiality breaches through adversarial attacks covers the following aspects: (i) types of adversaries, which are defined in terms of their goals and the amount of information they own, (ii) different sources of information that can be combined and exploited, (iii) vulnerabilities found in the query log and (iv) several attacks that can be performed by adversaries to obtain confidential information.

7.3.1. Defining Privacy for Businesses

Before specifying which information is considered *private* or *confidential* for an company on the Web, we first elaborate on the notion of private information for users on the Web. Many people have a personal Web site or Web page in which they publish a large extent of information about themselves. Other users instead are very careful about not revealing personal information. It is generally understood that what a user publishes is a matter of his or her own judgement. For example, some people put pictures of themselves in the Web, while others do not. Some employees may add a link from their company's Web page to their personal Web page and vice versa. Some users may list their hobbies in their page at their employer's site and some students may do alike in their University's Web page. Hence, what a person considers private varies from person to person.

Similarly, many institutions have Web sites. What an institution chooses to makes public in their Web site varies substantially. For example, some companies publish the number of hits on their Web site and others do not. Some companies maintain official blogs for interaction with their customers and with any interested users, and others may publish frequently asked questions. In relation to Example 1, many companies decide to provide detailed periodic reports of their sales per channel, as a refinement of periodic reports that they publish. On the other hand, other companies share these figures only with their employees. For instance, whether a company decides to publish statistics about failed sales opportunities (contacts with customers that failed to result in a sale) is purely a matter of institutional judgment.

In this study, we define as *private* or *confidential all* of the information about a company which (a) is not published data and (b) cannot be concluded trivially by combining publicly available information. In a similar way, we say that data or information about a company becomes *public* when (i) it is published by the institution itself or (ii) it is published by some entity that is legally authorized (or obliged) to do so. According to these definitions, a person's office phone number is public if the person itself publishes it or if their employer does it (and is authorized to). The number of hits on a company's Web site is public if the company itself publishes them or someone with appropriate authority does. Such an authorization for publishing may be given by the institution itself, e.g. a company may authorize a third-party to publish their annual sales in the Web as part of a service agreement.

Following the previous definitions, a *confidentiality breach* occurs if private information is disclosed in one of the following ways: (i) Non-authorized publication of private information, including the publication of anonymized information that can be de-anonymized, or (ii) non-trivial combination of public and private information, where the latter includes also the use of personal expertise. We concentrate on the first case and focus on confidentiality breaches that occur through the de-anonymization of an anonymized query log (or part of it). However, as we will see, such a breach usually involves the combination of multiple sources, both private and public ones, as mentioned in the second case.

Consequently the publication of the anonymized AOL log [13] constitutes a confidentiality breach. Referring to Example 1, the disclosure of the sales for company X's car *is* a confidentiality breach *unless* X has authorized the search engine to publish their query related traffic data.

7.3.2. Adversaries

Monitoring competitors' activities within a business domain (or market) is a legitimate and necessary task for a company. Decision makers use tools such as SWOT analysis on their businesses (SWOT stands for Strengths, Weaknesses, Opportunities and Threats) and Knowledge Maps [103] (which positions a company regarding its knowledge assets and expertise). These types of analysis require an understanding of the competitors in the business. The term *Competitive Intelligence (CI)* is used to describe the activities undertaken by a company to acquire information about its competitors. This information is important for the company's own strategic decisions. For a short overview on CI we refer the reader to the study of Vedder *et al* [96]. The role of public sources for CI and their analysis through data mining is discussed by Zanasi [105] and Vedder *et al* [96].

Monitoring the activities of companies is not a task performed only by competitors. For example, investment consultants and institutions that perform market studies are agents that regularly collect data about companies' activities and extract knowledge from it. They gather among other things, risk and growth indicator values.

It must be noted that information acquisition about a company's activities is a legal and well-expected operation for both types of agents (competitors and third-parties) *as long as* this information is obtained through legitimate means. This also holds for the analysis of publicly available documents, such as those that appear in a company's Web site, and knowledge extraction from a published query log. For example, submitting artificial queries to a search engine with the purpose of identifying them later in the published query log, is an activity whose legitimacy is less clear.

We consider two types of agents that are interested in extracting confidential information from a Web site:

- *Competitor adversary:* This agent tries to disclose confidential information about its own competitors. Usually, this agent already has background knowledge about the market share, product portfolio and its competitors' activities (such as, research, marketing and alliances). This background knowledge can be combined with an anonymized public query log. Furthermore, an important data source that this adversary can exploit to disclose confidential competitor information is *the private log* *of its own Web site*. As discussed in Section 7.3.5, this private log can be juxtaposed to the public query log for de-anonymization purposes.

- *General adversary:* This agent tries to collect confidential information from arbitrary Web sites, without having a particular *target* site in mind. This adversary is the counterpart of a company that collects publicly available information to perform market studies, investment consulting or search engine optimization for Web sites.

We refer with the term *adversary* to all of the agents that attempt to disclose confidential Web site information through the combination of data from a published query log and other sources. We use the term *attack* to refer to a sequence of activities that result in the disclosure of confidential information. We stress that our use of both terms, despite their apparent negative connotation, *do not imply on their own an illegitimate action*. Consequently attacks performed by adversaries are not necessarily illegal in this context.

7.3.3. Data Sources for Attacks on a Query Log

Our framework for attacks on a public query log considers three main data sources, shown in Fig. 7.1:

- *Anonymized query log:* This is a search engine query log which is made publicly available. It contains confidential information related to people and institutions and is anonymized according to a particular anonymization scheme.
  Log releases can take place periodically, this is the case we focus on. We denote the release published at time point i as L_i. The released logs do not need to be consecutive, nor need to be of the same size.

- *Search engine query results:* These are live results that the adversary agent obtains by issuing queries on-line on the search engine.
  If the adversary intends to exploit live results, then it must issue queries periodically, so that a set of live results R_i is available for the log L_i at the moment it is released. If the queries are not known in advance but are obtained through inspection of L_i, then there is always a time gap between the live results and the query log. However, most of the queries can be predicted beforehand using the adversary's background knowledge, or leaned fairly quickly when the log is released. Even with a large delay between the search results and the query log, many of the most relevant results are preserved.

- *Web site log:* This is the log S_i of the adversary's Web site (registered by their Web server), or of some other site that is available to the adversary. This log contains private information, to which the adversary has authorized access. Since Web servers record data continuously, the log S_i can be always aligned with L_i.

7.3.4. Vulnerabilities in the Query Log

In our model for confidentiality breaches we assume that the query log has the following signature (very similar to that used in the AOL log):

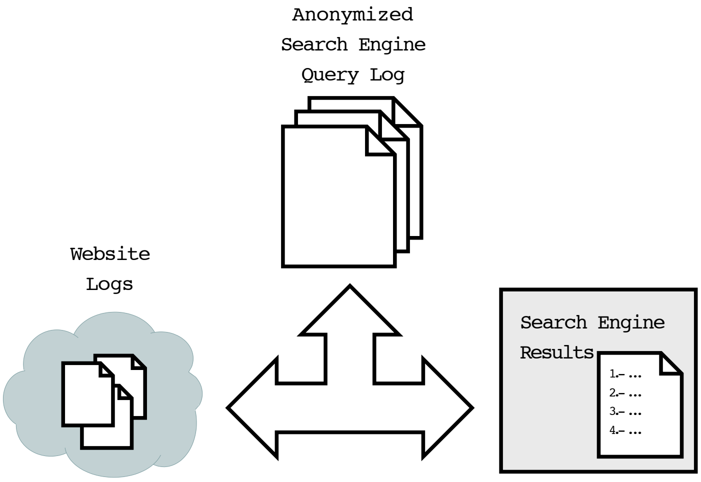


Figure 7.1: Data sources that can be exploited in confidentiality breaches

```
ANONuserId, query, timestamp, ANONclickedURL
```

Basic Anonymization

We assume that the query log contains one entry for each URL (or document) that is returned as a result for a query *and* then clicked by the user. The prefix `ANON` indicates that the field is anonymized by default. Therefore, the user identifier and the clicked URL are anonymized, while the query keywords and the timestamp of the request are not. Although more fields can be anonymized in this signature this reduces the usefulness of the log for analysis. In this work, we assume that the anonymization *of a query log entry* does not go beyond these already masked fields. Nevertheless, we apply further anonymization steps to the log as a whole.

Concerning the field `ANONclickedURL`, we note that there are many ways of anonymizing a URL. In the AOL dataset [1], URLs are truncated to the site's name (i.e. up to the first '/'). Instead, we assume that the `ANONclickedURL` consists of the *anonymized* identifier of the site and the *anonymized* identifier of the document *within the site*. This means that two documents `doc1.html` and `doc2.html` of site `www.somesite.com` would be anonymized into `anonsiteA.X` and `anonsiteA.Y` respectively, where `anonsiteA` is the anonymized ID of the site and `X,Y` are the anonymized identifiers of the two documents. This provides a rudimentary protection of the identity of the sites and their contents while allowing analysis which differentiates among different sites and considers the size of a site. Therefore, the new signature of the query log for our analysis is:

```
ANONuserId,query,timestamp,ANONsite.ANONclickedURLinSite
```

Vulnerable Queries

For query log anonymization, we concentrate on the identification and removal of what we consider to be *vulnerable queries*. We consider *vulnerability* in the context of k-anonymity [93]. This means that attacks are successful if they acquire less than k objects from the log and that anonymization must ensure that queries associated with less than k objects do not appear in the published dataset. The value of k is determined by a human expert. In Section 7.5, we study the impact of the values of k on the size and the amount of information retained in the query log. We consider two types of vulnerable queries:

- *Over-restrictive queries:* These type of queries are those which return less than k documents in the search engine. These queries present a privacy breach because their set of anonymized URLs can be mapped to k, or less, *real* URLs. To deal with this issue, one may either eliminate these queries or generalize them. Generalization can be implemented, for example by replacing the query with more abstract terms and merging all of the results that satisfy these terms. The initial approach taken in this study is to solve this vulnerability by eliminating these queries from the log.

- *Well-targeted queries:* The second type of vulnerable queries involves queries that contain the target URL, or at least the site of the URL, as a keyword. In this case, these queries are a subcategory of *navigational queries* discussed by Broder in [34]. The user knows some part of the target URL, for example the site name and/or some part of the URL's name. The user submits this information in the search engine to obtain the exact URL of the target page, essentially using the search engine as a bookmark manager. A navigational query which fully discloses a site is obviously vulnerable. For example, the query term `amazon.de` points to a specific site in most major Web search engines.

From the viewpoint of privacy preservation, a navigational query that does not fully disclose the site may still be vulnerable. This is the case if the distribution of clicks among the returned URLs is highly skewed towards the same site. For example, if most of clicks to the query `london transport` were found to go to the site `http://www.tfl.gov.uk/` (the homepage of London transport), then the query would be considered vulnerable[1].

The reason for the vulnerability of well-targeted queries is that an adversary can use background knowledge to de-anonymize the URLs in the log. This background knowledge can be acquired in two ways: First, the adversary may use another third-party published query log[2], where the sites are not properly anonymized, and derive the click distribution from that log. Second, the adversary may issue the vulnerable query in the live search engine, from which the published query log comes from, and study the distribution of returned (not clicked) URLs among the sites. Although not as accurate as the first approach, this

[1] The click distribution can be built by analyzing the query log.

[2] The published query log of AOL, which is taken off-line and should not be used for analysis, may be misused by a malicious party to reconstruct the click distribution.

distribution might allow an approximation of the unknown click distribution. A solution for this kind of vulnerability can be the removal of these queries or anonymize their keywords.

Even after the elimination of over-restrictive and well-targeted queries, the query log may still be vulnerable. Next, we describe two attacks against an anonymized query log: The first attack exploits properties of *pairs of queries* and calls for a more elaborate log cleaning method, proposed in Section 7.4. The second attack exploits the adversary's own Web site log, i.e. a source of background knowledge that is beyond the control of the owner of the query log.

7.3.5. Attacks on an Anonymized Query Log

We introduce a first attack on an anonymized query log, assuming the main data sources presented in Section 7.3.3. This attack juxtaposes the published query log $L \equiv L_i$ with the query results of the live search engine $R \equiv R_i$. Then we present two variations of this attack based on the incorporation of additional background knowledge or data from the adversarial side. In these attacks we assume that over-restrictive and well-targeted queries have already been eliminated from the log.

ATTACK 1: Combining Queries

This attack exploits pairs of queries which have a non-empty intersection between their *clicked* result sets in the query log. For some pairs of queries q, q' it is likely that a search engine returns overlapping sets of URLs, i.e. $results(q) \cap results(q') \neq \emptyset$. For example, this could be the case for different queries that contain one or more common keywords. However, these queries become vulnerable *only* when $clickedURLs(q) \cap clickedURLs(q') \neq \emptyset$. Our experimental evaluation shows that this situation is rare but not negligible.

First we assume that the agent launching the attack is an *competitor adversary* who tries to extract confidential information about some specific Web sites. Later we show how this attack can be extended to a *general adversary*. The competitor adversary attack is defined in ATTACK 1.

This attack combines the background knowledge of the competitor adversary about queries that retrieve target pages, the published query log and the live search engine. The adversary collects live URLs from the search engine (*Step 2*) and locates the anonymized URLs retrieved by the same queries from the published query log (*Step 3*). Then, the queries with non-empty intersection of clicked results are identified (*Steps 4 and 5*) and juxtaposed to the displayed results from the live search engine (*Steps 6 and 7*).

Step 7 of the attack repeatedly juxtaposes the intersection of anonymized clicked results to the intersection of live results for each pair of queries. As soon as an intersection has only one element it can be de-anonymized. Then, the URL and its anonymized identifier are removed from the lists, which in turn become smaller. The attack stops when no more removals take place.

The success of this attack depends on satisfying the first condition in *Step 7*. If there is no pair of queries, whose intersection of clicked results contains only one record, then the attack cannot start. If the counterpart intersection of displayed live results is much larger, then the attack may fail. A couple of remarks are due in this context: (1) We make the assumption that the likelihood of finding *one* pair of queries with a single common URL is not negligible, this is verified by our experimental results in Section 7.5. Even if the condition does not hold for a *pair* of queries, it may hold when combining 3, 4, 5, ...queries. As soon as the attack starts, URLs may be disclosed. (2) A set of displayed results is usually much larger than the set of clicked results. However, the adversary may exploit publicly available knowledge about the clicking habits of search engine users and reduce the sets of displayed results only to the first few entries for each query. As before, the adversary may combine 3 or more queries to reach an intersection that has only one element.

A variation of this attack can also be performed by a *general adversary*. This agent has no background knowledge to perform *Step 1*, but is still able to identify all pairs of queries that have a non-empty intersection of clicked results (*Step 4*). Then, these queries can form a (fairly large) set of queries $setQ$, which is launched against the live search engine to build the sets of displayed results. Once these sets are built, the mission-critical *Step 7* of the attack can be launched.

ATTACK 1a: Exploiting a Private Web Site Log

According to the data sources shown in Fig. 7.1, additionally an adversary can exploit the private Web server log of a given site. An competitor adversary is likely to own this type of log – the log of its own Web site. This extra source of information can be used to enhance ATTACK 1.

A site's Web server registers all of the requests sent to the site. A typical a log entry contains, at least, the target URL, the time of the request, the user agent that generated it and the IP address of the user. Furthermore, most Web site logs also include the *referrer URL*, i.e. the URL from which the request to the site was initiated. If the referrer is a search engine's result page, then the referrer URL contains the keywords of the query used to retrieve the target URL from the results.

This data can be used to de-anonymize information from the published query log. In particular, the adversary can find the queries and clicked URLs recorded by its own Web server. Then, they can match the timestamps in the server to the timestamps in the published query log and finally match the anonymized identifiers of the clicked URLs to their own URLs. This can be done easily even if the times on both logs are not synchronized, for example, using two consecutive request with different queries to the Web site.

The "benefit" of matching a Web site log with the published query log is a twofold: First, the adversary can attempt to re-engineer the algorithm used for URL anonymization. Second, the adversary use the already de-anonymized URLs in *Step 7* of ATTACK 1, increasing the probabilities of disclosing its competitors anonymized URLs. This attack can become a greater threat when many Web sites collude joining their query logs.

ATTACK 1:

1. Define a set of queries $setQ$
   which are known to return URLs of the target competitor's Web site in high-ranking positions.

2. Submit $setQ$ to the on-line search engine,
   collect the results acquiring the $liveResults(q)$ set for each query $q \in setQ$.

3. Find $setQ$ in the anonymized query log L and
   obtain for each $q \in setQ$ the anonymized identifiers of its clicked URLs $clickedURLs(q)$.

   The next task is to map anonymized identifiers to the live results.

4. Build the matrix of intersecting queries M:
   the cell $M[i, j] = clickedURLs(q_i) \cap clickedURLs(q_j)$ for each $q_i, q_j \in setQ$.

5. Clean the symmetric matrix M
   by removing all but each upper part and by eliminating cells with empty intersection.

   The result is a list of lists $\mathcal{M}$ where $\mathcal{M}[i]$ is the list of non-empty intersections on q_i and queries with index larger than i. Then, $\mathcal{M}[i][j] = clickedURLs(q_i) \cap clickedURLs(q_j)$ for $j > i$.

6. For each pair of queries $(q_i, q_j), j > i$ with $\mathcal{M}[i][j] \neq \emptyset$
   compute the list element $\mathcal{L}[i][j] = liveResults(q_i) \cap liveResults(q_j)$.

7. Traverse the lists $\mathcal{M}, \mathcal{L}$ and
   juxtapose them to de-anonymize URLs:

   - If $|\mathcal{M}[i][j]| = |\mathcal{L}[i][j]| = 1$,
     then the anonymized identifier $ANONu$ constituting $\mathcal{M}[i][j]$ corresponds to the de-anonymized URL u that constitutes $\mathcal{L}[i][j]$.

     Remove u from all entries in $\mathcal{L}$ and $ANONu$ from all entries in $\mathcal{M}$.
   - If $|\mathcal{M}[i][j]| > 1$,
     then proceed to the next j and then to the next i.

To avoid the consequences of ATTACK 1a one more constraint should be placed in the anonymization process of the query log: The results displayed by the search engine for any given query *must contain URLs of at least k different sites*, so that k-anonymity can be pursued.

ATTACK 1b: Exploiting Disclosed User Data

Many adversarial attacks on anonymized query logs, discussed in related work, aim towards identifying the actions of a particular user. We study a variation of ATTACK 1 in which a search engine user can be exploited to disclose confidential information in the query log. In particular, if the adversary identifies a (single) user in the query log it can then acquire knowledge about the results clicked by this user. With this information the adversary can trace the user and its clicked URLs in the published query log and map the anonymized URLs to the real ones. Similarly to ATTACK 1a, these de-anonymized URLs can be used in *Step 7* of ATTACK 1.

In periodical releases of an anonymized query log, this attack can be automated with an agent that submits queries to the search engine regularly. This user is then traced when the query log is published.

This attack can be prevented by avoiding the identification of a particular user with conventional privacy preserving methods for query log anonymization. As pointed out in Section 7.2, there are already efforts towards this problem.

7.4. Query Log Anonymization

We propose a heuristic method for the anonymization of a query log, intended to prevent the attacks described in the previous section. This anonymization process is meant to reduce the vulnerabilities identified in subsection 7.3.4 and prevent the attacks described in subsection 7.3.5. Our approach is based on the removal of the objects that cause the vulnerabilities, i.e. of vulnerable queries and queries that generate non-empty result intersections. We study the resulting data after the anonymization process to measure the amount of retained information in the log.

7.4.1. Heuristics for Log Anonymization

Our approach is based on a graph representation of the query log, in which all instances of a query are modeled as a node and two nodes are connected (by an undirected edge) when the intersection of their clicked URLs sets is not empty. In other words, a node represents the aggregation of all the occurrences in the query log of a particular query, and the clicked results are all of the URLs clicked by users for instances of that query. This type of query log graph is discussed by Baeza in [16]. A simple toy example of this model is presented in Fig. 7.2.

The conditions needed to prevent ATTACK 1 can be formalized into a well-defined optimization problem on the query log graph. The idea is to disconnect the graph while trying to preserve the most *important* nodes. If we consider the importance of a particular node as the weight of the node, then our problem translates into that of preserving the *maximum weighted graph* or the *maximum (weighted) independent set*. This is an NP-hard problem, therefore we use an heuristic approach to solve it. We begin by defining a density measure

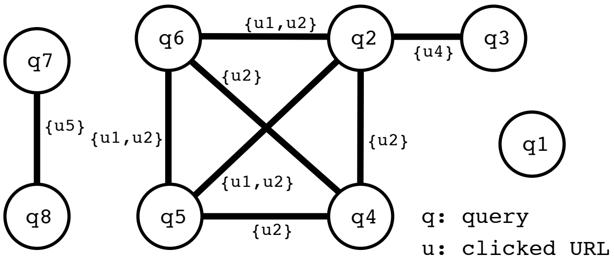


Figure 7.2: Example graph representation of a query log

for the graph, then we solve a baseline case in which all node weights are equal to 1 and later we introduce variations based on the relevance of each node.

Definition 1 (Graph density)**.** *Let $G(V, E)$ be the graph of a query log, where V is the set of queries in the log and E is the set of edges. Two nodes $v, v' \in V$ are connected with an undirected edge $(v, v') \in E$ if and only if $clickedURLs(v) \cap clickedURLs(v') \neq \emptyset$.*

The "density" of the graph is the likelihood of finding an edge among any two nodes and it is defined as:

$$Density(G) = \frac{2 \times |E|}{|V| \times (|V| - 1)}$$

i.e. the graph density is based on the cardinality of the set of edges and the set of nodes.

We use a greedy heuristic to solve the baseline case according to the concepts of graph *density* and the *degree* of each node. This heuristic is defined as follows:

Graph Disconnection Heuristic:

Input: Graph of the query log $G(V, E)$

Output: Disconnected graph

1. Sort V on node degree.
2. Identify the node with the highest degree, v_{max} and the set of its neighbors $X := \{x|(x, v_{max}) \in E\}$.
3. Remove v_{max}, i.e. set $V := V \setminus \{v_{max}\}$
4. Remove the edges involving v_{max}, i.e. set $E := E \setminus \{e|e = (x, v_{max})\}$.
5. Recalculate the degree of all nodes in X.
6. Recompute $Density(G)$.
7. If $Density(G) \neq 0$ then go to Step 1, else finish.

The graph disconnection heuristic eliminates the query with the highest degree first, i.e. the one involved in the most non-empty intersections of clicked results. This generates the removal of many edges. The heuristic proceeds gradually, until the graph is fully disconnected. This heuristic can be relaxed to stop when the value of Density reaches a preestablished minimum threshold.

7.4.2. Variations on the Graph Disconnection Heuristic

This basic heuristic takes only the connectivity of a query-node into account. However, since one goal of the anonymization is to allow statistical analysis over the anonymized published log, it is reasonable to model the “importance” of the eliminated queries and to remove less important queries first. In this work we will consider that the importance or *weight* of a query can be represented by its frequency or its number of clicked documents. However, the weight can represent any other measure of the significance of the queries.

This leads to two further variations of the basic graph disconnection heuristic, each of which sorts nodes based on a different property (Step 1 of the heuristic):

- *Method 1:* The property used for sorting is the degree of the nodes; this is the basic heuristic.
- *Method 2:* The property used for sorting is the degree multiplied by the inverse frequency of the query in the log, i.e. $\frac{degree(v)}{frequency(v)}$ for $v \in V$.
- *Method 3:* The sorting property is the degree multiplied by the inverse number of clicked documents for the query, i.e. $\frac{degree(v)}{clicks(v)}$.

In *Method 2*, $frequency(v)$ is the number of times that any instance of query v has been submitted to the search engine. In *Method 3*, $clicks(v)$ is the number of times any document in $clickedURLs(v)$ was clicked as a result of v.

7.4.3. Extending towards K-Anonymity

The graph disconnection heuristic and its variations focus on eliminating queries that share URLs among their clicked results. Although the heuristic disconnects the graph, anonymity is not yet guaranteed. In particular, the heuristic does not consider the number of results displayed by the search engine for each query. The number of displayed results is important to prevent attacks to over-restrictive queries. Furthermore, if all URLs returned for a query belong only to one site or just a few sites, then the adversary may be able to disclose them as discussed in ATTACK 1a.

The previous observations can be mapped into requirements for k-anonymity. In particular, a query must display at least k results in the search engine, which come from at least K different sites ($k \geq K$).

Thus, the complete anonymization process encompasses the removal of (a) all vulnerable queries, according to subsection 7.3.4, (b) all queries that return less than k documents and (c) those returning documents from less than K sites and (d) all queries that contribute to a non-zero density of the query graph.

This anonymization process inevitably incurs the loss of data and of potentially useful information. We model *information loss* as the decrease in the volume of the query log with respect to (i) the number of queries and (ii) the number of clicks. In the following Section, among other things we study how different values of k and K influence the retained volume of the query log and how the log shrinks with each of the variations of the graph disconnection heuristic.

7.5. Experiments

We study the behavior of the graph disconnection heuristic on a real query log. The goal of the experimental evaluation is to gain insight on the process of log anonymization and the information loss that it generates. First we present the log used for the experiments and report some of its characteristics. Then, we study the behavior of the three graph disconnection variants presented in Section 7.4. Finally, we elaborate on the effects of pursuing k-anonymity, i.e. of eliminating queries that are associated with less than k clicked documents or less than K sites. In this evaluation we do not analyze the specific case of well-targeted queries, since we consider that it requires more extended evaluation. Nevertheless, we consider all of the other previously mentioned attacks.

7.5.1. Dataset

We performed our experiments on a query log sample from the Yahoo! UK & Ireland search engine. The sample contains consecutive request registered by the search engine for a certain period of time[3]. For the purpose of this evaluation we did not use the raw log data but rather worked with its graph representation. Since our goal is to study log anonymization through graph disconnection heuristics this graph is appropriate for this task.

The query log graph representation is an application of the graph models presented in [18] and can be computed fast. The final graph contains over 3 million nodes and its computation took approximately 2 hours on a dual core AMD Opteron^{TM} Processor 270 with 6.7 GB of RAM[4]. The density of this graph is low to begin with, equal to 0.000089.

First, we identified all of the connected components in the graph and computed their size distribution, as shown in Fig. 7.3. Without considering components with only one element, we find that there is a large connected component that includes 70% of the connected nodes. The second largest component found is only 0.01% of the size of the largest one. Also, more than 80% of the clicks to documents in the log proceed from queries in the large component. The density of this component is 0.00075. We focus on studying the effects of the attacks on the largest component, since it involves most of the connected elements in the graph, which are the most vulnerable to attacks. Also the effects on the largest component represent an approximation of the worst case of log volume loss. Since

[3] The extension and date of the log is omitted to preserve the confidentiality of the data.

[4] The algorithm uses only one of the CPUs and less than 4 GB of RAM.

one of the worst scenarios of anonymization, according to our heuristics, is for a query log graph that is completely connected.

Next, we analyzed the distribution of the node degrees in the large component. If the number of edges per node were to follow a power law, then this would indicate that the edges can be reduced quickly by node removal: The graph would become disconnected very fast by removing a few high-degree nodes [12]. However, as shown in Fig. 7.4, the degrees do not follow a power law, which does not guarantee that quick graph disconnection can be achieved.

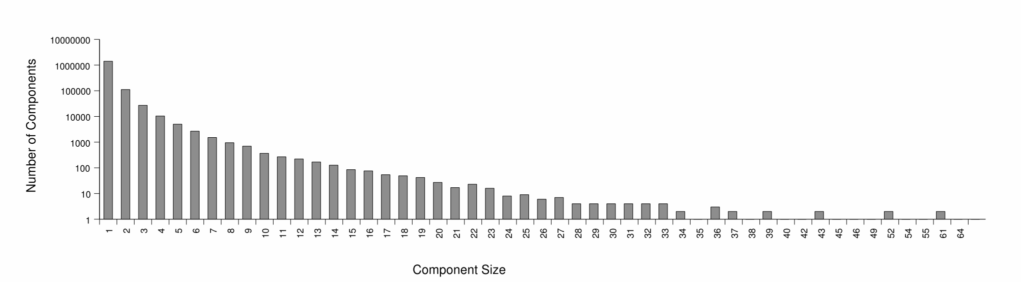


Figure 7.3: Size distribution for the connected components of the query log graph

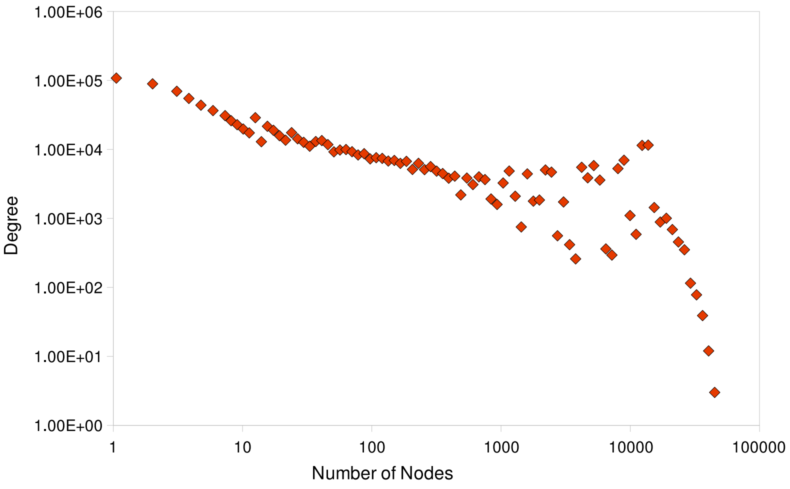


Figure 7.4: Degree distribution in the largest connected component of the graph

7.5.2. Disconnecting the Graph

We applied the three variations of the basic heuristic presented in Section 7.4 and studied the decrease in the size of the largest component in the log. We consider the reduction of the volume of the log as an initial approximation of the information loss induced by the anonymization heuristics.

We show the effects of the gradual removal of nodes on the volume of *retained queries* (Fig. 7.5) and on the *remaining document clicks* (Fig. 7.6). In this experiment we define as:

- *retained queries* the sum of the frequencies (in the query log) of the queries represented by the remaining nodes, and
- *remaining document clicks* as the sum of the clicks to documents from the queries retained in the log.

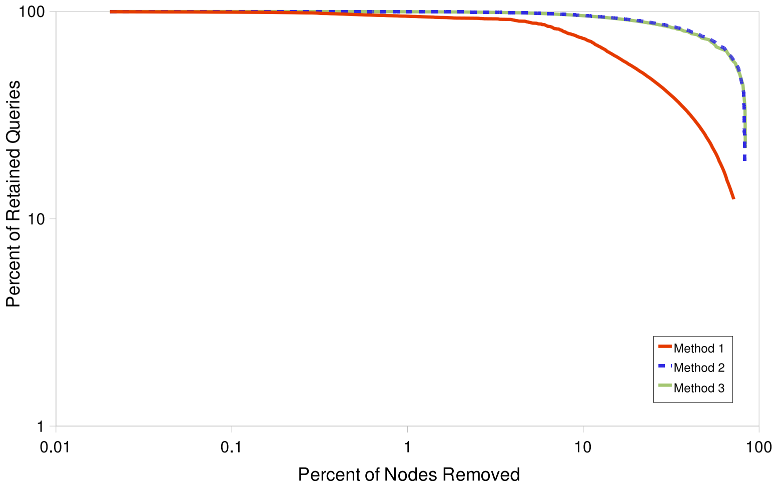


Figure 7.5: Percent of retained queries during graph disconnection

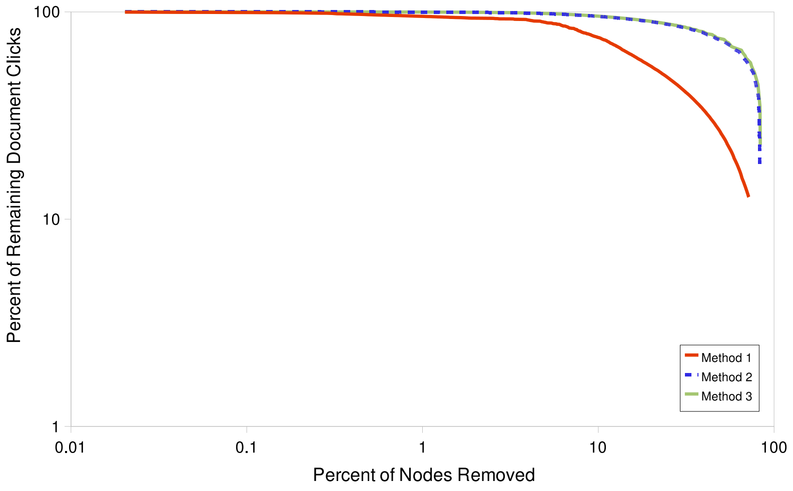


Figure 7.6: Percent of remaining document clicks during graph disconnection

In Fig. 7.7, we show the decrease in the number of retained edges as the percentage of removed nodes approaches 100%. The number of retained edges does not serve as a utility indicator, since all edges must be removed anyway. It rather indicates the speed at which the graph gets disconnected. Between 70% and 80% of the query nodes need to be removed in order eliminate all of the edges. The effect of the removal of nodes on the value of the density is presented in Fig. 7.8. The complete disconnection of the graph component dramatically affects the number of nodes remaining. Nevertheless, the value of the density in the large component can be decreased to under 0.000038 removing less than 10% of the nodes. This value significantly decreases the number of edges, to only a 6.3% of their original value, while preserving over 95% of the retained queries and document clicks, according to Methods 2 and 3 in Figures 7.5 and 7.6.

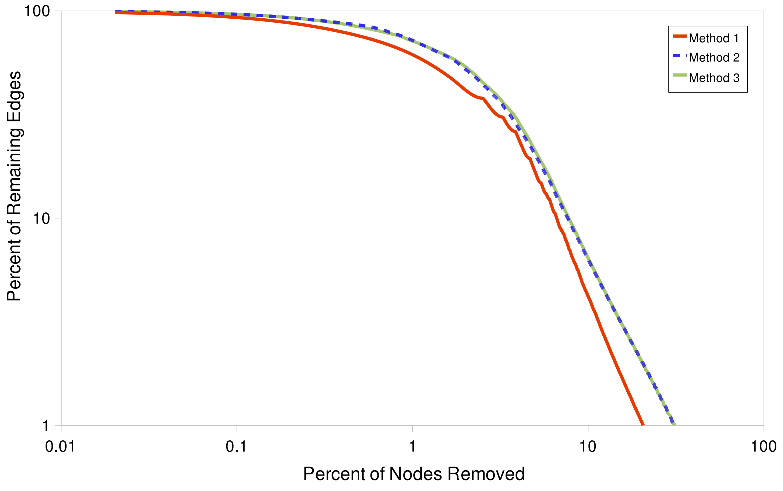


Figure 7.7: Percent of retained edges during node removal heuristics

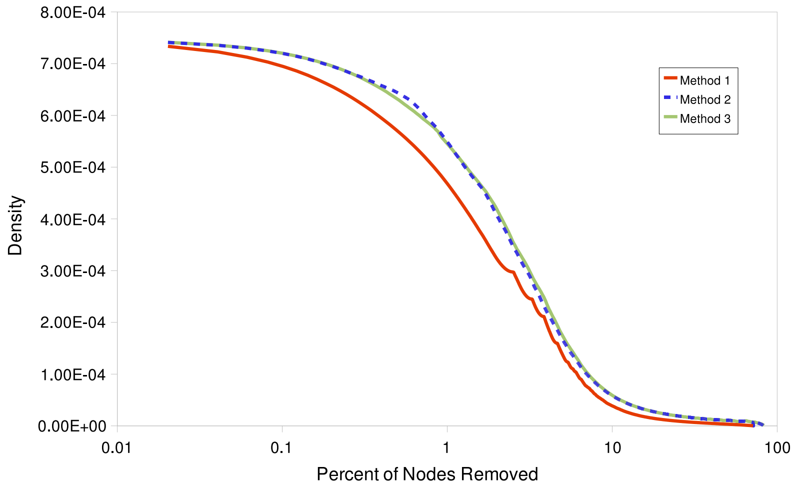


Figure 7.8: Density during node removal heuristics

The results show that Methods 2 and 3 perform better than Method 1. These methods consider both the degree of a node and its inverse weight, thus favoring the elimination of highly connected infrequent queries. As we can see in Fig. 7.5 and Fig. 7.6, Method 2 and 3 remove less queries and clicks than Method 1. So, they produce an anonymized log with larger volume.

7.5.3. The Impact of K-Anonymity Enforcement

As pointed out in subsection 7.4.3, queries that *display* less than k results in the search engine violate the k-anonymity requirement for conventional anonymization. Even if a query returns more than k documents, a confidentiality breach may occur if the documents come only from very few sites. So, we consider anonymity enforcement with respect to k results and to K sites, where $k \geq K$.

Similarly to the graph disconnection methods, we measure the impact of anonymity enforcement on the basis of retained queries and remaining clicked documents. Note that when removing a query, this eliminates all of the instances of that query in the log. This includes all of the documents clicked as a result of this query.

Removing Queries the Display less than k Results in the Search Engine

In Fig. 7.9, we show the decrease of the log volume (percentage of retained queries and clicks) as we remove queries with $n < k$ displayed results. We are particularly interested in queries that return less than 10 results, because 10 is the default number of URLs for the first page of search engine results: For queries returning more than 10 results, we do not know how many results were available, only the number of results *displayed* or shown to the users. On the other hand, for a query with less than 10 results we know that these were the *only* available results, and that they were all displayed to the user in the first results page.

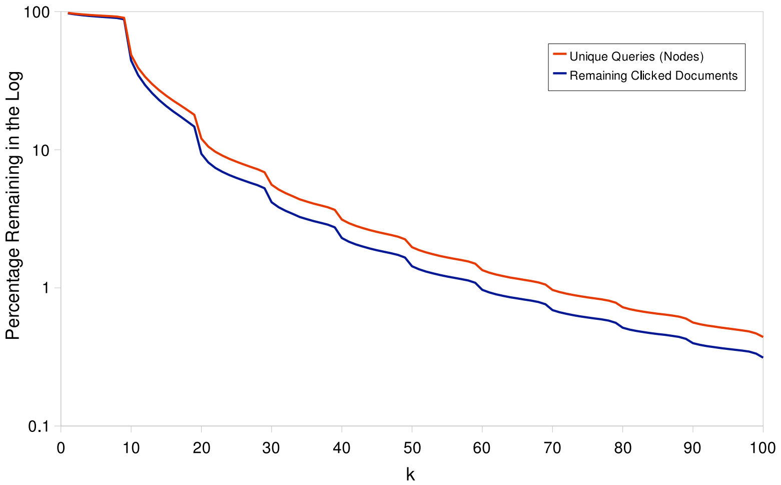


Figure 7.9: Log volume decrease when removing queries with less than k results

k results	percent of retained queries	percent of retained clicks
1	97.9	97.2
2	96.5	95.5
3	95.5	94.1
4	94.6	93.1
5	93.9	92.2
6	93.3	91.4
7	92.7	90.7
8	91.9	89.9
9	90.2	87.9
10	48.6	44.2

Table 7.1: Log volume decrease when removing queries with less than k results for $k \leq 10$

In Table 7.1, we show that the removal of queries with $k \leq 7$ displayed results does not affect the overall volume of the log, because these queries are not frequent nor have many clicks. We can see in the table that the largest portion of the log refers to queries with $k \geq 10$. Therefore enforcing k-anonymity for $k = 7$ would not affect very much the log volume.

Removing Queries with Results from Less than K Sites

In Fig. 7.10, we show the decrease of the log volume (percentage of retained queries and clicks) as we remove queries that return results from less than K Web sites. Table 7.2 shows in detail how the removal of queries with $K \leq 3$ different Web sites does not affect the overall volume of the log, as these cases are neither frequent nor have many clicks. In Table 7.2 we can also see that the largest portion of the log refers to queries with $K \geq 8$.

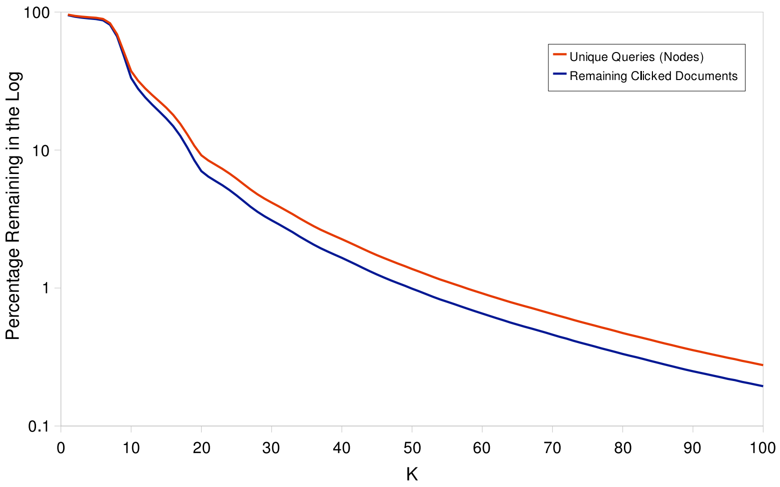


Figure 7.10: Log volume decrease when removing queries with less than K sites

K Web sites	percent of remaining queries	percent of remaining clicks
1	95.7	94.9
2	93.8	92.5
3	92.5	91.0
4	91.6	89.9
5	90.8	88.9
6	89.1	87.1
7	82.9	80.8
8	69.0	66.6
9	50.8	47.5
10	37.3	33.3

Table 7.2: Log volume decrease when removing queries referring to less than K sites for $K \leq 10$

7.6. Chapter Conclusions and Future Work

We have presented a new challenge for privacy preservation in query logs. While most of the research on privacy preservation in this context focuses on protecting private data about users, we show that the protection of confidential information of business institutions is an independent and no less challenging issue. We have formulated this new issue by showing types of adversaries and explaining attacks towards a published, anonymized query log. Then, we have proposed a heuristic approach that removes those queries from the log, which can be exploited in adversarial activities.

We have tested the three variations of our heuristic approach experimentally with a real query log. The variants which sort queries on connectivity divided by query and document click frequencies yield the best experimental results in preserving the most amount of log volume. However the complete removal of edges in the largest connected component of the graph still has a striking effect on the remaining log volume. This can be improved if the density of the graph is set to an acceptable threshold, reducing significantly the number of edges in the graph while still preserving most of the volume of the log. Our type of graph disconnection mostly involves the removal of infrequent queries. Since such queries are more likely to point to identifiable people or institutions, and since their contribution to statistical analysis is expected to be rather limited, their removal is justified.

It is difficult to estimate accurately the information loss induced by our anonymization heuristics. As a first approach we use an estimation of the retained log volume based on the remaining queries and clicks to documents. It is likely that the anonymization heuristics and information loss can be optimized for different types of applications improving the utility of the remaining log. So far, the anonymized log allows tasks which study accesses to Web sites but that do not require to reveal the identity of a particular site. Data mining applications which analyze rare and infrequent queries will not be able to perform as well.

The protection of confidential information is of vital importance for companies that use the Web as communication medium and as a marketing and sales channel. Although our approach is a first contribution towards confidentiality preservation, many open issues re-main. First of all, we have discussed and alleviated specific types of attacks. Different attacks are thinkable and need to be identified, studied and prevented as well. Furthermore, confidentiality preservation is closely associated to k-anonymity: Despite the efforts on preservation of k-anonymity in query logs, there is yet no anonymization method guaranteeing that private information cannot be disclosed.

A perspective worth studying in this context is the role of generalization. Queries can be replaced with some of their keywords, while keywords can themselves be replaced with more abstract terms. We would like to study whether generalization can mitigate the vulnerability posed by infrequent queries that share clicked results.

Chapter 8

Final Discussion

In this Ph.D. work we have explored and analyzed search engine queries as a tool to improve several Web data mining tasks. We have shown that queries are important for understanding user behavior on the Web. Queries provide a simple, straightforward and novel manner for extracting the interesting rules and patterns from the large amount of data in the Web. Furthermore, using queries is fast, because it requires less data processing than analyzing the full contents of documents or Web sites. By aggregating user trails and queries, we filter relevant information from noise and other irrelevant data.

Some of the applications covered in this research include automatic organization of Web documents, similar Web site discovery, improvement of Web site content and structure, and ranking of Web documents. These applications study the Web at two levels of granularity: *Web documents* and *Web sites*. From the Web document perspective, we have seen how to produce compact and efficient Web document models (Chapter 3). Additionally we have analyzed a graph representation of the Web, based on its documents' network structure and query clicks (Chapter 6). On the other hand, from the Web site perspective we have extended the approach described in Chapter 3, to discover similar Web sites independently of their size and structure (Chapter 4). Also, we have studied how queries can be used to improve a Web site according to their users' needs (Chapter 5). Furthermore, in Chapter 7 we have discussed privacy preservation implications of the use of query logs for general data mining tasks.

In detail, this thesis has produced the following results:

- In Chapter 3 we created a new Web document representation, showing that queries are excellent features to describe documents. This model, called the query-set model, reduces by over 90% the number of features needed for document representation, in comparison with the bag-of-words model. Also it improves significantly the quality of the resulting clustering solutions and provides more intuitive cluster labels for humans. This feature dimensionality reduction is very important specially for large document collections, because it reduces computational costs while providing good quality results.

- In Chapter 4 we extend the query-set model used for Web documents, to an aggregated model for Web sites. We also improve upon this model, to create several different combinations of query-based feature spaces. The resulting models allow to measure the similarity of Web sites independently of semantic and structural differences, while using a very small feature set. In particular, the selection of features for Web sites, called the FULLQUERIESPLUS model provides good clustering results while using only 5% of the features used in the full-text approach.

- In Chapter 5 we present a Web site data mining model, centered on user queries. This model aims to improve the contents, structure and organization of Web sites, to make them more intuitive to their users. This model is based on the classification of internal and external queries based on the behavior of the users that generated them. The application of our prototype to two test cases showed very good results by generating suggestions which improved traffic and navigation in the sites.
- In Chapter 6 we present and analyze a new graph representation of the Web, the union of the hyperlink and click graphs. In particular we study the usefulness of this new graph for ranking, proving it to be more robust than the hyperlink and click graphs independently, when faced with noisy data, such as spam.
- In Chapter 7 we discuss privacy issues related to query log publication. Specifically we present a new privacy concern in this area, that of business confidentiality. This work provides a new perspective into the query log privacy preservation field.

In retrospective, the main challenge in this work was related to the high amount of Web data processed for each experimental evaluation related to the volume of query log data and Web documents. Also, the lack of non-proprietary logs proved to be an issue when looking for a public data source to complement the experimental evaluation and improve their general repeatability. Even though the Yahoo! data was available for this thesis, this data is not available for public use, due to privacy concerns.

Although in all cases the experimental evaluation was carefully designed and performed, it would have always been ideal produce comparisons of our work with all of the existing state of the art for each topic. Indeed, experiments would have benefited from more long term evaluations. Nonetheless, given the time and processing constraints, the evaluations performed were within the best possible for these settings, and allowed to clearly view the benefits of each approach, Though in the future many of the evaluations can be extended to prove the usefulness of the models presented to different applications. For example, the query-sets model presented in Chapter 3 can also be studied for classification, and the Web graph presented in Chapter 6 has a great deal of potential in Web mining and query ranking.

Bibliography

[1] AOL research website, no longer online. http://research.aol.com.

[2] Google analytics. `http://analytics.google.com/`.

[3] Dmoz open directory project. `http://www.dmoz.org`.

[4] Yahoo! directory. `http://dir.yahoo.com`.

[5] Yahoo! `http://www.yahoo.com`.

[6] Web characterization activity. `http://www.w3.org/WCA/`.

[7] Broadvision. `http://www.broadvision.com`.

[8] Netperceptions. `http://www.netperceptions.com`.

[9] The webalizer. `http://www.webalizer.org`.

[10] Webtrends log analyzer. `http://www.webtrends.com`.

[11] Eytan Adar. User 4xxxxx9: Anonymizing query logs. In *Query Log Analysis: Social and Technological Challenges, Workshop in WWW '07*, 2007.

[12] Reka Albert, Hawoong Jeong, and Albert-Laszlo Barabasi. Error and attack tolerance of complex networks. *Nature*, 406(6794):378–382, July 2000.

[13] Michael Arrington. AOL proudly releases massive amounts of private data. http://www.techcrunch.com/2006/08/06/aol-proudly-releases-massive-amounts-of-user-search-data/, 2006.

[14] Ricardo Baeza-Yates. Mining the web (in spanish). *El profesional de la información (The Information Professional)*, 13(1):4–10, Jan-Feb 2004.

[15] Ricardo Baeza-Yates. Web usage mining in search engines. In *Web Mining: Applications and Techniques, Anthony Scime, editor.* Idea Group, 2004.

[16] Ricardo Baeza-Yates. Graphs from search engine queries. *SOFSEM 2007: Theory and Practice of Computer Science*, pages 1–8, 2007.

[17] Ricardo Baeza-Yates and Berthier A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

[18] Ricardo Baeza-Yates and Alessandro Tiberi. Extracting semantic relations from query logs. In *To appear in ACM SIGKDD international conference on Knowledge discovery and data mining*, 2007.

[19] Ricardo A. Baeza-Yates. Applications of web query mining. In David E. Losada and Juan M. Fernández-Luna, editors, *ECIR*, volume 3408 of *Lecture Notes in Computer Science*, pages 7–22. Springer, 2005.

[20] Ricardo A. Baeza-Yates, Carlos A. Hurtado, and Marcelo Mendoza. Query clustering for boosting web page ranking. In Jesús Favela, Ernestina Menasalvas Ruiz, and Edgar Chávez, editors, *AWIC*, volume 3034 of *Lecture Notes in Computer Science*, pages 164–175. Springer, 2004.

[21] Ricardo A. Baeza-Yates, Carlos A. Hurtado, and Marcelo Mendoza. Query recommendation using query logs in search engines. In *EDBT Workshops*, pages 588–596, 2004.

[22] Ricardo A. Baeza-Yates, Carlos A. Hurtado, and Marcelo Mendoza. Improving search engines by query clustering. *JASIST*, 58(12):1793–1804, October 2007.

[23] Michael Barbaro and Tom Zeller. A face is exposed for AOL searcher no. 4417749. New York Times, 2006.

[24] Luca Becchetti, Carlos Castillo, Debora Donato, Stefano Leonardi, and Ricardo Baeza-Yates. Link-based characterization and detection of Web Spam. In *Second International Workshop on Adversarial Information Retrieval on the Web (AIR-Web)*, Seattle, USA, August 2006.

[25] Doug Beeferman and Adam Berger. Agglomerative clustering of a search engine query log. pages 407–416, 2000.

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

[27] Bettina Berendt. Using site semantics to analyze, visualize, and support navigation. *Data Min. Knowl. Discov.*, 6(1):37–59, 2002.

[28] Bettina Berendt and Myra Spiliopoulou. Analysis of navigation behaviour in web sites integrating multiple information systems. In *VLDB Journal, Vol. 9, No. 1 (special issue on "Databases and the Web")*, pages 56–75, 2000.

[29] Krishna Bharat, Bay-Wei Chang, Monika Rauch Henzinger, and Matthias Ruhl. Who links to whom: Mining linkage between web sites. In *ICDM '01: Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 51–58, Washington, DC, USA, 2001. IEEE Computer Society.

[30] Paolo Boldi and Sebastiano Vigna. fastutil: Fast & compact type-specific collections for java. `http://fastutil.dsi.unimi.it/`.

[31] Paolo Boldi and Sebastiano Vigna. Webgraph. `http://webgraph.dsi.unimi.it/`.

[32] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117, 1998.

[33] Andrei Broder. Identifying and filtering near-duplicate documents. 1848/2000:1–10, 2000.

[34] Andrei Broder. A taxonomy of web search. *SIGIR Forum*, 36(2):3–10, 2002.

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

[36] Chris Clifton, Murat Kantarcioglu, and Jaideep Vaidya. Defining privacy for data mining. In *Proceedings of the National Science Foundation Workshop on Next Generation Data Mining*, Baltimore, 2002.

[37] Robert Cooley, Bamshad Mobasher, and Jaideep Srivastava. Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems*, 1(1):5–32, 1999.

[38] Robert Cooley, Jaideep Srivastava, and Bamshad Mobasher. Web mining: Information and pattern discovery on the world wide web. In *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'97)*, pages 558–567, 1997.

[39] Robert Cooley, Pang-Ning Tan, and Jaideep Srivastava. Discovery of interesting usage patterns from web data. In *WEBKDD*, pages 163–182, 1999.

[40] Robert Cooley, Pang-Ning Tan, and Jaideep Srivastava. Websift: the web site information filter system. In *KDD Workshop on Web Mining, San Diego, CA. Springer-Verlag, in press*, 1999.

[41] Nick Craswell and Martin Szummer. Random walks on the click graph. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 239–246, New York, NY, USA, 2007. ACM Press.

[42] Nick Craswell, Onno Zoeter, Michael Taylor, and Bill Ramsey. An experimental comparison of click position-bias models. In *WSDM '08: Proceedings of the international conference on Web search and web data mining*, pages 87–94, New York, NY, USA, 2008. ACM.

[43] Brian D. Davison. Topical locality in the web. In *Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval*, pages 272–279, Athens, Greece, 2000. ACM Press.

[44] Brian D. Davison, David G. Deschenes, and David B. Lewanda. Finding relevant website queries. In *Poster Proceedings of the Twelfth International World Wide Web Conference*, Budapest, Hungary, May 2003.

[45] Mukund Deshpande and George Karypis. Selective markov models for predicting web page accesses. *ACM Trans. Inter. Tech.*, 4(2):163–184, 2004.

[46] Georges Dupret, Vanessa Murdock, and Benjamin Piwowarski. Web search engine evaluation using clickthrough data and a user model. In *WWW2007 workshop Query Log Analysis: Social and Technological Challenges*, 2007.

[47] Magdalini Eirinaki, Charalampos Lampos, Stratos Paulakis, and Michalis 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.

[48] Magdalini Eirinaki and Michalis Vazirgiannis. Web mining for web personalization. *ACM Trans. Inter. Tech.*, 3(1):1–27, 2003.

[49] Martin Ester, Hans-Peter Kriegel, and Matthias Schubert. Web site mining: a new way to spot competitors, customers and suppliers in the world wide web. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 249–258, New York, NY, USA, 2002. ACM.

[50] Dennis Fetterly. Adversarial information retrieval: The manipulation of web content. *ACM Computing Reviews*, July 2007.

[51] Josef Fink, Alfred Kobsa, and Andreas Nill. User-oriented adaptivity and adaptability in the avanti project. In *Proceedings of Designing for the Web: Empirical Studies (Redmond, WA, 1996), Microsoft Usability Group*, 1996.

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

[53] Zoltán Gyöngyi and Hector Garcia-Molina. Spam: It's not just for inboxes anymore. *IEEE Computer Magazine*, 38(10):28–34, 2005.

[54] Khaled Hammouda and Mohamed 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.

[55] Zhexue Huang, Joe Ng, David Cheung, Michael Ng, and Wai ki Ching. A cube model for web access sessions and cluster analysis. In *Proc. of WEBKDD 2001 (San Francisco CA, August 2001), 47-57.*, 2001.

[56] Glen Jeh and Jennifer Widom. Simrank: a measure of structural-context similarity. In *KDD '02: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 538–543, New York, NY, USA, 2002. ACM Press.

[57] Thorsten Joachims, Dayne Freitag, and Tom M. Mitchell. Web watcher: A tour guide for the world wide web. In *IJCAI (1)*, pages 770–777, 1997.

[58] Rosie Jones, Ravi Kumar, Bo Pang, and Andrew Tomkins. ”i know what you did last summer”: query logs and user privacy. In *CIKM ’07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 909–914, New York, NY, USA, 2007. ACM.

[59] In-Ho Kang and GilChang 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.

[60] George 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.

[61] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

[62] William Kruskal and Leo Goodman. Measures of association for cross classifications. *Journal of the American Statistical Association*, 49, 1954.

[63] Ravi Kumar, Jasmine Novak, Bo Pang, and Andrew Tomkins. On anonymizing query logs via token-based hashing. In *WWW ’07: Proceedings of the 16th international conference on World Wide Web*, pages 629–638, New York, NY, USA, 2007. ACM Press.

[64] Maxim Lifantsev. Voting model for ranking Web pages. In Peter Graham and Muthucumaru Maheswaran, editors, *Proceedings of the International Conference on Internet Computing*, pages 143–148, Las Vegas, Nevada, USA, June 2000. CSREA Press.

[65] James MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability - Vol. 1*, pages 281–297. University of California Press, Berkeley, CA, USA, 1967.

[66] Florent Masseglia, Pascal Poncelet, and Maguelonne 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.

[67] Bamshad Mobasher. Web usage mining and personalization. In Munindar P. Singh, editor, *[88]*. Chapman Hall & CRC Press, Baton Rouge, 2004.

[68] Bamshad Mobasher, Robert Cooley, and Jaideep Srivastava. Automatic personalization based on web usage mining. *Commun. ACM*, 43(8):142–151, 2000.

[69] Bamshad Mobasher, Honghua Dai, Tao Luo, and Miki Nakagawa. Effective personalization based on association rule discovery from web usage data. In *WIDM*

'01: Proceedings of the 3rd international workshop on Web information and data management, pages 9–15, New York, NY, USA, 2001. ACM Press.

[70] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[71] Olfa Nasraoui, Hichem Frigui, Raghu Krishnapuram, and Anupam Joshi. Extracting web user profiles using relational competitive fuzzy clustering. *International Journal on Artificial Intelligence Tools*, 9(4):509–526, 2000.

[72] Olfa Nasraoui, Raghu Krishnapuram, and Anupam Joshi. Relational clustering based on a new robust estimator with application to web mining. In *Proceedings of NAFIPS 99, (New York)*, pages 705– 709, 1999.

[73] Jian Pei, Jiawei Han, Behzad Mortazavi-asl, and Hua Zhu. Mining access patterns efficiently from web logs. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 396–407, 2000.

[74] Mike Perkowitz and Oren Etzioni. Adaptive sites: Automatically learning from user access patterns. Technical Report TR-97-03-01, 1997.

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

[76] Peter Pirolli. Computational models of information scent-following in a very large browsable text collection. In *CHI*, pages 3–10, 1997.

[77] James E. Pitkow and Peter Pirolli. Mining longest repeating subsequences to predict world wide web surfing. In *USENIX Symposium on Internet Technologies and Systems*, 1999.

[78] Barbara Poblete. A web mining model and tool centered in queries. M.sc. in Computer Science, CS Dept., Univ. of Chile, 2004.

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

[80] Diego Puppin, Fabrizio Silvestri, and Domenico 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.

[81] Filip Radlinski. Addressing malicious noise in clickthrough data. In *Learning to Rank for Information Retrieval Workshop at SIGIR 2007*, 2007.

[82] Filip Radlinski and Thorsten 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.

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

[84] Masakazu Seno and George 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.

[85] Dou Shen, Jian-Tao Sun, Qiang Yang, and Zheng 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.

[86] Narayanan Shivakumar and Hector Garcia-Molina. Finding near-replicas of documents and servers on the web. In *WebDB '98: Selected papers from the International Workshop on The World Wide Web and Databases*, pages 204–212, London, UK, 1999. Springer-Verlag.

[87] Ahu Sieg, Bamshad Mobasher, Steve Lytinen, and Robin Burke. Using concept hierarchies to enhance user queries in web-based information retrieval. In *IASTED International Conference on Artificial Intelligence and Applications*, 2004.

[88] Munindar P. Singh, editor. *Practical Handbook of Internet Computing*. Chapman Hall & CRC Press, Baton Rouge, 2004.

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

[90] Myra Spiliopoulou and Lukas C. Faulstich. WUM: a Web Utilization Miner. In *Workshop on the Web and Data Bases (WebDB98)*, pages 109–115, 1998.

[91] Jaideep Srivastava, Robert Cooley, Mukund Deshpande, and Pang-Ning Tan. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1(2):12–23, 2000.

[92] Michael Steinbach, George Karypis, and Vipin Kumar. A comparison of document clustering techniques. *In Proceedings of Workshop on Text Mining, 6th ACM SIGKDD International Conference on Data Mining (KDD'00)*, pages 109–110, August 20–23 2000.

[93] Latanya Sweeney. k-anonymity: a model for protecting privacy. In *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, pages 557–570, 2002.

[94] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining, (First Edition)*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2005.

[95] Paolo Tonella, Filippo Ricca, Emanuele Pianta, and Christian Girardi. Using keyword extraction for Web site clustering. *Web Site Evolution, 2003. Theme: Architecture. Proceedings. Fifth IEEE International Workshop on*, pages 41–48, 2003.

[96] Richard G. Vedder, Michael T. Vanecek, C. Stephen Guynes, and James J. Cappel. Ceo and cio perspectives on competitive intelligence. *Commun. ACM*, 42(8):108–116, 1999.

[97] Vassilios Verykios, Elisa Bertino, Igor Nai Fovino, Loredana Parasiliti Provenza, Yucel Saygin, and Yannis Theodoridis. State-of-the-art in privacy preserving data mining. *SIGMOD Record*, 33(1):50–57, 2004.

[98] Emmanouil Vozalis, A. Nicolaou, and Konstantinos G. Margaritis. Intelligent techniques for web applications: review and educational application. In *Proceedings of the 5th Hellenic-European Conference on Computer Mathematics and its Applications (HERCMA-01), Athens, Hellas, September 20-22, 2001*, pages 312–317, 2001.

[99] Xuanhui Wang and ChengXiang 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.

[100] Yong Wang and Julia E. Hodges. Document clustering using compound words. In *IC-AI*, pages 307–313, 2005.

[101] Wensi Xi, Benyu Zhang, Zheng Chen, Yizhou Lu, Shuicheng Yan, Wei-Ying Ma, and Edward Allan Fox. Link fusion: a unified link analysis framework for multi-type interrelated data objects. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 319–327, New York, NY, USA, 2004. ACM.

[102] Gui-Rong Xue, Hua-Jun Zeng, Zheng Chen, Wei-Ying Ma, and Chao-Jun 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.

[103] Michael H. Zack. Developing a knowledge strategy. *California Management Review*, 41:125–145, 1999.

[104] Osmar R. Zaïane, Man Xin, and Jiawei Han. Discovering web access patterns and trends by applying OLAP and data mining technology on web logs. In *Advances in Digital Libraries*, pages 19–29, 1998.

[105] Alessandro Zanasi. Competitive intelligence through data mining public sources. *Competitive Intelligence Review*, 9(1):44–54, 1998.

[106] Ying Zhao and George Karypis. Criterion functions for document clustering: Experiments and analysis. Technical report, University of Minnesota, Department of Computer Science / Army HPC Research Center, Minneapolis, MN 55455, 2001.

[107] Jianhan Zhu, Jun Hong, and John 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.