Thierson Couto Rosa

Orientador - Nivio Ziviani

Co-orientador - Edleno Silva de Moura

Uso de Apontadores na Classificação de Documentos em Coleções Digitais

Tese de doutorado apresentada ao Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Minas Gerais, como requisito parcial para obtenção do título de doutor em Ciência da Computação.

Belo Horizonte

Dezembro de 2007

Abstract

In this work, we show how information derived from links among Web documents can be used in the solutions of the problem of document classification. The most obvious form of link between two Web documents is a hyperlink connecting them. But links can also be derived from references among documents of digital collections hosted in the Web, for instance, from citations among articles of digital libraries and encyclopedias.

Specifically, we study how the use of measures derived from link information, named bibliometric measures can improve the accuracy of classification systems. As bibliometric measures, we used co-citation, bibliographic coupling and Amsler. We obtained distinct classifiers by applying bibliometric and text-based measures to the traditional k-nearest neighbors (kNN) and Support Vector Machine (SVM) classification methods.

Bibliometric measures were shown to be effective for document classification whenever some characteristics of link distribution is present in the collection. Most of the documents where the classifier based on bibliometric measures failed were shown to be difficult ones even for human classification.

We also propose a new alternative way of combining results of bibliometric-measure based classifiers and text based classifiers. In the experiments performed with three distinct collections, the combination approach adopted achieved results better than the results of each classifier in isolation.

Resumo

Este trabalho mostra como informações derivadas de apontadores entre documentos da Web podem ser utilizadas na solução do problema de classificação de documentos. A forma mais comum de apontadores entre documentos da Web corresponde aos *hyperlinks* entre documentos. Entretanto, apontadores também podem ser derivados a partir de referências entre documentos de coleções digitais hospedadas na Web, por exemplo, a partir de referências entre artigos de bibliotecas digitais ou de enciclopédias.

Especificamente, investigamos como a utilização de medidas derivadas de informação de apontadores, denominadas *medidas bibliométricas*, podem ser utilizadas para melhorar a qualidade de sistemas de classificação de documentos. As medidas bibliometricas utilizadas foram: co-citação, acoplamento bibliográfico e Amsler. Obtivemos classificadores com estas medidas e classificadores com informações de texto, utilizando os seguintes métodos de classificação: o método dos k vizinhos mais próximos (kNN) e o método *Suport Vector Machine* (SVM).

Classificadores com medidas bibliométricas mostraram ser eficazes sempre que a distribuição de apontadores na coleção possui determinadas características. Além disto, os documentos para os quais classificadores baseados nestas medidas falham mostraram-se difíceis também na classificação feita por pessoas.

Propomos, ainda, um modo alternativo de combinar resultados de classificadores que usam medidas bibliométricas com resultados de classificadores que usam informações de texto. Experimentos mostram que a combinação de resultados é superior ao resultados individuais em todas as coleções de teste.

Written Papers

1. Thierson Couto, Marco Cristo, Marcos A. Gonçalves, Pável Calado, Nivio Ziviani, Edleno Moura and Berthier Ribeiro-Neto. A Comparative Study of Citations and Links in Document Classification. *ACM Joint Conference on Digital Libraries*, p.75-84, Chapel Hill, NC, USA, June 11-15, 2006.

2. Klessius Berlt, Edleno Silva de Moura, André Carvalho, Marco Antônio Cristo, Nivio Ziviani, Thierson Couto. Hypergraph Model For Computing Page Reputation on Web Collections. *SBBD Simpósio Brasileiro de Banco de Dados*, p.35-49, João Pessoa, PB, Brazil, October 5-19, 2007 (elected best paper for the symposium).

3. Fernando Mourão, Leonardo Rocha, Renata Araújo, Thierson Couto, Marcos Gonçalves and Wagner Meira. Characterizing and Understanding the Impact of Temporal Evolution on Document Classification. *First ACM International Conference on Web Search and Data Mining - WSDM 2008* (to appear).

4. T. Couto, N. Ziviani, P. Calado, M. Cristo, M. Gonçalves, E. S. de Moura and W. Brandão. Classifying Web Documents with Bibliometric Measures. *Information Retrieval Journal* (submited).

5. Klessius Berlt, Edleno Silva de Moura, André Carvalho, Marco Antônio Cristo, Nivio Ziviani, Thierson Couto. Hypergraph Model For Computing Page Reputation on Web Collections. *Information Systems Journal* (submited).

Thierson Couto Rosa

Advisor - Nivio Ziviani
Co-Advisor - Edleno Silva de Moura

The Use of Links for Classifying Documents in Digital Collections

Thesis submited to the Computer Science Graduate Program of the Federal University of Minas Gerais in fulfillment of the thesis requirement to obtain the degree of doctor in Computer Science.

Belo Horizonte
December, 2007

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The appearance and expansion of the World Wide Web have brought great challenges and opportunities to the Information Retrieval area. Among the many challenges, one have attracted the attention of many researchers, the task of automatic classifying Web documents. However, the Web also presents new opportunities for IR researchers that were not available in traditional collections of digital documents. For example, links among documents are important additional sources of information that have been used both for the ranking and for the classification tasks. Link analysis is an Information Retrieval technique that aims to extract information from the link structure of the Web. This work is related to research in link analysis with the aim of improving the classification of documents. In this chapter, we discuss the goals and the contributions of our work.

1.1 Information Retrieval Systems

Information Retrieval (IR) systems deal with the problems of collecting, representing, organizing and retrieving documents. The main task of an IR system is to retrieve documents that are relevant to a query formulated by a user. The part of the system responsible for this task is named *search engine*.

When a query is received, it is parsed by the search engine and its component words are extracted. The extracted terms are matched against the *index* that is an internal data structure that maps terms to documents where they occur. Each matched document is assigned a value of its relevance to the query, according to an internal model of relevance, named IR *model*. Finally, the matched documents are ranked by some relevance value and the final ranking is presented to the user.

Another important task of IR is document classification, the activity of assigning predefined category labels to unlabeled documents. Research on document classification dates back to the early 60s. However, until the 80s the most popular approach to document classification was the *knowledge engineering* one, consisting in manually defining a set of rules encoding expert knowledge on how to classify documents under given categories. Knowledge engineering is a semi-automatic process that demands a *knowledge engineer* to code the rules and an *expert* to classify documents into a specific set of categories. Thus, it is an expensive and laborious process. Also, this approach is not flexible, since even minor changes on the categories require two professionals again to adapt the classifier [48].

Before the 90s, IR systems were developed for specific collections of documents like scientific articles, electronic newspapers, business articles and encyclopedias. The great majority of the indexed documents followed editorial restrictions that imposed some form of control over the format and the content of documents. Although some IR systems were developed to allow users to access many different collections, each collection followed some patterns and was concentrated in a specific subject area. Another important characteristic was that users of these IR systems usually had some training in how to formulate queries using system operators, which allowed for better expression of his or her information need. In a so organized specialized context, the task of ranking documents did not bring much challenges and most of the classification tasks were executed manually by an expert.

The IR area has grown up in the context just described. However, at the beginning of the 90s the World Wide Web was introduced and brought important changes on the needs for the tasks of ranking and classifying documents. The new demands and also new sources of information introduced with the Web changed the scenario for IR completely. In parallel, advances in IR techniques and in machine learning contributed to augment enormously the research on IR for the Web.

1.2 New IR Requirements for the Web

The Web is characterized as a medium that allows for cheap and easy publishing of multimedia documents. In contrast to collections of traditional IR systems, there is no editorial restriction or control about the format and content of documents to be published. These conditions have led to the rapid emergence of the Web as a repository of documents that is huge, diverse in format and content, and very dynamic, with some documents being removed, others being updated and many being added constantly.

The diversity of content and great dynamics in the Web have imposed urgent demand on new ranking techniques, but also, have caused renewed research interest in the task classifying documents automatically into categories according to their topics in order to assist the user with finding information [16, 53].

One important application of document classification is directory maintenance. Web directories (e.g. Yahoo, Open Directory and Google Directory) have been created and maintained with the intent to organize Web pages into hierarchical topics. A Web directory is an important access tool for two reasons. First, it allows for a search focused in some specific topic. Second, it allows for browsing the directory hierarchy.

Expansion of directories with new URLs, however, has been done manually. This expensive and inefficient approach is not able to keep directories up-to-date with the creation of new pages. Thus, the automatic classification of Web pages into topics is essential for directory expansion.

Automatic classification is also very important in other contexts inside the Web. Many digital collections of documents that already existed before the Web have migrated to the Web environment (specially, collections of scientific papers and encyclopedias). The majority of these collections are organized by topics. Automatic classification is useful to classify new incoming documents into the proper topics.

Research in automatic classification has been an intensive research topic since the early 90s, when researches adopted machine learning techniques to develop automatic classifiers. In this approach, a general inductive process, also called *learner*, automatically builds a classifier for a given category c by observing a set of documents manually classified under c [48].

Despite bringing new challenges to research in IR, the Web also offers new evidence. The tags of the HTML text inside Web pages reveal some structures in the page like headings, tables, and items of lists that can be useful to accentuate the importance of some terms inside a page.

Another important source of evidence found in many Web documents are the links between documents. These links may be directly derived from the hyperlinks among pages, or from citations between documents of digital libraries or between articles of digital encyclopedias hosted in the Web. Hyperlink derived information has been shown useful for ranking [8, 10, 34] and for classifying [13, 39] Web pages. In this work we investigate further how we can use link information for enhancing the task of classifying documents.

1.3 Link Analysis in IR

Citation is a form of linkage between documents that is as old as written language. There are many reasons a written work may cite another, but a citation from a document A to another document B reveals two facts: 1) The author of document A states that document B is somehow related to document A or at least to the part of document A where the citation occurs. 2) The author of document A considers that document B has some importance, because B was chosen to be cited.

Bibliometrics [22] is a research area that is concerned to the bibliographic citation graph of academic papers and has explored the two facts just cited, for two applications: (1) the finding of scientific papers on related topics and (2) the measure of the relative importance of scientific papers in the citation graph. Both applications are also very important in the Web context. For instance, solutions to the problem of finding pages related to a topic are useful for automatic classification of Web documents. They can be used for the automatic expansion of directories [20], as well as for expansion of categorized collections of linked documents hosted in the Web, like digital libraries and encyclopedias, among others.

The measure of relative importance of Web pages, on the other hand, has been used to enhance Web search engine ranking. Traditional IR models based on text only are not sufficient for ranking Web pages due to the large number of documents containing the query terms. A common approach is to combine the text-based ranking with an importance or "reputation"-based ranking of pages derived only from hyperlink information. Many ideas from Bibliometrics have influenced algorithms that assign hyperlink-based reputation values to Web pages [8]. The ranking of pages by their reputation values have became an intensive research area in IR [6, 28, 34, 41] and the combined ranking has been shown to be better than the ranking based only on text [4, 10, 14].

However, hyperlinks and citations are two different document connections. The concept of hyperlink extends that of citation by allowing the reader of a pointing document to have direct access to the pointed document. So it is possible to use hyperlinks with navigational purpose only. Also, the environments where scientific citations and Web hyperlinks occur are distinct. Scientific papers are peer-reviewed, thus ensuring the referencing of other important papers on the same subject. On the other hand, Web pages may reference other unimportant and unrelated pages and, for commercial reasons, two important related pages may not link to each other.

Link analysis algorithms are then faced with two problems: 1) How to obtain, from the

noisy Web link structure, information that is useful for some of the IR tasks. 2) How to define methods to make appropriate use of the obtained information in order to enhance execution of a given IR task.

1.4 Objectives and Contributions

This work concerns the particular case of using link information available in distinct kinds of document collections, hosted in the Web, to improve document classification. As link information, we use bibliometric similarity measures which allow for evaluating how related two documents are, considering the links they have. The main objectives of this work are:

- Investigate how effective classifiers based on bibliometric similarity measures are to classify documents in distinct collections.
- Investigate strategies for combining the results of classifiers based on bibliometric measures and text information, in order to obtain a final, more effective classification.
- Analyze, in each collection, the documents that the classifiers using bibliometric measures did not classify correctly.

As a contribution towards the above objectives, we present a comparative study of the use of bibliometric similarity measures for classifying documents. We refer to these classifiers as *bibliometric classifiers*. Three different link-based bibliometric measures were used: co-citation, bibliographic coupling and Amsler. They were derived from hyperlinks between Web pages and citations between documents. These bibliometric measures were combined with two content-based classifiers: k-nearest neighbors (kNN) and Support Vector Machines (SVM).

In our comparative study we run a series of experiments using a digital library of computer science papers, a Web directory and an on-line encyclopedia. Results show that both hyperlink and citation information, when properly distributed over the documents and classes, can be used to train reliable and effective classifiers based on the kNN classification method. By reliable we mean that when the classifier assigns a class to a document with high probability, the class is the correct one most of the time. Conversely, if the classifier assigns a class to a document with low probability, the class is generally incorrect most of the time. By effective we mean that experiments performed with ten-fold cross validation have reached values of macro and micro-average $F1$ superior to state-of-the-art text-based

classifiers in two of the collections studied and, in the sub-collection of an encyclopedia, the micro-average $F1$ value is only marginally distinct from the one obtained with a text-based classifier trained using the SVM model.

As another contribution, we investigate the possible reasons for the failures of bibliometric classifiers. We suspected that these cases are hard even for humans, since the test documents might have some kind of strong relation to documents of the wrong class. In order to confirm this hypothesis we conducted a user study, asking volunteers to classify a random sample of these supposedly difficult cases. The experiment shows that most cases are in fact difficult and that there is little consensus among human classifiers regarding the correct class of a same document. Also, there are test documents for which the second most probable class assigned by the classifier was the class assigned by specialists. Thus, these cases could be considered multilabel classification cases according to the taxonomy.

In summary, the experiments with the three collections have shown that the use of bibliometric measures perform well for classifying Web collections and digital library collections where most of documents form pairs that have links in common to other documents. We present empirical evidences that (i) the number of in-links and out-links is important to train bibliometric classifiers and (ii) the co-occurrence of in-links and out-links is important to determine the existence of a bibliometric relation between documents. We also study alternative ways of combining classifiers based on links with text-based classifiers, performing an analysis of the gains obtained by each alternative combination studied. We present comparisons to the effectiveness of an ideal perfect combiner and show that the gains obtained with combination are important even when they are small, since in these cases even a perfect combiner could not perform much better.

1.5 Related Work

In this section, we review previous work about links among documents. Links were first studied as a source of information in Bibliometrics where they corresponded to the citations among scientific papers. Citations were used mainly to find articles related to some topic and to find important articles in a topic. Works addressing these two purposes contributed and inspired a number of work in link analysis for hypertext collections. Thus, for chronological reasons, in Section 1.5.1 we first review some work in bibliometrics. In Section 1.5.2, we review previous works that use link information in classification of linked documents and in Section 1.5.3 we review works that make use of links for assigning reputation values

to Web pages.

1.5.1 Citation-Related Measures

In 1963, Kessler [33] presented the bibliographic coupling measure that measures the similarity of two documents by counting the number of documents they cite in common (see Section 2.3.2). Small et al. [51] used the bibliometric coupling for clustering scientific journals.

About ten years later, the measure of co-citation was introduced in [50] (see Section 2.3.1). Co-citation between two documents A and B is a measure that counts the number of documents that cite both A and B. Co-citation and bibliographic coupling have been used as complementary sources of information for document retrieval and classification [3,5]. Amsler [3] introduced another similarity measure that combines and extends co-citation and bibliographic coupling (see Section 2.3.3).

Citations also were suggested as a means to evaluate the importance of scientific journals [26], where the importance of a journal was assumed proportional to the number of citations to its papers. In [45], Salton introduced the idea of using citations for automatic document retrieval. Pinski and Narin [25] proposed a citation-based measure of reputation, stemming from the observation that not all citations are equally important. They introduced a recursive notion of reputation such that a scientific journal has high reputation if it is heavily cited by other reputable journals. Geller [27] further extended the work of Pinski and Narin and observed that the values of reputation correspond to the stationary distribution of the following random process: beginning with an arbitrary journal j, one chooses a random reference that appears in j and moves to the journal specified in the reference. Geller also demonstrated that the values of the reputations converge. Doreian [40] proposed a recursive algorithm that is able to approximate the convergence values of reputation as proposed by Geller.

1.5.2 Document Classification

The Companion algorithm [21] was proposed to find pages in the same topic of a page given as input. Companion constructs a vicinity graph for a given input page and applies the HITS algorithm over the generated graph. The algorithm ranks pages by authority degree and uses the top pages as the most similar to the input page.

The authors in [30] used a similarity matrix between Web pages resulting from the

combination of co-citation between pages and a text-based similarity between pages. The matrix was used for clustering of Web pages.

For the task of classification in the Web, the authors in [20] compared classification using only link information with classification using only textual information over a directory of Web pages. They trained kNN classifiers using as similarity measures co-citation, bibliographic coupling, Amsler and hub and authority values derived from the Companion algorithm. As text-based similarity they used the cosine measure. Their experiments have shown superior effectiveness of the link-based classifiers.

In [23], authors used link information with a probabilistic latent indexing and probabilistic HITS classifiers and conclude that whenever there is sufficient high link density and good link quality, link information is useful.

Several works have used some kind of combination of link information with other hypertext sources of evidence in order to improve classification effectiveness. For instance, Joachins et al. [32] studied the combination of support vector machine kernel functions representing co-citation and text-based information.

Cohn et al. [18] used a probabilistic combination of link-based and text-based probabilistic methods and shown that the combination presented better effectiveness than the text-based classifier in isolation.

Pavel et al. [9] extended the work in [20] using other classification methods and used a Bayesian network to combine the output of link-based classifiers with the output of text-based classifiers. Their experiments over a directory of Web pages have shown that the combination of results presented better effectiveness than any of the two classifiers in isolation. However, the gains of combination over the text-based classifier were much more significant than the gains over the link-based classifiers.

Some authors have not used links directly in the classification, but only the textual evidence related to linked documents. For instance, in [24], [29] and [52], good results were achieved by using anchor text together with the paragraphs and headlines that surround the links. Yang et al. [61] show that the use of terms from linked documents works better when neighboring documents are all in the same class.

As another approach, some works classify a test document using its textual features combined with the class information of its neighbors (documents that link to it or are linked to by it). The neighbors may be a classified document or other test documents. Chakrabarti [12] presented a two-step classification method named HyperClass. In the first step, HyperClass constructs the neighborhood for a test document t and assigns an

initial most probable class to each test document of the neighborhood (including t) by means of a traditional text-based classifier. The second step is a recursive one in which the most probable class of t is computed as a conditional probability given the text evidence of t and the classes of the neighbors of t. The second step is applied to all test documents and it is shown, using Markov Random Fields, that the values of the probabilities converge. Oh et al. [39] improved on this work by using a filtering process to further refine the set of linked documents to be used.

In this work we present an empirical comparative study of the bibliometric similarity measures used in [9], when applied to distinct collections of digital documents. Also, we analyze the characteristics of collections that have influence on the results of classifiers based on bibliometric measures and present a method for combining the results of text-based and bibliometric classifiers. Finally, we present a user study that confirm that most documents that lead bibliometric classifiers to fail are indeed difficult to classify even by people.

1.5.3 Computing Reputations of Web Pages

The simplest idea for ranking documents using link information is to rank the documents by the number of in-links each document has. This solution is also known as the Indegree algorithm [7]. The intuition behind the algorithm is that pages with more incoming links have more visibility, and thus may have high reputation in terms of quality.

In [11], the authors proposed the use of page connectivity for searching and visualizing the Web. For the experiments, they used a different set of documents for each query. This set was built by submitting a query to a search engine and by using the resulting pages to create an initial set of pages I. Then they expanded I with the root URLs of each web site present in I and with all the URLs that point to or are pointed by pages in I. To each page p of the expanded set I they associated a rank value that is the number of incoming and outgoing links of p. The results presented were positive, but only experiments with a few queries were performed. This approach and the Idegree algorithm can be seen as a simple electoral process, since the reputations of web pages are computed by the number of links (votes) given by other web pages.

Later, the ideas used for citations among scientific documents were transposed to the Web environment. Specifically, the recursive notion of reputation and the mathematical foundations used to compute reputations, commented in last section, were used and extended giving rise to PageRank and HITS algorithms.

PageRank [8] is the most well-known link-based ranking algorithm for Web. It is based on the idea that a page is recursively defined as important when it is linked to by many other important pages. PageRank models the process of ranking pages by a random walk in the Web graph, but differently from previous work, it deals with loops in the graph and with disconnected components, by including a dumping factor.

HITS (Hyperlink-Induced Topic Search) algorithm was introduced by Kleinberg in [35]. In HITS, pages assume two distinct roles: hubs and authorities. An authority is a page that contains important information on a given subject. A hub is a page that may not have relevant information, but links to many authority pages. Thus, a good hub page links to many good authority pages and, recursively, a good authority page is linked to by many good hub pages.

Several subsequent works proposed solutions for some of the problems still found on the above algorithms. For instance, in the Web, we frequently find groups of pages highly linked to each other, such as the pages belonging to a same site. In this case, many links do not necessarily indicate higher reputation, what can make HITS classify certain pages as good hubs/authorities when they are not. To avoid this problem, the SALSA algorithm [37] computes the degrees of hub and authority for Web pages by examining random walks through the Web graph.

We can identify two distinct approaches in the literature that are used for computing page reputation. The first approach considers reputation of a page as a measure of its *popularity*. In this case, the reputation of a page depends only on the number of references to it. The Indegree algorithm is a representative of this category. The second approach considers reputation as a measure of *relative authority*. In this case, the reputation value of a page interferes with the reputation value of the pages it links to. PageRank, HITS and the algorithms derived from them are representatives of this approach. Some works have compared these two approaches. Amento et al. [2] presented experiments for a site level version of Indegree, where all the links to a web site were computed as links to its root page. The sites were then ranked and the results obtained slightly outperformed PageRank. Their experiments indicate that simple count of votes may produce good results.

Westerveld et al. [57] presented experiments using Indegree combined with textual information for ranking pages on a search engine, with conclusions that such combination produce good results.

Upstill et al. [55] studied the usefulness of several kinds of evidence on the home page finding task, where two of them are Indegree and PageRank. In all the experiments per-

formed by them, PageRank and Indegree presented extremely close performances. The authors also comment that combining pieces of evidence can hurt the final result and lead to erroneous conclusions.

Borodin et al. [6] study and propose several methods to calculate the reputation of a page. The experiments are performed using the method described in [11] and the Google search engine to create a local database. Results obtained indicate that Indegree is superior to PageRank on the experimented scenario. However, the results are not conclusive about the comparison between Indegree and PageRank, since authors presented no experiments with a complete search engine database.

1.6 Organization of this Work

This text is organized as follows. Chapter 2 introduces basic concepts related to the text and bibliometric information we use, document classification and evaluation of the effectiveness of classifiers. These concepts are essential to the understanding of this work. Chapter 3 describes our approaches to use bibliometric information to derive classifiers and the methods used to combine the results of text-based and bibliometic classifiers. It also present the collections we used to evaluate the classification and combination methods. Chapter 4 presents the results of series of experiments using the classification and combination methods described in Chapter 3, as well as, the results of user a study about the failures of automatic classifications. Finally, in Chapter 5 presents conclusions and future work.

Chapter 2

Basic Concepts

This chapter introduces basic concepts used in subsequent chapters. Section 2.1 describes the Vector Space Model, which is one of the most used model in IR, both for ranking and classifying documents. Section 2.2 presents the graph model for collections of linked documents. In Section 2.3 we present the definitions of bibliometric measures we use. In Section 2.4 we discuss about document classification and present some classification methods. Section 2.5 presents some measures commonly used to evaluate the effectiveness of classifiers, which are useful to understand the results shown in the following chapters. In Section 2.6 we present the Bayesian network, a formalism we use in this work to combine results of classifiers that were trained using distinct source of information, in order to obtain a final enhanced classification.

2.1 The Vector Space Model

The vector space model is a simple, traditional and effective text-based model for retrieving and ranking documents from a collection [47]. Its is also much used in the task of document classification. Documents (and queries) in the vector space model are represented as vectors in a space composed of index terms, i.e., words extracted from the text of the documents in the collection [58]. This vector representation allows us to use any vector algebra operation to compare queries and documents, or to compare a document to another one.

Let $\mathcal{D} = \{d_1, d_2, \ldots, d_N\}$ be a collection of documents. Let $\mathcal{K} = \{k_1, k_2, \ldots k_T\}$ be the set of all distinct terms that appear in documents of $\mathcal{D}$. With every pair (k_i, d_j), $k_i \in \mathcal{K}$, $d_j \in \mathcal{D}$ is associated a weight w_{ij}. A document d_j is, thus, represented as a vector of the term weights $\vec{d_j} = (w_{1j}, w_{2j}, \ldots w_{Tj})$, where T is the total number of distinct terms in the

entire document collection. Each w_{ij} represents the importance of term k_i in document d_j. The computation of w_{ij} we use in this work was proposed in [46] and corresponds to the product:

$$w_{ij} = log_2(1 + tf_{ij}) \times \log_2 \frac{|\mathcal{D}|}{n_i} \tag{2.1}$$

where tf_{ij} is the number of times the term k_i occurs in document d_j, n_i is the number of documents in which k_i occurs, and $|\mathcal{D}|$ is the total number of documents in the collection. The component tf_{ij} is usually referred to as TF (*term frequency*) and reflects the idea that a term is more important in a document if it occurs many times inside the document. The factor $\log(|\mathcal{D}|/n_i)$ is called the *inverse document frequency* (IDF) and measures how rare is the term k_i in the collection $\mathcal{D}$. The entire product w_{ij} is referred to as *term frequency - inverse document frequency* (TF-IDF).

The vector space model is much used model for ranking documents in response to a user query. Users formulate their queries as sets of words. Thus, a query q also can be represented as a vector of term weights $\vec{q} = (w_{1q}, w_{2q}, ..., w_{Tq})$. With this representation, we can use any vector related measure to compare a query with a document. The most commonly used measure is the so called *cosine similarity*, i.e., the cosine value of the angle between both vectors. Thus, we define the similarity between a document d_j and a query q as:

$$cos(\vec{d_j}, \vec{q}) = \frac{\sum_{i=1}^{t} w_{ij} \times w_{iq}}{\sqrt{\sum_{i=1}^{t} w_{ij}^2} \times \sqrt{\sum_{i=1}^{t} w_{iq}^2}} \tag{2.2}$$

Given a query, the vector space model computes a similarity value to each document that has at least one term in common with the query. Thus a set of documents is generated. This set is ordered in decreasing order of similarity to the query, and the resulting ranking is presented to the user. The documents on the top of the raking are the most relevant to the query, according to the vector space model.

Figure 2.1 shows the vectors corresponding to a query q and a document d_j, in space with two dimension, that is, containing two terms k_1 and k_2.

The cosine measure defined in Equation 2.2 can also be used as a similarity measure between documents of a given collection. In this case, we substitute the query q in the equation for a vector representing another document d_k and obtain Equation 2.3. In this work, we use Equation 2.3 to obtain kNN classifiers which make use of similarity values

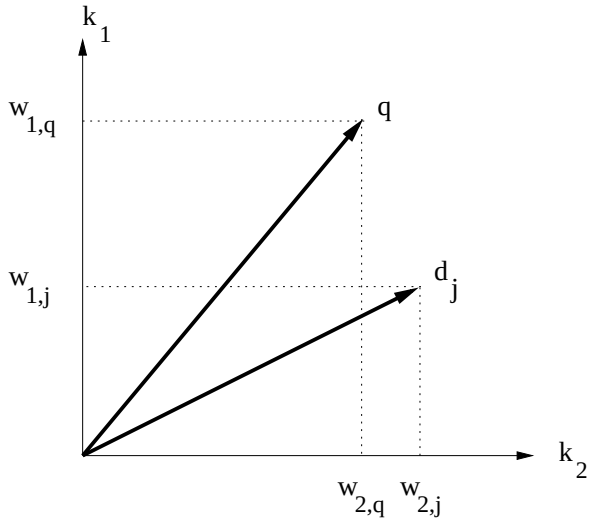


Figure 2.1: Representation of document d_j and query q in the vector space model, considering only two terms k_1 and k_2.

among documents to decide the class of a non-classified document.

$$cos(\vec{d_j}, \vec{d_k}) = \frac{\sum_{i=1}^{t} w_{ij} \times w_{ik}}{\sqrt{\sum_{i=1}^{t} w_{ij}^2} \times \sqrt{\sum_{i=1}^{t} w_{ik}^2}} \tag{2.3}$$

2.2 Graph-Based Model

Collections of documents which have direct linkage between documents can be modeled as a direct graph $\mathcal{G} = (\mathcal{D}, \mathcal{E})$, where the set of vertices $\mathcal{D}$ represents the set of documents and the set $\mathcal{E}$ is the set of direct edges representing the linkages between documents. For example, graphs can be derived from collections of Web pages, digital libraries of scientific papers, encyclopedias, etc.

In this work we use the term *link* to refer generically to the direct edges of the graph derived from a given collection. We also use the terms *pages* or *papers* in place of *vertex* when it is clear from the context that we are referring to a graph derived from a collection of such documents. We define *out-link* of a document (vertex) d as an edge from d to another document. An *In-link* of d is an edge incident to d.

The graph model just described is very important in modern IR, for instance, it is used as input by most algorithms that compute page reputation values [7,8,28,34,34] for ranking documents in response to a user query. The model is also used to compute bibliometric

measures which are presented in Section 2.3.

2.3 Bibliometric Similarity Measures

In Chapter 3, we use three similarity measures derived from link structure to train classifiers: co-citation, bibliographic coupling, and Amsler. These measures were introduced in *Bibliometrics*[1] [22] to measure similarity between scientific documents by means of the citations they have in common. Here, we extend the use of these measures to any collection of documents that can be represented as a directed graph, as described in Section 2.2. Let d be a document of the set $\mathcal{D}$ of documents of the collection. We define the *parents* of d (P_d) as the set formed by all the documents in $\mathcal{D}$ that link to d. We also define the *children* of d (C_d) as the set of all documents d links to. We now describe each link-based similarity measure.

2.3.1 Co-Citation

Co-citation was proposed by Small in [50]. Given two documents d_1 and d_2 of $\mathcal{D}$, co-citation between d_1 and d_2 is defined as:

$$co{-}citation(d_1, d_2) = \frac{|P_{d_1} \cap P_{d_2}|}{|P_{d_1} \cup P_{d_2}|} \tag{2.4}$$

Equation (2.4) indicates that, the more parents d_1 and d_2 have in common, the more related they are. This value is normalized by the total set of parents, so that the co-citation similarity varies between 0 and 1. If both P_{d_1} and P_{d_2} are empty, we define the co-citation similarity as zero.

For example, given the documents and links in Figure 2.2, we have that $P_A = \{D, E, G, H\}$ and $P_B = \{E, F, H\}$, $P_A \cap P_B = \{E, H\}$ and $P_A \cup P_B = \{D, E, F, G, H\}$. Thus $co{-}citation(A, B) = \frac{2}{5}$.

2.3.2 Bibliographic Coupling

Kessler [33] introduced the measure of bibliographic coupling. Bibliographic coupling between two documents $d_1 \in \mathcal{D}$ and $d_2 \in \mathcal{D}$ is defined as:

[1] The study of written documents and their citation structure.

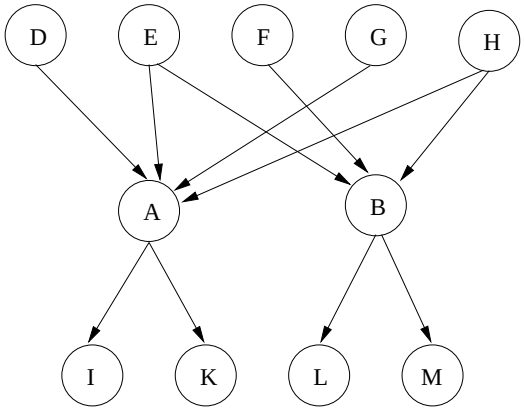


Figure 2.2: Documents A and B with their parents and children.

$$bibcoupling(d_1, d_2) = \frac{|C_{d_1} \cap C_{d_2}|}{|C_{d_1} \cup C_{d_2}|} \tag{2.5}$$

According to Equation (2.5), the more children in common page d_1 has with page d_2, the more related they are. This value is normalized by the total set of children, to fit between 0 and 1. If both C_{d_1} and C_{d_2} are empty, we define the bibliographic coupling similarity as zero.

Consider the example shown in Figure 2.2. Consider documents E and H. $C_E = \{A, B\}$, $C_H = \{A, B\}$. So, by Equation (2.5), $bibcoupling(E, H) = 1$.

2.3.3 Amsler

Amsler [3] proposed a measure of similarity that combines both co-citation and bibliographic coupling, to measure the similarity between two papers. Generalizing the Amsler original idea, we can say that two documents d_1 and d_2 are related if they have at least one document in common among their child or parent documents. Formally, the Amsler similarity measure between two documents d_1 and d_2 is defined as:

$$amsler(d_1, d_2) = \frac{|(P_{d_1} \cup C_{d_1}) \cap (P_{d_2} \cup C_{d_2})|}{|(P_{d_1} \cup C_{d_1}) \cup (P_{d_2} \cup C_{d_2})|} \tag{2.6}$$

Equation (2.6) tells us that the more linked documents (either parents or children) d_1 and d_2 have in common, the more they are related. The measure is normalized by the total number of links. If the set of parents and the set of children of both d_1 and d_2 are empty, the similarity is defined as zero.

Considering Figure 2.2 again, and documents A and B in it, we have that $(P_A \cup C_A) \cap (P_B \cup C_B) = \{E, H\}$, and, $(P_A \cup C_A) \cup (P_B \cup C_B) = \{D, E, F, G, H, I, J, M\}$, Thus, $amsler(A, B) = \frac{2}{8}$.

2.4 Document Classification

Given a collection $\mathcal{D}$ of documents and a set $\mathcal{C}$ of categories or classes, document classification is the task of assigning a boolean value to each pair $(d_j, c_i) \in \mathcal{D} \times \mathcal{C}$. The value T assigned to (d_j, c_i) corresponds to the decision of labeling document d_j with class c_i, whereas the value F indicates that d_j is not to be labeled with class c_i. This process is also referred to as *hard classification* [48] and corresponds to a function $\Phi : \mathcal{D} \times \mathcal{C} \rightarrow \{T,F\}$.

Until the '80s, the most popular approach for the automatic classification of documents consisted in manually building an expert system capable of deciding the class of a set of documents. These systems are built specifically for a collection of classes and involve two kinds of professionals: a *knowledge engineer* and a *domain expert*. The knowledge engineer builds the expert system by manually coding a set of logical rules with the aid of an expert in the membership of documents in the chosen set of classes (the domain expert). One logical rule is created for each class and has the format:

if (*expression*) **then** *class*.

The main drawback of this approach is that it is inflexible. If the set of classes is updated, the two professionals must intervene again. Besides, it is also expensive and time consuming.

Since the early 90s, another paradigm, that of *machine learning*, has gained popularity in research community. The approach consists in the use of a general inductive process (named *learner*) to automatically build an automatic document classifier by learning, from a set of pre-classified documents, the characteristics of the classes of interest [48].

Machine learning approach to document classification has become attractive, mainly because of the great number of applications in the Web which demand for document classification. Among these applications, we can cite classification of documents in intranets of huge companies, expansion of Web directories and classification of new articles in digital libraries.

2.4.1 Training Classifiers Automatically

The machine learning process relies on the following premises: there is an *initial corpus* $\Omega = \{d_1, d_2, \ldots, d_{|\Omega|}\} \subset \mathcal{D}$ of documents pre-classified (maybe by a domain expert) under classes $\mathcal{C} = \{c_1, c_2, \ldots, c_{|\mathcal{C}|}\}$, that is, the values of the some function $\Phi\colon \mathcal{D} \times \mathcal{C} \to \{T, F\}$ are known for every pair $(d_j, c_i) \in \Omega \times \mathcal{C}$.

The learning process consists in deriving a function $\Psi\colon \mathcal{D} \times \mathcal{C} \to \{T, F\}$ named *classifier* such that Φ and Ψ coincide as much as possible. Thus, after a classifier Ψ is obtained by the learning process, it is necessary to evaluate its effectiveness by comparing its results to the values of the Φ function. In order to train the classifier and evaluate it, two disjoint subsets of Ω, not necessarily of equal size, are used:

- The *training set* used to obtain the classifier Ψ. The classifier is trained by learning the characteristics of the documents of the training set that help to identify the classes of these documents.

- The *test set*, used for testing the effectiveness of the obtained classifier Ψ. Each document d_j in the test set is submitted to Φ. The classifier infers the class (or classes) of d_j by matching the characteristics of d_j with the characteristics learned during the training process that most identify the classes in $\mathcal{C}$. Finally, the classifier takes a decision for each pair (d_j, c_i) which is compared to $\Phi(d_j, c_i)$. A measure of classification effectiveness is based on how often the values of $\Psi(d_j, c_i)$ match the values of $\Phi(d_j, c_j)$.

2.4.2 Hard and Ranking Classification

Many classification methods do not output directly a T or F value for each pair (d_j, c_i). Instead, they implement a function $\Gamma : \mathcal{D} \times \mathcal{C} \to [0, 1]$ such that the value $\Gamma(d_j, c_i)$ corresponds to a *confidence score* that document d_j should be classified under c_i. Confidence scores allow for *ranking* the classes of $\mathcal{C}$ for a given document d_j. Classifications that produce a rank of classes (instead of a hard classification) are named *ranking classifications* and are useful to some applications. For instance, a ranking classification is of great help to a human expert in charge of taking the final classification decision, since she could thus restrict the choice to the class (or classes) at the top of the list, rather than having to examine the entire set. Also, ranking classification is useful when the results of two classifiers are to be combined to produce a final classification. In this case, the confidence scores

produced by different classifiers are used to take the final decision.

Finally, a ranking classification Γ can be transformed in a hard classification Ψ by means of a threshold τ_i for each class c_i. Decision of classifying d_j under class c_i is taken as follows:

$$\Psi(d_j, c_i) = \begin{cases} T & \text{if } \Gamma(d_j, c_i) \geq \tau_i \\ F & \text{otherwise} \end{cases} \tag{2.7}$$

2.4.3 Single-label and Multilabel Classifications

Depending on the application, different constraints may be imposed on the classification task. One of them is to limit the number of classes of $\mathcal{C}$ that the classifier Ψ may assign to a given document. The case in which each document is to be assigned to exactly one class, is called *single-label* classification, whereas the case in which any number of classes from 0 to $|\mathcal{C}|$ may be assigned to a document is called *multilabel classification*. A special case of single-label classification is the *binary classification*, in which, each document $d_j \in \mathcal{D}$ must be assigned to the class c_i or to its complement $\bar{c}_i$. Examples of application of binary text classifiers are spam filters, which must classify incoming mails as spam or non-spam mails.

In this work, we use two machine learning methods (also called *learners*) to derive document classifiers: the kNN and Support Vector Machine (SVM). These methods have been extensively evaluated for text classification on reference collections and offer a strong baseline for comparison. We now briefly describe each of them.

2.4.4 The kNN Method

A kNN classifier assigns a class label to a test document based on the classes attributed to the k most similar documents in the training set, according to some similarity measure. In the kNN algorithm [59], each test document d_j is assigned a score s_{d_j,c_i}, which is defined as:

$$s_{d_j,c_i} = \sum_{d_t \in N_K(d_j)} similarity(d_j, d_t) \times f(c_i, d_t), \tag{2.8}$$

where $\mathcal{N}_k(d_j)$ are the k neighbors (the most similar documents) of d in the training set and $f(c_i, d_t)$ is a function that returns 1 if the training document d_t belongs to class c_i and 0 otherwise. The scores s_{d_j,c_i} may be transformed in confidence scores $\Gamma(d_j, c_i)$, that is, values in the interval $[0, 1]$ by means of a normalizing process:

$$\Gamma(d_j, c_i) = \frac{s_{d_j,c_i}}{\sum_{c_i}^{|\mathcal{C}|} s_{d_j,c_i}} \tag{2.9}$$

These confidence scores allows kNN to produce a ranking classification. In Chapter 3, we discuss how we derive kNN classifiers using the cosine measure and bibliometric measures as similarity measures.

2.4.5 The SVM Classifier

SVM is a relatively new method of classification introduced by Vapnik in [56] and first used in text classification by Joachims in [31]. The method is defined over a vector space where the problem is to find a hyperplane with the maximal margin of separation between two classes. Classifying a document corresponds to determining its position relative to this hyperplane.

Figure 2.3 illustrates a space where points of different classes are linearly separable. The dashed line represents a possible hyperplane separating both classes. This hyperplane can be described by:

$$(\vec{w} \cdot \vec{x}) + b = 0, \tag{2.10}$$

where $\vec{x}$ is an arbitrary data point that represents the document to be classified, and the vector $\vec{w}$ and the constant b are derived from a training set of linearly separable data. The classification of a vector is achieved by applying the decision function

$$f(\vec{x}) = \text{sign}((\vec{w} \cdot \vec{x}) + b) \tag{2.11}$$

which determines the position of $\vec{x}$ relative to the hyperplane.

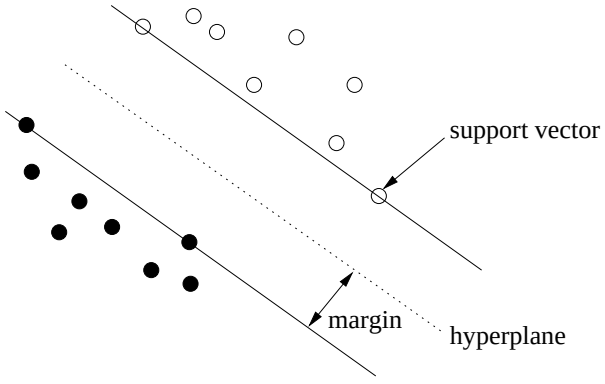


Figure 2.3: The SVM classifier. A separating hyperplane is found by maximizing the margin between the candidate hyperplane and the classes.

In Figure 2.3, the solid lines represent how much the hyperplane can be moved while still separating the classes. The SVM classifier tries to maximize the margin between the

hyperplane and the points in the boundaries of each class. This is achieved by solving a constrained quadratic optimization problem. The solution can be found in terms of a subset of training patterns that lie in the marginal planes of the classes, the *support vectors*, and is of the form:

$$\vec{w} = \sum_i v_i \vec{x}_i \tag{2.12}$$

where each v_i is a learned parameter and each x_i is a support vector. The decision function can be written as:

$$f(\vec{x}) = \text{sign}(\sum_i v_i (\vec{x} \cdot \vec{x}_i) + b) \tag{2.13}$$

In the original data space, also called the *input space*, classes may not be separable by a hyperplane. However, the original data vectors can be mapped to a higher dimensional space, called the *feature space*, where classes are linearly separable. This is achieved through the use of *kernel functions*. Using kernel functions the optimization problem is solved in the feature space, instead of the input space, and the final decision function thus becomes:

$$f(\vec{x}) = \text{sign}(\sum_i v_i \kappa(\vec{x} \cdot \vec{x}_i) + b) \tag{2.14}$$

where κ is the kernel function.

The support vector machine method originally performs only binary classification: a document belongs or not to a given class. However some recent implementations, like the LIBSVM package [15] we use in this work, also offer the option to generate ranking classifications.

2.5 Evaluation

In this section, we describe the measures we use in Chapter 4 to evaluate the effectiveness of the classifiers we obtained using link and text information. We also describe the ten-fold cross-validation method used to obtain distinct samples of collections in order to evaluate a classification method.

2.5.1 Precision and Recall

Classification effectiveness is usually measured in terms of the classic information retrieval notions of precision (p) and recall (r), adapted to the case of document classification [48].

Measures of precision and recall can be derived for each class c_i in the set of classes $\mathcal{C}$. Precision and recall for a given class c_i are better defined by considering the *contingency table* of class c_i (see Table 2.1). FP_i (*false positives under* c_i) is the number of test documents

Class c_i		Expert judgments	
		YES	NO
Classifier judgments	YES	TP_i	FP_i
	NO	FN_i	TN_i

Table 2.1: The contingency table for class c_i.

incorrectly classified under c_i. TN_i (*true negatives under* c_i), TP_i (*true positives under* c_i) and FN_i (*false negatives under* under c_i) are defined accordingly. Precision p_i and r_i of a classifier for class c_i are defined as:

$$p_i = \frac{TP_i}{TP_i + FP_i} \tag{2.15}$$

$$r_i = \frac{TP_i}{TP_i + FN_i} \tag{2.16}$$

2.5.2 The F-measure

In classification tasks, precision and recall are computed for every class as in last section. This yields a great number of values, making the tasks of comparing and evaluating algorithms more difficult. It is often convenient to combine precision and recall into a single quality measure. One of the most commonly used such measures is the *F-measure* [60].

The F-measure combines precision and recall values and allows for the assignment of different weights to each of these measures. It is defined as:

$$F_\alpha = \frac{(\alpha^2 + 1)pr}{\alpha^2 p + r} \tag{2.17}$$

where α defines the relative importance of precision and recall. When $\alpha = 0$, only precision is considered. When $\alpha = \infty$, only recall is considered. When $\alpha = 0.5$, recall is half as important as precision, and so on.

The most used of the *F-measure* is the F_1-measure which is obtained by assigning equal weights to precision and recall by defining $\alpha = 1$:

$$F_1 = \frac{2rp}{p + r} \tag{2.18}$$

The F_1 measure allows us to conveniently analyze the effectiveness of the classification algorithms used in our experiments on each of the used classes.

It is also common to derive a unique F_1 value for a classifier, by computing the average of F_1 of individual classes. Two averages are considered in the literature [60]: *micro-average* $F1$ ($micF_1$) and *macro-average* F_1 ($macF_1$). Micro-average F_1 is computed by considering recall and precision over all classes, that is, the global precision is computed as:

$$p_g = \frac{\sum_{i=1}^{|\mathcal{C}|} TP_i}{\sum_{i=1}^{|\mathcal{C}|}(TP_i + FP_i)} \tag{2.19}$$

and the global recall is computed as:

$$r_g = \frac{\sum_{i=1}^{|\mathcal{C}|} TP_i}{\sum_{i=1}^{|\mathcal{C}|}(TP_i + FN_i)} \tag{2.20}$$

Thus, micro-average F_1 is defined as:

$$micF_1 = \frac{2 r_g p_g}{p_g + r_g} \tag{2.21}$$

Macro-average F_1 is computed as:

$$macF_1 = \frac{\sum_{i=1}^{|\mathcal{C}|} F_{1i}}{|\mathcal{C}|} \tag{2.22}$$

where F_{1i} is the value of F_1 measure for class c_i.

2.5.3 Cross-Validation

Cross-validation has become a standard method for evaluating document classification [38, 48]. It consists in building k different classifiers: $\Psi_1, \Psi_2, \ldots, \Psi_k$. The classifiers are built by dividing the initial corpus Ω (See Section 2.4.1) into k disjoint sets: $Te_1, Te_2, \ldots, Te_k$. classifier Ψ_i is trained using $\Omega - Te_i$ as the training set and is evaluated using Te_i as the test set. Each classifier is evaluated, usually using precision, recall or F1 measures and the average of the k measure is taken as the final evaluation. The most used value of k is 10 and the the method is called *ten-fold cross-validation*.

2.6 Bayesian Networks

A Bayesian network [42] (also known as *inference network* or *belief network*) is a graphical formalism for representing independences among the variables of a joint probability

distribution. Bayesian networks have been shown to produce good results when applied to information retrieval problems, both for simulating traditional models [43, 44, 54] and for combining information from different sources [9, 49]. In this section, we give a general introduction to Bayesian networks and in Section 3.2.2 we show how the formalism can be used to combine the results of classifiers.

In a Bayesian Network, the probability distribution is represented through a directed acyclic graph, whose nodes represent the random variables of the distribution. Thus, two random variables, X and Y, are represented in a Bayesian network as two nodes in a directed graph, also referred to as X and Y. An edge directed from Y to X represents the influence of the node Y, the *parent* node, on the node X, the *child* node. Let x be a value taken by variable X and y a value taken by variable Y. The intensity of the influence of the variable Y on the variable X is quantified by the conditional probability $P(x|y)$, for every possible set of values (x, y).

In general, let $\mathbf{P}_X$ be the set of all parent nodes of a node X, $\mathbf{p}_X$ be a set of values for all the variables in $\mathbf{P}_X$, and x be a value of X. The influence of $\mathbf{P}_X$ on X can be modeled by any function $\mathcal{F}$ that satisfies the following conditions:

$$\sum_{x \in \mathbf{x}} \mathcal{F}(x, \mathbf{p}_X) = 1 \tag{2.23}$$

$$0 \leq \mathcal{F}(x, \mathbf{p}_X) \leq 1. \tag{2.24}$$

where $\mathbf{x}$ is the set of possible values for variable X. The function $\mathcal{F}(x, \mathbf{p}_X)$ provides a numerical quantification for the conditional probability $P(x|\mathbf{p}_X)$. Let $\mathbf{X} = \{X_1, X_2, ..., X_n\}$ be the set of variables in a Bayesian network. The joint probability distribution over $\mathbf{X}$ is given by:

$$P(x_1, x_2, ..., x_n) = \prod_{i=1}^{n} P(x_i|\mathbf{p}_{X_i}) \tag{2.25}$$

To illustrate, Figure 2.4 shows a Bayesian network for a joint probability distribution $P(x_1, x_2, x_3, x_4, x_5)$, where x_1, x_2, x_3, x_4, and x_5 refer to values of the random variables X_1, X_2, X_3, X_4, and X_5, respectively. The node X_1 is a node without parents and is called a *root node*. The probability $P(x_1)$ associated with a value x_1 of the root node X_1 is called a *prior probability* and can be used to represent previous knowledge of the modeled domain. By applying Equation (2.25), the joint probability distribution for the network shown in Figure 2.4 can be computed as:

$$P(x_1, x_2, x_3, x_4, x_5) = P(x_1)P(x_2|x_1)P(x_3|x_1)P(x_4|x_2, x_3)P(x_5|x_3)$$

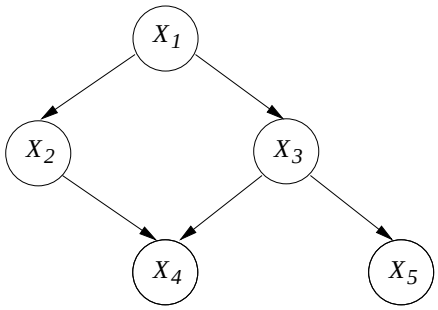


Figure 2.4: Example of a Bayesian network.

The most common task we wish to solve using Bayesian networks is probabilistic inference. Given an evidence, we can calculate the posterior probability of a possible explanation by applying the Bayes' rule:

$$P(r|e) = \frac{\sum_{\mathcal{U}-\{r\}} P(\mathcal{U}, e)}{P(e)} \tag{2.26}$$

where $P(r|e)$ denotes the probability that random variable R has value r given evidence e and $\mathcal{U}$ is a set representing the universe of variables in the model. The denominator is just a normalizing constant that ensures the posterior probability adds up to 1. Notice that $P(\mathcal{U}, e)$ can be obtained through application of Equation (2.25).

To illustrate this inference process, we calculate the probability $P(w|x)$ for the Bayesian network presented in Figure 2.5. In this network all the variables are binary, that is, they can assume only two possible values. The network in the figure presents a method for combining evidences, using a *noisy-OR* node. In particular, the "or" mark above node W means that $P(W|Z_1, Z_2)$ is defined in such way that W is true if anyone of their parent nodes, Z_1 and Z_2, are true and W is false if both nodes Z_1 and Z_2 are false. In other words, $P(w|\overline{z}_1, \overline{z}_2) = 0$ and $P(w|z_1, z_2) = P(w|\overline{z}_1, z_2) = P(w|z_1, \overline{z}_2) = 1$, where z_i denotes that node $Z_i = 1$ and $\overline{z}_i$ denotes that node $Z_i = 0$. We calculate the probability $P(w|x)$ for this case.

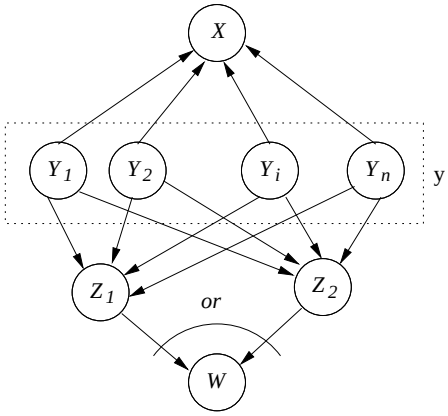


Figure 2.5: Example of a Bayesian network with a *noisy-OR* node.

$$
\begin{aligned}
P(w|x) &= \frac{\sum_{\mathbf{y},\mathbf{z}} P(x,\mathbf{y},\mathbf{z},w)}{P(x)} \\
&= \eta \sum_{\mathbf{y},\mathbf{z}} P(w|\mathbf{z})\ P(\mathbf{z}|\mathbf{y})\ P(x|\mathbf{y})\ P(\mathbf{y}) \\
&= \eta \sum_{\mathbf{y}} P(x|\mathbf{y})\ P(\mathbf{y}) \sum_{\mathbf{z}} P(w|\mathbf{z})\ P(\mathbf{z}|\mathbf{y}) \\
&= \eta \sum_{\mathbf{y}} P(x|\mathbf{y})\ P(\mathbf{y})\ [P(z_1,\overline{z}_2|\mathbf{y}) + P(\overline{z}_1,z_2|\mathbf{y}) + P(z_1,z_2|\mathbf{y})] \\
&= \eta \sum_{\mathbf{y}} [1-(1-P(z_1|\mathbf{y}))(1-P(z_2|\mathbf{y}))]\ P(x|\mathbf{y})\ P(\mathbf{y}) \qquad (2.27)
\end{aligned}
$$

where $\mathbf{z}$ is used to refer to any of the possible states of nodes Z_1 and Z_2. Notice that similar network and an equation similar to Equation (2.27) will be used in evidence combination methods in following sections.

Chapter 3

Classification Approaches and Collections

In this Chapter, we detail the approaches we adopt to use bibliometric measures to classify documents and describe the document collections we use to evaluate these approaches. In Section 3.1, we describe the first approach which corresponds to deriving classifiers based on bibliometric measures that we call *bibliometric classifiers*. In Section 3.2, we present the second approach, that of combining the results of bibliometric classifiers with the results of text-based classifiers, aiming to obtain a final improved classification. In Section 3.3 we describe the three collections of linked documents we use to evaluate the classifiers and the combination methods. A detailed experimental evaluation of the methods over the three collections is presented in the next chapter.

3.1 The Obtained Classifiers

In this Section, we describe how we derived bibliometric and text-based classifiers using the kNN an SVM classification methods. These methods were chosen because they are considered two of the most successful methods for classifying documents [31, 48]. Besides, Yang et al [60] have shown that the two methods are robust to skewed category distribution, which is a common characteristic in document collections.

Both versions of kNN and SVM classifiers we use generate ranking classifications. In this work, we need ranking classification for the purpose of combining results of classifiers, as discussed in Section 3.2. However, the documents of the derived collections we use are single-labeled documents. Thus, for each classification method we obtain a final

single-labeled classification by choosing for each test document the class at the top of the corresponding rank, that is, the class with highest confidence score.

kNN Classifiers

As described in Section 2.4.4, the kNN infers the class of a given test document d_j by considering the classes of the k training documents most similar to d_j according to some similarity function $similarity(d_j, d_t)$, where d_t is a document in the training set. Any similarity measure between documents can be used in place of the function $similarity(d_j, d_t)$ in Equation (2.8). Consequently, we can directly derive bibliometric kNN classifiers by substituting the function for the value of the corresponding bibliometric similarity measure computed between test document d_j and each document of the training set. For instance, we can obtain a kNN classifier based on the co-citation measure by rewriting Equation (2.8) as:

$$s_{d_j,c_i} = \sum_{d_t \in N_K(d_j)} co{-}citation(d_j, d_t) \times f(c_i, d_t), \tag{3.1}$$

where $N_k(d_j)$ is the set of k documents in the training set most similar to d_j by the co-citation measure as defined in Equation 2.4. In the same way, we can derive kNN classifiers using the bib-coupling and Amsler measures, by substituting the $similarity(d_j, d_t)$ function for the bib-coupling and Amsler measures, as defined in Equations (2.5) and (2.6), respectively.

Similarly, any text-based similarity measure between documents could be used to derive text-based versions of kNN classifiers. In this work, we use the cosine similarity measure defined in Equation (2.3). We consider each document as a vector of term weights TF-IDF and the cosine of the angle between any two vectors is used as the similarity measure between the the corresponding documents.

We experimented with different values for k, both for bibliometric and cosine kNN classifiers. Since values greater than 30 did not cause any significant change in the results, we fixed k equal to 30 in all kNN classifiers we used.

SVM Classifiers

The SVM classifier considers each document as vector in a n-dimensional feature space, where n is the number of distinct features of documents in the training set. It expects as

input for each document d_j a set of pairs $\langle feature_f, feature\_value_f \rangle, 1 \leq f \leq n$. We obtain SVM classifiers for a given bibliometric measure $\mathcal{B}$, by using as features all the documents d_f for which there is at least one training document d_t such that $\mathcal{B}(d_t, d_f) > 0$. We use the value $\mathcal{B}(d, d_f)$ as the the value of feature d_f in a document d. We obtain text-based SVM classifiers by using the terms of each document as the features of the document and the TF-IDF as the feature value. The SVM classifiers were generated with the SVMLIB software [15], using the *Radial Basis Function*(RBF) Kernel.

Classifiers and No-Information Documents

Some test documents are *no-information* documents for a given classifier, that is, a document that contains no useful information for the classifier to infer its class. In the case of text-based classifiers, a no-information document is one that has no term or has no term in common with any training document.

For bibliometric classifiers, a no-information document is one that has no links or has no parent or no child document in common with any training document, according to the specific bibliometric measure considered. In the case of co-citation, no information corresponds to absence of common parents, whereas for bib-coupling, it corresponds to the absence of common children and, for the Amsler measure, it corresponds to the absence of any linked document (parent or child) in common with some training document.

In order to minimize classification error, the classifiers always assign the most popular class to no-information test documents. We refer to this assignment strategy as *default classification*.

3.2 Methods for Combining Results of Classifiers

Methods that combine the results of two classifiers decide the class of a given test document by choosing between the classes output by two distinct classifiers. In the decision process, these methods use the highest confidence score associated by each classifier to its output class. In this section, when we refer to a confidence score of a classifier we mean the highest confidence score the classifier assigns to a given document and which is used to determine the document's class.

We present two methods we use in the experiments of the next chapter to automatically combine the results of bibliometric and text-based classifiers. The first method we describe

is the *Reliability-Based Combination* [19], a method we propose which uses the most reliable classifier to decide which classifier result to choose. The second method [9] uses a Bayesian network to obtain a linear combination of results of individual classifiers.

3.2.1 Reliability-Based Combination

Since combination methods use confidence scores output by classifiers to decide the class of a test document, it is important that these classifiers be reliable regarding the confidence scores they assign to classes. An *ideally reliable* classifier is one that provides confidence scores exactly proportional to its classification effectiveness. In other words, given a set of documents $\mathcal{D}_{cs}$, for which the ideally reliable classifier assigns class labels with confidence score cs, it should correctly classify $p \times |\mathcal{D}_{cs}|$ documents of the set $\mathcal{D}_{cs}$.

We define the accuracy acc_{cs} for a given confidence score cs output by a classifier Ψ as the ratio: $correct_{\Psi_{cs}}/docs\Psi_{cs}$, where $correct_{\Psi_{cs}}$ is the number of documents that Ψ correctly classifies when its confidence score is cs, and $docs\Psi_{cs}$ is the total number of documents to which Ψ assigns cs. The reliability of a classifier Ψ can be evaluated by associating each confidence score cs assigned by Ψ with its corresponding accuracy acc_{cs}. Once we have the pairs (cs, acc_{cs}), we obtain a linear regression of these points. The more reliable a classifier is, the more its corresponding regression line approximate the identity function $acc(cs) = cs$ which corresponds to the ideally reliable classifier.

The notion of reliable classifier can be used to derive the *reliability-based combination* which is based on the following idea: If one of the classifiers to be combined presents high accuracy and provides reliable confidence scores, it is possible to use it as a guide in the combination process. In other words, in the cases where the more reliable classifier assigns a document to a category with low confidence score we can expect it to be wrong (low accuracy). Thus, in such cases, it would be better to use the classification decision provided by the second classifier. This idea is formally presented in the algorithm of Figure 3.1.

The algorithm first obtains the set $\mathcal{A}_{tr}$, containing for each document i the pairs (c_{Ai}, y_{Ai}) (lines 3-6). It then executes similar steps to obtain the set $\mathcal{B}_{tr}$ (lines 7-8). Next, it computes the accuracy acc_{cs} for each distinct confidence score cs among the values c_{Ai} output by classifier A. The value of acc_{cs} is obtained by dividing the number of pairs $(cs,1)$ in $\mathcal{A}_{tr}$ by by the total number of documents to which A assigns cs, i.e., the number of pairs $(cs, 1)$ plus the number of pairs $(cs, 0)$ in $\mathcal{A}_{tr}$. Then, the algorithm obtains the regression line from pairs (cs, acc_{cs}) for classifier A (lines 9-10). In the same way, it obtains

```
1 Let A be the most reliable classifier to be combined;
2 Let B be the least reliable classifier to be combined;
3 Let A_tr be a set of points {c_Ai, y_Ai}, where c_Ai, 0 ≤ c_Ai ≤ 1, represents
4     the confidence score of A in the classification given for document i in
5     the training collection (0 ≤ c_Ai ≤ 1) and y_Ai is 1 if the classification
6     provided by A for i is correct and is 0 otherwise;
7 Let B_tr be a set of points {c_Bi, y_Bi}, where y_Bi is 1 if the classification
8     provided by B for i is correct and is 0 otherwise;
9 Let f_A(x) = b + ax be the function that best fits the points (cs, acc_cs)
10    derived from A_tr;
11 Let f_B(x) = d + cx be the function that best fits the points (cs, acc_cs)
12    derived from B_tr;
13 if (a == c) {
14    if (b > d)
15        p = 0;
16    else
17        p = 1;
18 } else
19        p = (b−d)/(c−a);
20 for each document i in the test collection {
21    if (c_Ai > p)
22       classification of document i is given by A;
23    else
24       classification of document i is given by B;
25 }
```

Figure 3.1: Reliability-Based Combination.

the regression line from pairs (cs, acc_{cs}) corresponding to classifier B (lines 11-12). It then finds the confidence score p where the most reliable classifier A tends to be always better than the least reliable classifier B, that is, the point p where the regression lines cross each other (lines 13-19) and uses this point to determine which classifiers provide the best decisions (lines 19-23). In sum, decisions from classifier A are preferable if it yields belief estimations greater than p.

3.2.2 Combination Using Bayesian Network

In this work, we use the Bayesian network shown in Figure 3.2, proposed by Pavel et al. [9], as a means of combining the results of two distinct classifiers. In the figure, the root nodes,

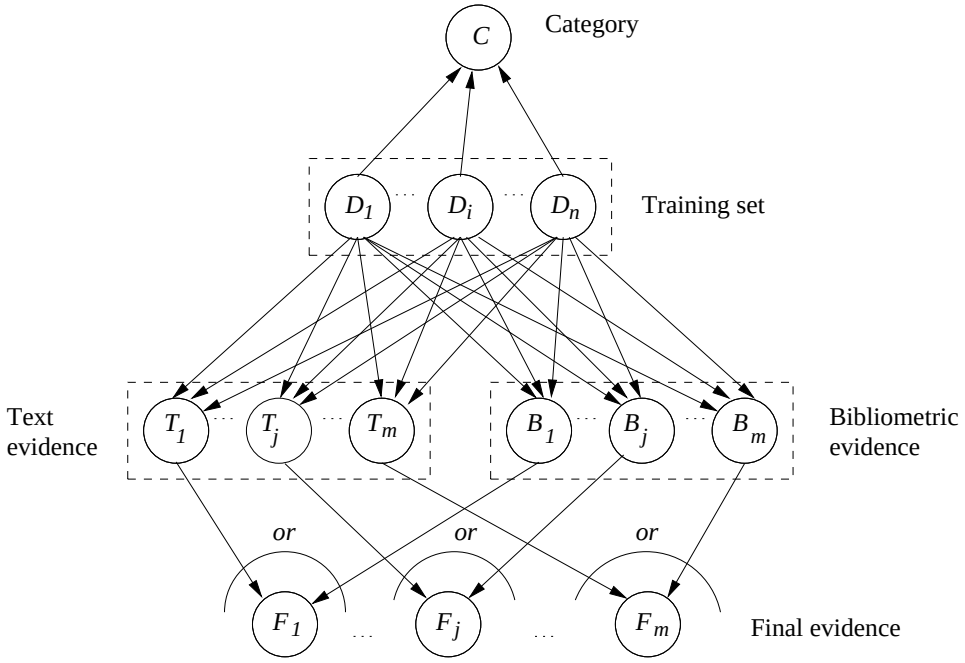


Figure 3.2: Bayesian network model to combine a text-based classifier with evidence from link structure.

labeled D_1 through D_n, represent our prior knowledge about the problem, i.e., the training set. Node C represents a category c. The edges from nodes D_i to c represent the fact that observing a set of training documents will influence the observation of a category c.

Each node T_j represents evidence from the text-based classifier indicating that test document j belongs to category c. An edge from a node D_i to a node T_j represents the fact that the training document i is related to test document j by text information. Thus if i belongs to class c we may infer that there are chances for document j to belong to class c.

Each node B_j represents evidence, given from the bibliometric classifier, indicating that document j belongs to category c. An edge from a node D_i to a node B_j indicates the evidence given by bibliometric information that the training document i is related to the test document j. Thus, if i is classified under category c, there are grounds to infer that j should be also considered as candidate for category c.

Given these definitions, we can use the network to determine the probability that a test document j belongs to category c, by deriving an equation in a way similar to one used to derive Equation (2.27). This translates to the following equation:

$$P(f_j|c) = \eta \sum_{\mathbf{d}} \Big[1 - \big(1 - W_t\ P(t_j|\mathbf{d})\big)\big(1 - W_b\ P(b_j|\mathbf{d})\big)\Big]\ P(c|\mathbf{d}) \tag{3.2}$$

where $\eta = 1/P(c)$ is a normalizing constant and $\mathbf{d}$ is a possible state of all the variables D_i. The probability $P(c|\mathbf{d})$ is now used to select only the training documents that belong to the category we want to process. We define $P(c|\mathbf{d})$ as:

$$P(c|\mathbf{d}) \quad = \quad \begin{cases} 1 & \text{if } \forall i,\ d_i = 1 \text{ if } i \in \mathcal{C} \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

where $\mathcal{C}$ is the set of training documents that belong to category c. By applying Equation (3.3) to Equation (3.2), we obtain:

$$P(f_j|c = \eta \sum_{\mathbf{d}_c} \Big[1 - \big(1 - W_t\ P(t_j|\mathbf{d}_c)\big)\big(1 - W_b\ P(b_j|\mathbf{d}_c)\big)\Big] \tag{3.4}$$

where $\mathbf{d}_c$ is the state of variables D_j where only the variables corresponding to the training documents of class c are active. Constants W_t and W_b are the weights given to the text-based classifier and to the bibliometric classifier, respectively. They can be used to regulate the importance of each source of evidence on the final result. The introduction of weights in the model is accomplished by the use of a *noisy-OR* combination [42].

To compute the final probability, we simply define $P(t_j|\mathbf{d}_c)$ and $P(b_j|\mathbf{d}_c)$ as the confidence scores assigned by the text-based classifier and the bibliometric classifier, respectively, to the association of document j with class c. Since the confidence scores are values between 0 and 1 they can be used as probability values.

The Bayesian Network combination method produces a ranking classification for each test document j. The rank is given by the values of $P(f_j|c)$ for the distinct classes c. The class c with highest probability $P(f_j|c)$ is the one chosen to be the class of document j by the combination method.

3.3 Document Collections

In this section, we describe the collections used in our comparative study of classification and combination methods to be described in the next chapter. We use three collections

with distinct characteristics of link and textual information. In section 3.3.1, we describe ACM8, a sub-collection of the ACM digital library[1]. In section 3.3.2, we describe Cade12, a collection of Web pages derived from the the Cadê directory[2]. Section 3.3.3 describes Wiki8, a sub-collection of the Wikipedia[3] encyclopedia.

3.3.1 The ACM8 Collection

The ACM8 collection is a sub-collection of the ACM Digital Library. All the text contained in the title and abstract, when available, was used to index the documents. Note that many citations in the original ACM Digital Library could not be traced to the corresponding paper for several reasons. Among them, the fact that many cited papers do not belong to this digital library and also due to the imprecise process used to match the citation text to the corresponding paper [36]. High precision and recall in this pre-processing phase is hard to be achieved due to problems such as differences in the writing style for names of authors and conferences in the citations. This problem is particularly important in the case of the ACM Digital Library, since most citations were obtained with OCR after scanning, which introduces many errors, making the matching process even harder.

To simulate a more realistic situation in which most citations are available, we selected a subset of the ACM Digital Library having only documents with at least four matched citations to distinct references. This is a very reasonable assumption since most papers of the ACM Digital Library (even short ones) have more than four citations. In fact, the average number of citations in the ACM Digital Library is 11.23.

The resulting ACM8 collection is a set of 6,680 documents, without stop words. Documents are labeled under the 8 largest categories of the ACM Digital Library taxonomy, which we list here in descending order of their sizes: (1) *D-Software*, (2) *H-Information Systems*, (3) *I-Computing Methodologies*, (4) *B-Hardware*, (5) *C-Computer Systems Organization*, (6) *F-Theory of Computation*, (7) *K-Computing Milieux* and (8) *G-Mathematics of Computing*. Classes *A-General Literature*, *E-Data* and *J-Computer Applications* of the ACM taxonomy were not used because they contain less than 20 documents in this sub-collection. Similarly to our Web collection, each paper is classified into only one category. These classes are first-level classes defined by the ACM Computing Classification System [1]. Usually, the authors assign sub-classes to their documents to make classification

[1] `http://portal.acm.org/dl.cfm`

[2] `http://www.cade.com.br/`

[3] `http://en.wikipedia.org/wiki/Main_Page`

more specific and are free to use sub-classes of different first-level classes. Since, in our classification experiments, we were mainly concerned with one-label classification, the ACM8 was composed only of documents that were assigned sub-classes of the same first-level class. Thus, none of the ACM8 documents were considered multilabel cases by their authors.

Figure 3.3 shows the category distributions for the ACM8 collection. Note that the ACM8 collection has a very skewed distribution, where the two most popular categories represent more than 50% of all documents.

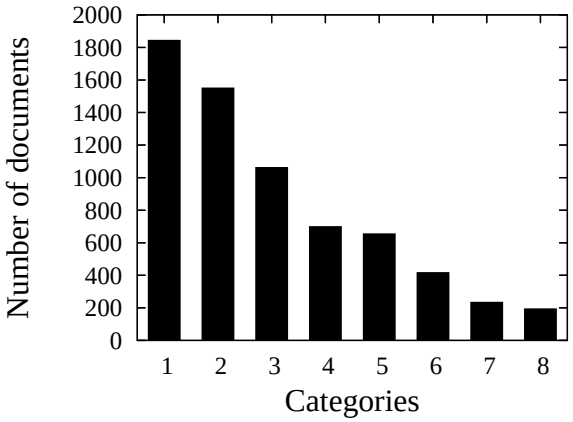


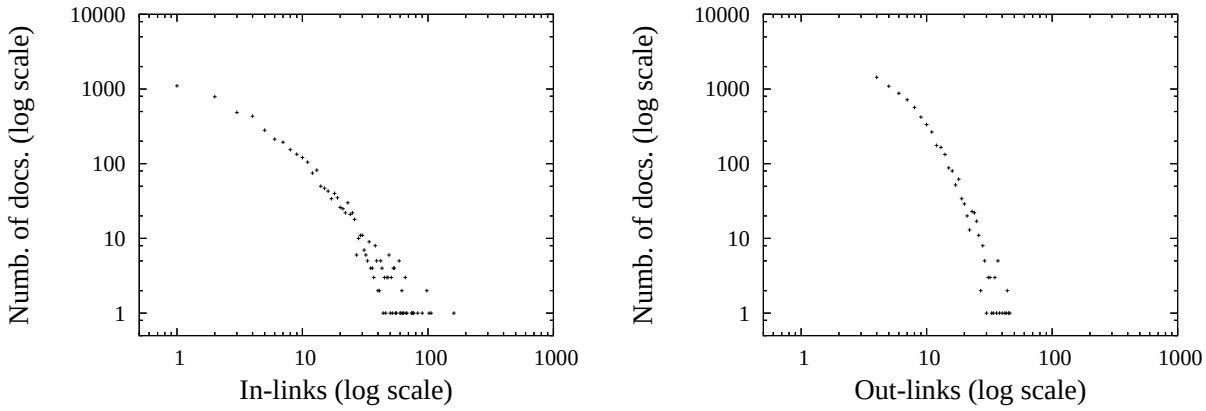
Figure 3.3: Category distribution for the ACM8 collection.

Table 3.1 shows some statistics about links (citations) in the ACM8 collection. Links from ACM8 articles to articles outside ACM8 correspond to 77.8% of the links in the collection. The external documents cited include both articles of the ACM digital libraries not included in ACM8 and publications outside the ACM digital library. The information about these publications came from the DBLP(http://dblp.uni-trier.de/) collection. Since we have no information about the external documents, in-links can be derived only from internal links, while out-links can be derived from all links. Thus the number of in-links in the ACM8 collection is $11,510$, while the number of out-links is almost four times higher. The first two percentages in Table 3.1 are computed over the total of 51,897 links in the collection.

Figure 3.4 shows the distribution of in-links and out-links for the ACM8 collection. It can be seen that the majority of the documents has less in-links than out-links.

Statistics	ACM8.
Internal links	11,510 (22.18%)
Links from external documents to ACM8 documents	0
Links from ACM8 documents to external documents	40,387 (77.82%)
ACM8 documents with no in-links	1,941 (29.0%)
ACM8 documents with no out-links	0
Average of in-links by document	4.72
Average of out-links by document	7.77

Table 3.1: Statistics for the ACM8 collection.



(a) In-link distribution.

(b) Out-link distribution.

Figure 3.4: Link distribution for the ACM8 collection.

3.3.2 The Cade12 Collection

The Cade12 is a sub-collection of pages indexed by the Brazilian Web directory Cadê. All pages in the Cadê directory were manually classified by human experts. Since they were also indexed by the TodoBR search engine[4], we built the Cade12 collection by obtaining text and links directly from the TodoBR database. The content of each document in Cade12 collection is composed of the text contained in the body and title of the corresponding Web page, after discarding HTML tags.

The resulting collection is composed of 42,391 documents, containing a vocabulary of 191,962 distinct terms. In our experiments we used only the 10,000 terms with highest information gain (*infogain*). Information gain is used to measure the capacity of a feature (term) of separating documents into classes. It is defined by Equation (3.5) [48].

$$infogain(t_k, c_i) = \sum_{c_i \in \mathcal{C}} P(c_i) \times \sum_{c \in \{c_i, \bar{c}_i\}} \sum_{t \in \{t_k, \bar{t}_k\}} P(t, c) \times log \frac{P(t, c)}{P(t) \times P(c)} \qquad (3.5)$$

where $\mathcal{C}$ is the set of classes and the probabilities are interpreted on an event space of documents. For intance, $P(\bar{t}_k, c_i)$ denotes the probability that, for a random document x, term t_k does not occur in x and x belongs to class c_i. The probabilities are computed over the training set. Information gain is used as feature selection – only the m terms with the greatest information gain are used as features of the documents, for some arbitrary $m > 0$. Text classifiers are trained using only these m terms..

Documents in Cade12 are labeled under 12 first-level classes of the Cadê directory, listed in descending order of their sizes: (1) *Services*, (2) *Society*, (3) *Recreation*, (4) *Computers*, (5) *Health*, (6) *Education*, (7) *Internet*, (8) *Culture*, (9) *Sports*, (10) *News*, (11) *Science* and (12) *Shopping*. Figure 3.5 shows the category distribution for the Cade12 collection. Note that the collection has a skewed distribution and the three most popular categories represent more than half of all documents.

The link information of the Cade12 collection was extracted from the set of 40,871,504 links of the TodoBR database. As observed by the authors in [9], the richer the link information considered, the better the accuracy obtained by link-based classifiers. In fact, this was an important reason for choosing Cadê. With Cadê we are not restricted to a limited source of links since Cadê is a subset of TodoBR, which is a large collection containing most of the link information available in Brazilian Web pages. Links from Web

[4]TodoBR is a trademark of Akwan Information Technologies, which was acquired by Google in July 2005.

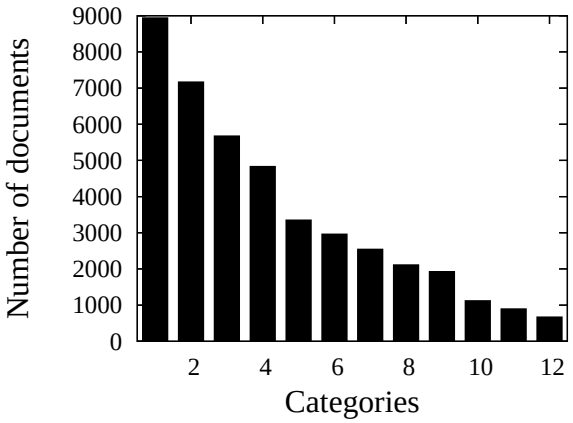


Figure 3.5: Category distribution for the Cade12 collection.

pages of directory web sites were removed to avoid a bias in the results.

Table 3.2 shows statistics about link information of the Cade12 collection. The first three percentages in the table are computed over the total of 56,4316 links in the collection.

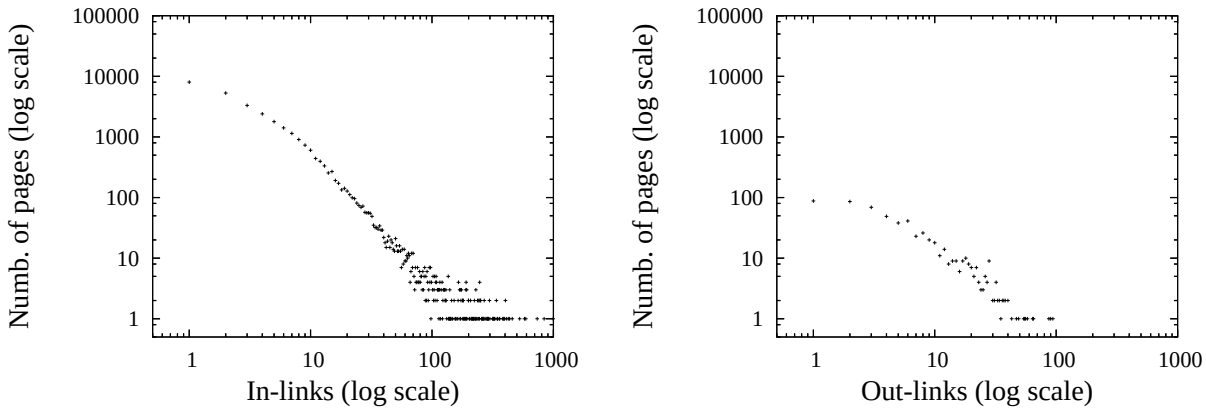
Statistics	Cade12
Internal links	3,830 (0.68%)
Links from external pages to Cade12 pages	554,592 (98.28%)
Links from Cade12 pages to external pages	5,894 (1.04%)
Cade12 pages with no in-links	4,392 (10.36%)
Cade12 pages with no out-links	40,723 (96.06%)
Mean of in-links by document	12.57
Mean of out-links by document	0.13

Table 3.2: Link statistics for the Cade12 collection.

Figure 3.6 presents the distribution of in-links and out-links in the Cade12 collection. Note that most pages have no out-links at all, but the majority does have in-links.

3.3.3 The Wiki8 Collection

Wiki8 is a sub-collection of the Wikipedia encyclopedia in English, captured from the Wikipedia dump files. The Wikipedia collection is periodically stored in compressed format files called *dump files* to facilitate download. We used the dump file obtained from "http://download.wikimedia.org/enwiki/2006816". The text of each Wiki8 document is derived from the text of the corresponding Wikipedia article by discarding HTML tags



(a) In-link distribution.

(b) Out-link distribution.

Figure 3.6: Link distribution for the Cade12 collection.

and by filtering out stop words. We also removed meta-information inside documents that contained information about the classes of the document. As a consequence some documents became empty. The resulting collection contains 28.045 documents and a total of 101,563 terms, however, in our experiments we used only the 10,000 terms with best information gain. For each document we maintained the links to other documents. Links to Wikipedia categories were not used.

The Wiki8 collection is composed of 28,044 documents labeled under 8 categories: (1) *History*, (2) *Politics*, (3) *Chemistry*, (4) *Philosophy*, (5) *Biology*, (6) *Mathematics*, (7) *Astronomy* and (8) *Computer Sciences*. We chose these classes due to their general nature, easily assessed by the human judges that participated in the user study to be described in Section 4.3. As shown in Figure 3.7 the category distribution in Wiki8 is also skewed. More than half of the documents belong to the most popular class, History.

Table 3.3 shows some statistics about the links in the Wiki8 collection. By *external in-links* we mean links to pages in Wiki8 from pages of Wikipedia that were not included in Wiki8. *External out-links* are links from any page in Wiki8 to any page out of it (including pages out of Wikipedia). The first three percentages in Table 3.3 are computed over the total of 2,994,659 links in the collection. There are more in-links than out-links in Wiki8, but about 6% of the documents do not have in-links and 0.3% do not have outlinks. Figure 3.8 shows the distribution of in-links and out-links in the collection.

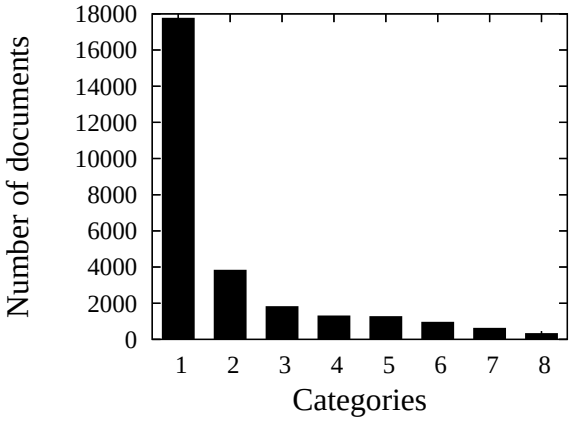
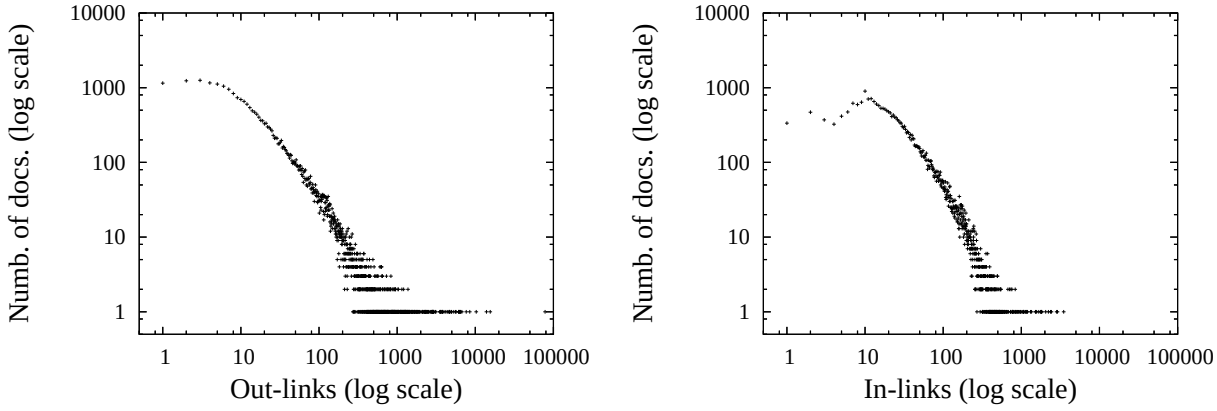


Figure 3.7: Category distribution for the Wiki8 collection.

Statistics	Wiki8
Internal Links	186,844 (6.24%)
External in-links	1,584,587 (52.91%)
External out-links	1,223,228 (40.85%)
Wiki8 pages with no in-links	1,6862 (6%)
Wiki8 pages with no out-links	84(0.3%)
Mean of in-links by document	63.16
Mean of out-links by document	50,28

Table 3.3: Link statistics for the Wiki8 collection.



(a) In-link distribution.

(b) Out-link distribution.

Figure 3.8: Link distribution for the Wiki8 collection.

Chapter 4

Experimental Results

In this chapter, we describe experimental evaluations of the bibliometric classifiers and the combination methods presented in Chapter 3. The experiments were conducted using the ACM8, Cade12 and Wiki8 collections as test beds. We also present the results of investigations about the documents to which bibliometric classifiers failed to assign the correct class.

Section 4.1 depicts the results of a series of experiments comparing the effectiveness of the distinct bibliometric classifiers and text-based classifiers. Section 4.2 describes experiments that show the reliability of bibliometric classifiers in the distinct collections and presents comparative analysis of the methods we use to combine the results of both bibliometric classifiers and text-based classifiers. Section 4.3 describes the investigations about the documents that bibliometric classifiers failed to classify.

4.1 Experimenting with Bibliometric Classifiers

In this section, we present the results of experiments with bibliometric and text-based classifiers trained using the kNN and the SVM classification methods, as described in Section 3.1. The experiments were conducted with the following objectives:

- Compare the effectiveness of bibliometric classifiers and text-based classifiers in each collection.

- Evaluate how the three bibliometric measures (co-citation, bib-coupling and Amsler) are influenced by link distribution in each collection.

In all classification experiments, we used a ten-fold cross validation and we evaluated each run using macro and micro F_1 measures. The final results of each experiment represent the average of the ten runs for both measures. For each collection, we used the text-based classification of each method as the baseline for the method. Thus, the results of the kNN classifier using the cosine measure and the results of the SVM classifier with TF-IDF were taken as baselines.

Table 4.1 presents the micro-averaged and macro-averaged F_1 values for the link and text-based classifiers over the ACM8 collection. The two last columns of the table show the percentage of gain of each classifier over the text-based classifier for each method.

Method	Similarity	$micF_1$	$macF_1$	Gains over text classifier (%)	
				$micF_1$	$macF_1$
kNN	co-citation	61.60	52.56	-20	-25.5
	bib-coupling	83.20	78.29	8.1	10.9
	Amsler	84.43	79.41	9.7	12.5
	Text Cosine	76.95	70.57	–	–
SVM	co-citation	59.33	49.98	-26.17	-34.21
	bib-coupling	80.72	74.59	0.4	-0.18
	Amsler	83.08	77.08	3.37	1.46
	Text TF-IDF	80.37	75.97	–	–

Table 4.1: Macro-averaged and micro-average F_1 results for kNN and SVM classifiers applied over the ACM8 collection.

Co-citation-based classifiers presented the worst results over all classifiers. This is because there are few documents that share the same in-links and since co-citation is a measure of the number of in-links two documents have in common, this measure is not sufficiently precise for the classifier to decide the class of a test document. In fact, 29% of the documents do not have co-citation. In the case of ACM8, this can be justified by the small number of in-links in the collection (less than twice the number of documents). For instance, 85% of the documents that kNN with co-citation failed to classify have less than 4 in-links. On the other hand, of the 61.75% of documents that kNN with co-citation correctly classified, only 28% of them have less than two in-links.

Classifiers using the Amsler similarity achieved the best effectiveness for both kNN and SVM methods. However, results are only slightly better than for bib-coupling. Since

the Amsler similarity is a kind of combination between co-citation and bib-coupling, we can conclude that bib-coupling contributed most to the results. This is because there are many pairs of documents that have at least one out-link in common. In fact, 97% of the documents have bibliographic coupling with at least some other document in the collection. This means that not only there are many out-links in ACM8, but cited documents (children) tend to be cited by two or more documents. Also, when using a kNN classifier with the Amsler measure, most test documents have co-occurrence children in the training set. In fact, only 74 no-information cases were found. Thus, kNN rarely used the default classification in ACM8 as a means to decide the class of the test documents.

The text in documents of the ACM8 collection, despite being short, is not noisy, since content-based classifiers also presented a good effectiveness. Table 4.1 also shows that link information is better used to obtain classifiers based on the kNN method, while textual information is better used with SVM.

The same set of experiments using the kNN and SVM methods with bibliometric and text-based information was applied to the Cade12 collection. The results are shown in Table 4.2. Contrary to the ACM8 collection, bib-coupling-based classifiers presented the worst results among the bibliometric classifiers. This is because only 1% of the documents has at least one parent document which is also a parent of another document in the collection. In spite of this scarcity, all the classifiers achieved micro-average F_1 values superior to 21% due to the default classification. This strategy works because of the large number of documents that belong to the most popular class. The small number of documents with bib-coupling values are due to the rareness of out-links in the collection.

On the other hand, 70.49% of the documents are co-cited with other documents. This means that if we do not consider the default classification, the maximum accuracy that could be achieved is about 70%. As Table 4.2 shows, classifiers using Amsler similarity or co-citation similarity almost achieved this limit.

Although the kNN classifier using co-citation performed better than the one using bib-coupling, and better than text-based classifiers, about 30% of the documents were classified using the default classification. Thus, in order to make clear the true contribution of bibliometric information for this collection, we conducted an experiment where we removed the documents for which the classifier applied the default classification. Since the results between the SVM and kNN classifiers presented in Table 4.2 are only slightly different, we used only the kNN classifier, which presented a better effectiveness. Similar experiments were not conducted with the ACM8 and Wiki8 collections because the no-information cases

Method	Similarity	$micF_1$	$macF_1$	Gains (%) over text classifier	
				$micF_1$	$macF_1$
kNN	co-citation	68.51	75.60	36.9	69.9
	bib-coupling	22.09	5.39	-55.8	-87.9
	Amsler	68.56	75.53	37.0	70
	Text cosine	50.03	44.50	–	–
SVM	co-citation	68.91	76.9	27.2	55.7
	bib-coupling	24.08	6.40	-55.6	-87.0
	Amsler	68.09	74.8	25.6	51.47
	Text TF-IDF	54.18	49.38	–	–

Table 4.2: Macro-averaged and micro-average F_1 results for kNN and SVM classifiers applied over the Cade12 collection.

in these collections are rare, corresponding to less than 2% of the documents. The results are shown in Table 4.3. The results of the classification for the whole collection were copied to the first line of the table to facilitate comparison.

kNN with co-citation	$micF_1$	$macF_1$
Using Default Classification	68.51	75.60
Not Using Default Classification	85.29	80.73

Table 4.3: Results for the kNN when considering all documents and when considering only documents that are not no-information documents.

The difference between the two results shows that the lower values for macro-averaged and micro-average F_1 obtained in the first experiment involving all documents are mainly due to lack of link information.

In fact, whenever co-citation information is available, its quality can be considered good for classification in Cade12. Only about 15% of the classification failures in the collection are due to wrong conclusions extracted from the co-citation measure itself. For example, one of the documents has class label *Society* but, kNN assigned label *Recreation* to it because among the k documents that are most related to it by co-citation, 68.3% of them have class label *Recreation* and 31.7% have class label *Society*.

We also trained kNN and SVM classifiers using the Wiki8 Collection. The results are shown in Table 4.4.

Method	Similarity	$micF_1$	$macF_1$	Gains (%) over text classifier	
				$micF_1$	$macF_1$
kNN	co-citation	81.3	68.43	0.5	-0.1
	bib-coupling	86.95	82.31	7.51	20.16
	Amsler	87.73	82.05	8.48	19.78
	Text cosine	80.87	68.50	–	–
SVM	co-citation	74.68	60.09	-15.4	-27.6
	bib-coupling	86.07	80.61	-2.5	-2.9
	Amsler	85.66	80.84	-3.0	-2.5
	Text TF-IDF	88.27	82.99	–	–

Table 4.4: Macro-averaged and micro-average F_1 results for kNN and SVM classifiers applied over the Wiki8 collection.

We note that text-based classifiers presented very good effectiveness in the Wiki8 collection. This high effectiveness is due to the high specificity of text information within each class. There are many terms that occur frequently in one class and are rare in other classes. For example, terms like *stars*, *earth*, *sun*, *solar*, *moon* and *space*, among others occur in at least 20% of the documents of the class Astronomy and are rare in other classes. Also, each of the terms *biology*, *cell*, and *cells* occurred at least in 30% of the documents of the class *Biology* and are nonexistent in other classes. We can also find sets of discriminative terms like these for the other classes of the collection.

The quality of text in Wiki8 is even more evident when we compare it to the quality of text in the other two collections. This comparison was performed by computing the information gain of the terms in the three collections. For each collection, we ranked the terms in descending order of their information gain values and computed the mean information gain of the top k terms. Table 4.5 shows the mean information gain for values of k equal to 100, 1 000 and 10 000 in each collection. We note that the mean values of information gain for Wiki8 is in all cases greater than those of the other collections.

As Table 4.4 shows, bib-coupling-based classifiers performed better than the co-citation-based classifiers, in spite of the fact that there are more in-links than out-links in Wiki8. This happens because co-occurrent children documents are more evenly distributed over the collection than co-occurrent parent documents. In fact, only 1% of the documents have no children in common with any other document, while 12% of the documents do not have

Collection	Average Infogain for k Best Terms		
	$k = 100$	$k = 1000$	$k = 10000$
Wiki8	0.038	0.013	0.0033
ACM88	0.020	0.006	0.000126
Cade12	0.012	0.0049	0.000125

Table 4.5: The Average information gain of the k terms with best informatio gain in each collection.

parents in common with any other document. Since bib-coupling is directly related to the number of children two pages have in common, classifiers using this measure produce better results.

Since only 1% of the documents do not have bib-coupling with any other document, the chances for a document to have no bib-coupling similarity to any training document is also small. In fact, only 0.03% of the test documents are no-information cases. So, as is in the case of the ACM8 collection, almost all mistakes and hits are consequence of the usage of the method (kNN or SVM) and not due to default classification.

The bibliometric classifiers presented accurate results in all the trhee collections, although, in the Wiki8 collection text-based classifiers were more effective. The experiments indicate that, in spite of the differences in the purposes, density and distribution of the links found in the three studied collections, the information extracted from links may play an important role in classification tasks.

4.2 Combining Results of Classifiers

In this section, we present and discuss two set of experiments. With the first set we aim to evaluate the reliability of bibliometric classifiers. As the results of these experiments show, bibliometric classifiers obtained for the three collections are reliable classifiers and this conclusion stimulates the use of the reliability-based combination method. With the second set of experiments we intend to compare the two methods we use to combine classifiers' results: the reliability-based method and the Bayesian Network method.

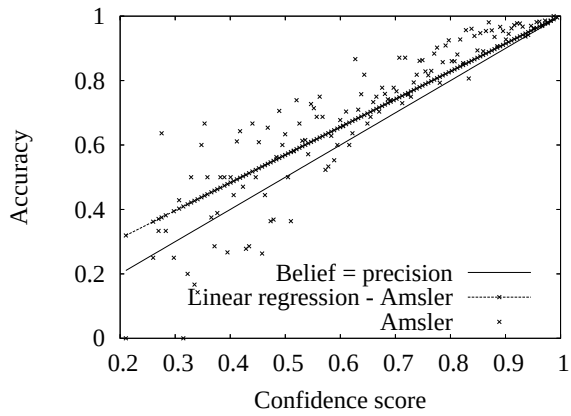
4.2.1 Reliability of Bibliometric Classifiers

Recalling Section 3.2.1, an ideally reliable classifier provides confidence scores proportional to its accuracy. In spite of not being ideal classifier, kNN classifiers using bibliometric measures present the property of providing confidence scores proportional to their accuracy in ACM8, Cade12 and Wiki8 collections. Figures 4.1(a), 4.1(b) and 4.1(c) show the accuracy values obtained for confidence scores estimated by kNN using the Amsler similarity measure. We chose Amsler-based kNN because it was the classifier with best micro F_1 values over all bibliometric classifiers (loosing only for the co-citation-based SVM in the Cade12 collection). Also, its macro F_1 values are only slight inferior to some of the other bibliometric classifiers. In all figures, the dashed lines are derived by linear regression applied over the (confidence score, accuracy) points for a classifier, and the solid lines correspond to an ideal classifier for which the confidence score would correspond exactly to the accuracy obtained, as explained in Section 3.2.1.

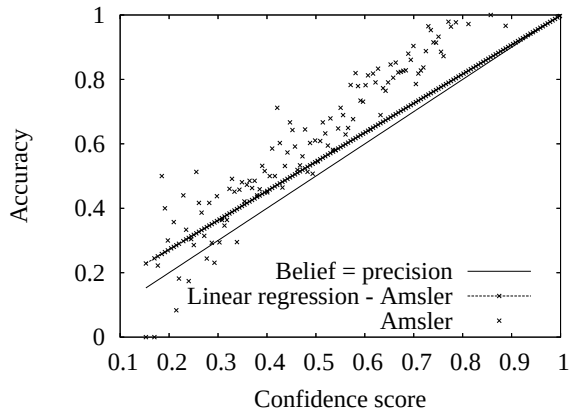
We note that in the first three graphics the regression lines are very similar to the line representing the reliability of an ideal classifier. This implies that, in general, the values provided as confidence scores approximately correspond to the accuracy obtained by the classifier. Thus, we can take these values as good estimates of how many documents will be assigned to the correct classes. Similar figures were obtained for the other kNN classifiers using the other similarity measures in the three collections, which we do not include here to avoid repetition of arguments. The only exception occurs with kNN based on bib-coupling measure in the Cade12 collection, where the regression line clearly differs from the ideal line, as shown in Figure 4.1(d). This occurs because there are only few documents that have bib-coupling similarity to some other document in the collection. as discussed in Section 4.1.

4.2.2 Combining the Results of Bibliometric and Textual Classifiers

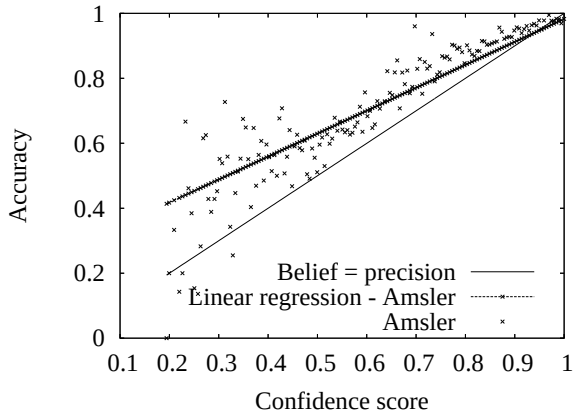
In this section we report the experiments with the two combination methods described in Section 3.2. We applied the reliability-based combination method to the three collections we studied. In each collection, we used the kNN classifier based on the Amsler similarity as the first classifier to be combined. As the second classifier, we used the text-based classifier that performed better in each collection. Figure 4.2 shows the regression lines obtained by applying the algorithm of Figure 3.1 to each collection. The two lines in each graphic



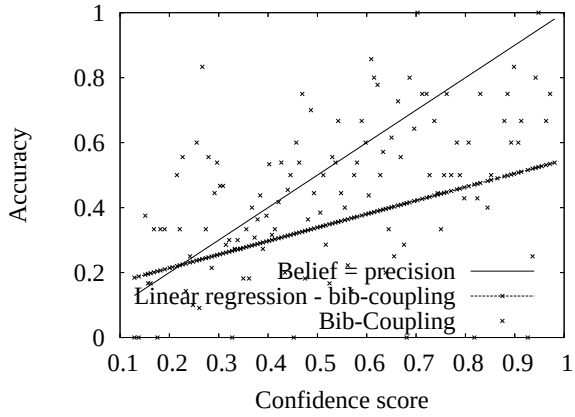
(a) Regression line for Amsler in ACM8.



(b) Regression for Amsler in Cade12



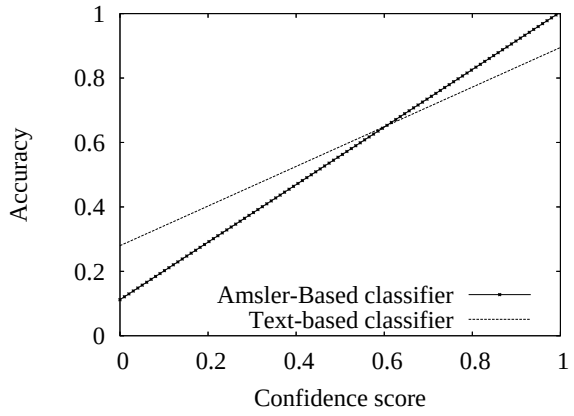
(c) Regression for Amsler in Wiki8.



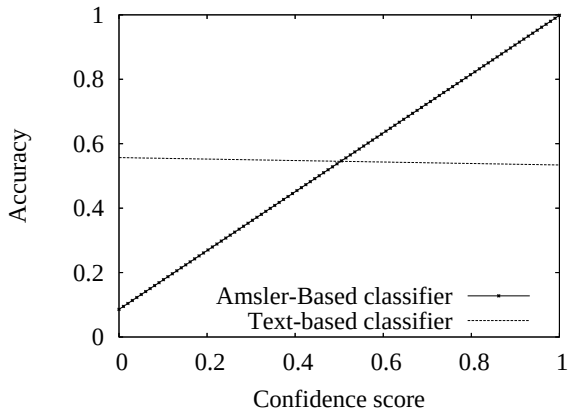
(d) Regression for Bib-coupling in Cade12.

Figure 4.1: Accuracy per confidence score. Graphics (a), (b) and (c) show the regression line for the Amsler-based kNN classifier in ACM8, Cade12 and Wiki8 collections, respectively. Graphic (d) shows the regression line for bib-coupling-based kNN in Cade12.

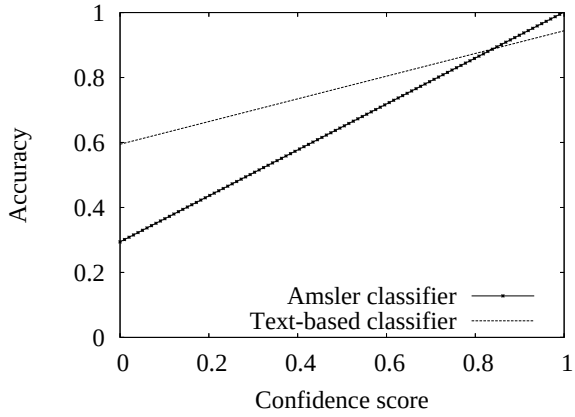
correspond to the lines computed by the linear functions derived from lines 9 and 11 of the algorithm. For all the three collections, the classifiers based on the Amsler measure were used as guides since they are more reliable than text-based classifiers.



(a) ACM8



(b) Cade12



(c) Wiki8

Figure 4.2: Regression lines for confidence scores of Amsler-based kNN classifier and for confidence scores of TF-IDF-based SVM classifier in the three collections.

For the ACM collection we can see that the regression lines for the Amsler-based classifier and the best text classifier are very similar. This means that, in general, for any confidence scores of the Amsler classifier the effectiveness of both Amsler-based and text-based classifiers are very similar. Consequently, the reliability-based combination, in this case, is not able to present much gain over the Amsler classifier as will be seen later in this section.

As Figure 4.2(b) shows, the regression line for the text-based classifier for the Cade12 collection is close to a constant function. This is a consequence of the poor quality of text information in Cade12. On the other hand the regression line for the kNN based on Amsler measure is very similar to an ideal classifier. So the confidence score of the Amsler-based classifier can be used to drive the combination of results. Both lines cross each other at point p corresponding to a confidence score of 50%. If the confidence score of the Amsler-based classifier falls at p or above, the class it indicates is adopted. If this is not the case, the class pointed by the text-based classifier is preferred. Contrary to what happened in ACM8, reliability-based combination in this case is expected to perform better than text and Amsler-based classifiers considered in isolation.

The regression lines for the Wiki8 collection are shown in Figure 4.2(c). As can be seen, the kNN classifier using the Amsler measure is also more reliable than the text-based classifier for this collection, since its confidence score values are similar to its accuracy values. However, the regression lines of the two classifiers cross each other for values of confidence score superior to 0.85. This means that the link-combination method will take the output of the text-based classifiers for most of the confidence scores. Also, the accuracies for both classifiers are similar for confidence scores superior to 0.85, thus the reliability-based combination for this collection is expected to perform only slightly better than text-based classifier alone.

The results of the reliability-based combination strategy for the three collections are listed in Table 4.6. For comparison purposes, the results obtained by text-based and Amsler-based classifiers taken in isolation are also shown in the table.

Table 4.6 also shows the results for the *Bayesian combination* method described in Section 3.2.2. This method was used in [9] with a subset of the Cadê collection slightly different from the Cade12 and presented good improvements over text-based classifiers and bibliometric classifiers in isolation. Thus, we use this method as the baseline for comparison with the reliability-based. For each collection, We tuned the weights W_t and W_b in Equation3.4 by splitting the training set in two disjoint sets: a training subset and a validation subset [48]. We used the training subset to train bibliometric and text-based classifiers and used the validation subset to test the classifier. The results of classifiers with the validation documents were extensively combined, by varying values of W_t and W_b until the best values were found that maximise F_1 measure of the combination process. Once the values of the two weights were found we retrained the bibliometric and text-based classifiers, this time using the entire training set and evaluate the combination process

Collection	Methods	$micF_1$	$macF_1$	Gains over link classifier (%)	
				$micF_1$	$macF_1$
ACM8	kNN-Amsler	84.43	79.41	–	–
	SVM-TFIDF	80.37	75.97	-4.8	-4.33
	Bayesian Comb.	87.04	82.76	3.0	4.2
	Reliability Comb.	85.76	81.37	1.6	2.46
Cade12	kNN-Amsler	68.56	75.53	–	–
	SVM-TFIDF	54.18	49.38	-20.97	-34.62
	Bayesian Comb.	76.51	79.29	11.6	4.97
	Reliability Comb.	78.04	80.39	13.82	6.43
Wiki8	SVM-TFIDF	88.27	82.99	–	–
	kNN-Amsler	87.15	82.05	-1.26	-1.13
	Bayesian Comb.	90.75	87.28	2.8	5.16
	Reliability Comb.	90.44	86.44	2.45	4.15

Table 4.6: Macro-averaged and micro-average F_1 results for combining approaches in the ACM8, Cade12 and Wiki8 collections.

using documents of the test set. This process was repeated for each round of the ten-fold cross-validation process, so that we obtained values of W_t and W_b in each round.

As we can see in Table 4.6, the reliability-based combination method presented results inferior to the Bayesian method for the ACM8 and Wiki8 collections. As we stated before, for the case of the ACM8, the regression lines of both text and Amsler-based classifiers are similar to each other. Thus, reliability-based combination could not improve much by using the output of text classifier for confidence scores of the Amsler classifier smaller than about 0.7, which is the confidence score where the two lines cross each other.

For Wiki8, the accuracy of the text-based classifier is superior to the Amsler-based classifier for all confidence scores inferior to about 0.85. Also, the accuracy above this point is very similar for both classifiers, thus reliability-based combination could not improve much the result by choosing the output of the Amsler classifier for confidence scores superior to 0.85.

The gains obtained from any combination strategies seem at first sight quite small both in the ACM8 and Wiki8 collections. However, suppose we had a perfect combination method that would be able to choose between the two classifiers the one which assigned the

right class, whenever one of them gives a right assignment. The $macF_1$ and $micF_1$ average values for such perfect combiner can be obtained with 10-fold cross validation, using the same folds that were used in all the other experiments for each collection.

When comparing this perfect classifier to the results obtained on each collection, we realise that the possible improvements in results are not so high. For the ACM8 collection, this perfect combiner would achieve $micF_1$ average value of 91.24% and $macF_1$ average value of 88.50%, which correspond to gains of only 4.8% in $micF_1$ and 9.6% in $macF_1$ over the Bayesian method, which is our best method. Also, for the Wiki8 collection, the perfect combiner would achieve values of 94.22% for $micF_1$ and 92.32% for $macF_1$, which correspond to gains of 3.8% and of $5,7\%$, respectively, over the Bayesian method. A perfect combiner for Cade12 would achieve 83.99% for $micF_1$ and 86.58% for $macF_1$, which correspond to gains of 7.62% and of 7.7%, respectively, over the reliability-based combination method. The results of the perfect combiner correspond to the superior limits for combination of results of classifiers. As we can see, there is room for enhancement, but the possible gains over the ones obtained tend to be small, considering the optimal case.

4.3 Further Understanding the Classification Failures

In this Section we investigate the possible reasons for the classification failures produced by the bibliometric classifiers. We performed two types of study to evaluate the origins and meaning of the failures produced. First, we use information available in the ACM8 collection to study the failures that are consequence of documents containing multiple classes. Second, we perform a more comprehensive study with users to understand the failures occurred in the three experimented collections.

Possible Multilabel Classification Cases

Since kNN using the Amsler similarity measure was the best bibliometric classifier, we decide to further investigate its cases of misclassification. We found that in 58% of the failures, the class assigned by the documents' authors appears as the second most probable class assigned by the classifier. Although all documents of ACM8 were assigned to only one first level class of the ACM hierarchy by their authors, we intended to investigate if some of the above cases could be considered correct in a multilabel classification setting, as follows.

In the ACM computing classification system tree (CCST) [1], associations between classes are declared explicitly. For instance, Figure 4.3 shows an entry in CCST describing the sub-class *I.7 - Document and Text Processing*. The labels appearing on the right hand side of the sub-class title (*H.4* and *H.5*) indicate that a document classified under sub-class *I.7* is also related to sub-classes *H.4* and *H.5*. As a consequence, a document classified under class *I.7* (or its subclasses) might be also classified under classes *H.4* and *H.5* (or its subclasses).

- I.7 DOCUMENT AND TEXT PROCESSING [Revised] (H.4, H.5)
  - I.7.0 General
  - I.7.1 Document and Text Editing [Revised]
    - *Document Management* NEW!
    - *Languages***
    - *Spelling***
    - *Version control* NEW!
  - I.7.2 Document Preparation
    - *Desktop Publishing*
    - *Format and Notation*
    - *Hypertext/hypermedia*
    - *Index generation* NEW!
    - *Languages and systems*
    - *Markup languages* NEW!
    - *Multi/mixed media*
    - *Photocomposition/typesetting*
    - *Scripting languages* NEW!
    - *Standards*
  - I.7.3 Index Generation**

Figure 4.3: Part of the ACM classification tree showing relations among sub-classes of different first-level classes.

To find the proportion of misclassification cases that could be considered correct assignments in a multilabel classification setting, we have to determine the misclassified documents which could be assigned to multiple classes, among them, the one chosen by

the kNN classifier. Thus, given a test document d_t for which the kNN classifier failed, let L_{kNN} be the list of the sub-classes of the k most similar documents to d_t and let L_{auth} be the list of sub-classes of d_t assigned by its authors. By inspecting both lists, we can find pairs of sub-classes (c_i,c_j), where $c_i \in L_{kNN}$ and $c_j \in L_{auth}$, such that c_i and c_j or some of their ancestors are explicitly related in the ACM hierarchy. Once we find these pairs, we select c_i as a potential class of d_t if its first-level ancestor was assigned by the kNN classifier and its occurrence count in L_{kNN} is greater than a certain threshold f, determined experimentally after sampling some documents. In our experiments, we used $f = 3$.

Table 4.7 shows an example of a misclassified document that was assigned to sub-class I.7.2 ($L_{auth} = \{I.7.2\}$ in this case) by its author. The second column shows L_{kNN}, the sub-classes of the k nearest neighbours of this document. The numbers in parentheses correspond to the occurrence of each sub-class. The subclasses in bold face occurred more than three times in L_{kNN} and have as ancestor class $H.5$ that is related to sub-class $I.7$. This class, by its turn, is ancestor of the class assigned by the author of the document (see Figure 4.3). Thus, if the kNN classifier assigns class H to this document, this should be considered a correct decision in a multilabel classification setting.

Author assigned	Sub-classes in k most similar documents
I.7.2	**H.5.1** (18), **H.5.2** (10), H.2.4 (7), H.2.1 (5), **H.5.3** (5), **H.5.4** (5), H.3.1 (4), H.2.8 (3), H.3.4 (3), H.3.3 (3), H.1.2 (3), H.2.3 (2), C.0 (1), H.3.7 (1), H.4.3 (1), D.2.13 (1), D.2.6 (1), H.3.2 (1), H.3.5 (1)

Table 4.7: Example of the detection of a candidate for multilabel classification.

Table 4.8 summarises the cases of misclassifications that could be considered correct decisions in a multilabel classification setting. For obtaining these data, we used kNN with Amsler similarity. Additionally, to confirm that the document could be really considered as pertaining to both classes, we manually checked them.

The second column of Table 4.8 contains the total of misclassified documents per class. Third column contains the number of failures that were considered multilabel classification cases. The fourth column contains the percentages of these cases. As we can see, 24% of the misclassifications should be considered correct decisions if we had used multilabel classification.

Class	kNN Failures		
		Multilabel classification	
	Total failures	Total	%
B	123	43	34.95
C	168	50	29.76
D	175	55	42.8
F	159	34	31.42
G	71	12	16.9
H	97	18	18.56
I	107	19	17.76
K	72	2	2.78
Average		29.12	24.36

Table 4.8: The number of kNN classification failures by class and the number and percentage of these failures that can be considered multilabel classification cases.

User Study

Motivated by the difficulty in improving classification results in the ACM8 and Wiki8 collections by meanas of our combination methods, as well by the failures of the bibliometric-based kNN classifier in the Cade12 collection even when bibliometric relations exist, we decided to perform a user study using the cases that our classifiers did not succeed in providing a correct class to a test document.

When a bibliometric-based kNN classifier assigns a wrong class to a test document it does so because the parents or children of the test document are more linked to training documents of the wrong class than to training documents of the correct class. Since links are an explicit indication from an author that his work is somehow related to another one, we suspected that even humans would have difficult to classify documents that had a wrong class assigned by the bibliometric classifiers.

To test this assumption we conducted another experiment in order to study human classification of those unsuccessful cases. We removed all the no-information documents in the collections since our objective was to study only the cases the classifiers have failed because the bibliometric information led them to fail. We then repeated the classification using kNN classifier with the Amsler similarity and ten fold cross-validation with each of the three new collections. For each collection, we grouped the classifier results by the corresponding categories, such that each class could be considered as a stratum from where

we derived the samples. We considered in each class the proportions of hits and failures of the classifier, and we computed the sizes of the samples to be classified by people using Equation 4.1, derived in [17] for computing the sample size using proportions:

$$n_c = \frac{\frac{t^2 PQ}{d^2}}{1 + \frac{1}{N_c}(\frac{t^2 PQ}{d^2} - 1)} \tag{4.1}$$

where n_c is the size of the sample for class c, N_c is the total number of documents of class c, P is the fraction of N_c that the were misclassified by kNN, $Q = 1 - P$ is the fraction of N_c that were correctly classified, d is the size of the error interval and t is the abscissa of the normal curve that cuts off an area α at the tails of the curve. In our experiment we used $\alpha = 0.05$, thus, $t = 1.96$ and $d = 0.05$.

Given the large number of subjects necessary to classify the documents, and since we are mostly interested in failures of the classifier, we decided to evaluate only classifier failures. Thus, we reduced the size n_c of the samples by using only $n_c * P$ elements that correspond to the proportion of misclassified documents of the sample. Once randomly obtained the samples for each class of a given collection, we joined all the class samples forming a unique sample with documents of distinct classes that subjects were asked to classify. We obtained 214 documents for the ACM8 collection, 323 for the Cade12 and 82 documents for the Wiki8 collection.

For each collection sample we generated a replica of each document and randomly distributed the resulting duplicated sample in pools, in a way that each document would be evaluated by two distinct human classifiers. We assigned a pool to each subject. We generated 20 pools for Cade12 and ACM8 and 10 pools were generated for the Wiki8 collection. We assigned each pool of a same collection to a distinct subject. The 20 pools of ACM8 were evaluated only by computer science graduate students, since expertise in this subject area was required. The pools for for Cade12 and Wiki8 were evaluated by graduate and undergraduate students.

For each document, a person had to choose one among four options. The first two options were two classes: the correct one and the one assigned by the classifier. The order of presentation of correct and wrong classes was randomly changed among the documents. The third option was to choose both classes and the fourth option was to choose none of the classes.

For the ACM8 and Cade12 collections, subjects had access to much more information than the automatic classifier had. Besides link information, in the case of the ACM8 sample, people could analyse the title, authors, keywords, abstract (when available), the

conference name, and the links' text. Evaluators of the Cade12 collection sample had access to the full page (which included images and photos). For the Wiki8 collection, however, the only information available was the raw text of each document. Further, people had the advantage of deciding between two classes only, for a given document, in contrast to the automatic classifier that had to chose one among all the possible classes.

Table 4.9 summarises the classifications made by subjects for the three collections. It shows the percentages of occurrence of each option among the classifications. Note that in all the collections the percentage of correct classification is low, not reaching even 45%. Most of the human classifications show doubt or disagreement with the correct class. This confirms our expectation about the difficulty of classifying the sampled documents.

Human classification	ACM8	Cade12	Wiki8
correct	38.31%	43.34%	41.46%
wrong	28.97%	19.50%	12.80%
marked both classes	20.79%	28.94%	40.24%
marked none of the two classes	11.91%	8.20%	5.48%

Table 4.9: Results of classifications made by subjects.

We also collected some statistics about the documents that were evaluated, shown in Table 4.10. The experiment shows the consensus among the subjects. For instance, if we sum the values of each column of Table 4.10, we have that 42.5% of the documents in the ACM8 sample and 53.5% of the documents in Cade12 samples received the same evaluation by the two subjects. Conversely, consensus is high in Wiki8, where 71.86% of the document received the same evaluation from the two subjects. However, the number of documents for which consensus was achieved for the two-class opinion is almost the same of those that consensus about the correct class. Finally, in the three collections the number of documents that were assigned the correct class by the two subjects is small. This emphasises the difficulty of these documents that lead the classifier to fail.

We also investigated users' opinion for the documents that we denominate *hard decisions* of the classifier. Hard decisions correspond to misclassified documents for which the classifier assign the correct class as the second choice and the probability difference between the first and second choices was very small (less or equal to 0.2 in our experiments). The second line of Table 4.11 shows that the majority of the documents that are hard decision cases were misclassified or received a two-class vote by at least one human evaluator. Also, only a few hard decision cases were correctly classified by all subjects. Thus hard decision

% of documents classified:	ACM8	Cade12	Wiki8
correctly by all subjects	21.49	28.79	34.14
wrongly by all subjects	13.55	9.29	4.87
as both by all subjects	5.14	12.07	30.48
as not belonging to any by all subjects	2.33	3.40	2.44

Table 4.10: Percentage of documents that reached consensus by the two human classifiers, in three collections.

cases are really very difficult even for human classification.

The above results and observations tend to indicate that the failures of the classifier based on bibliometric measures are really difficult cases. Even human classification, using much more information, did not achieve much success. Further, consensus on the correct class is very rare among human evaluators and the doubt cases for the classifier are even harder ones to correctly classify.

	ACM8	Cade12	Wiki8
% of documents that are doubt cases	13.08	23.83	26.83
% of doubts wrongly classified or that received 2 classes	71.4	72.72	54.54
% of doubts correctly classified by all subjects	25.0	19.48	31.81

Table 4.11: Human classification of documents that were doubt cases.

Chapter 5

Conclusions and Future Work

In this work, we study about the use of classifiers based on bibliometric similarity measures for classifying Web collections. We use three bibliometric measures: co-citation, bibliographic coupling and Amsler. For each bibliometric measure we derived kNN and SVM based classifiers and conducted experiments training these classifiers over three important, but very distinct collections of documents found in the Web: a directory of Web pages, a digital library of scientific articles and a sub-collection of an encyclopedia.

We compared the effectiveness of bibliometric classifiers and text-based classifiers. Experiments have shown that bibliometric classifiers performed better than text classifiers in two of the collections studied and presented results only marginally inferior to text-based classifier in the collection derived from the encyclopedia.

The experiments allowed us to reach important conclusions about the circumstances where bibliometric measures are effective for classifying documents. The first conclusion is that bibliometric classifiers are strongly affected by the distribution of the co-occurrence of parent and children documents over the collection. This is a consequence of the fact that a bibliometric similarity measure between any two documents is the computation of the number of children or parent both documents have in common. Thus, there are two necessary conditions for the existence of some bibliometric information for a given document d_1: (a) document d_1 must have at least one in-link, a parent, or at least one out-link, a child; (b) there must be at least another document d_2 that shares with d_1 a common in-linked (parent) or out-linked (child) document. Obviously, condition (a) is a prerequisite for condition (b).

For example, in Cade12, the distribution of links is in a such a way that the great majority of the documents does not have parents in common with any other document and

about 70% of the documents do not have children in common with any other document. Thus, bibliometric classifiers, although being superior to text classifiers in this collection, have a limited accuracy dictated by the lack of information.

Given the first conclusion, we have that, for a new collection, it is easy to predict that a certain kind of bibliometric measure must not be used if many documents in the collection do not have one kind of link. For instance, if out-links are rare, as is the case of Cade12, classifiers based on bibliographic coupling certainly will not perform well. However, when most of documents have at least one kind of links (in-links or out-links) we need to, verify if they have parent or children in common with the other documents in the collection and this imply in computing at least the Amsler measure for the test documents. Thus, whenever link information is present in many documents it is difficult to infer that any bibliometric measure is appropriate without computing it.

The second conclusion is that the existence of bibliometric information to classify a given test document is not sufficient. There must also be coherence between the correct class of a test document and the class of most of its neighbors relative to a given bibliometric measure. This happens because both kNN and SVM classifiers using bibliometric measures take their decision based on the neighbors of the test document. The kNN classifier considers the class that is most frequent among the k neighbors, whereas the SVM classifier, we trained, uses the test documents' neighbors as its features. Thus, if most of the neighbors belong to a distinct class, the classification of the test document will fail.

While the first conclusion is related to the existence of bibliometric information, the second one is related to the quality of this information whenever it is available. However, it is expected that most documents link to documents of the same topic or of topics related to its own topic. Thus, we hypothesize that cases where most of the documents related to a given document d have class distinct from d's class are both rare and of difficult cases. Both the rareness and the difficulty hypothesis were confirmed in all the three collections studied. The rareness hypothesis can be used to explain why bibliometric classifiers are reliable ones.

The difficulty hypothesis could be confirmed by means of a user study conducted over those cases where the classifier failed although bibliometric information was available. Most of the cases were assigned a wrong class by at least one of the human classifiers and consensus is rare among classifiers.

Given the fact that bibliometric classifiers are reliable, whenever the corresponding bibliometric measure is available, we devised a method for combining the results of biblio-

metric classifiers and text-based classifiers. In this method, we use the estimation of the reliability of each classifier to decide the one to be used in the classification of a given document. The combination method was compared to the Bayesian combination used in [9] and was shown to be better than the Bayesian method for one of the collections tested and slightly inferior in the other two collections.

Except for the Cade12 collection, where the proposed combination method achieved gains up to 13.8% over the best classifier, the combination of classifiers' results did not present significant gains. Also, the difference in effectiveness between both combination method is too tight. Thus, we investigated the effectiveness of an ideal combination method that could be able to chose the correct classifier for a given document, whenever at least one of the classifiers could classify the document correctly. For the three collection, this analysis shows that an ideal combination method could achieve gains up to 7.6% of $micF_1$ values and gains up to 9.6% of $macF_1$ over the best combination tested for each of the collections studied. This means that there is still room for improvements, but given the tight margin between the ideal combination and those investigated here, we can infer that improvement by combining classification results is hard to achieve in practice.

In summary, we conclude that bibliometric measures, whenever available, are useful for building document classifiers and most of the cases where bibliometric classifiers fail to classify were shown to be really difficult cases.

Future Work

In this work, we derived bibliometric measures directly from explicit links found among documents in collections of the Web. Our study revels some general conclusions about the effectiveness of bibliometric measures. However, we could also observe that each collection has its own specificity about the availability and distribution of link information inside it. Thus, we suggest that exploring features in specific collections may be more effective to enhance effectiveness of bibliometric classifiers in each collection. These features can be used to derive *artificial links* which may increase the number of bibliometric relations among documents. An artificial link is created between two documents to represent some kind of relation between them that is expected to be useful for the classification task. In what follows, we suggest possible derivation of artificial links in some specific collections:

1. **Deriving links from authoring information** – In a digital library of scientific papers, we may create a link between two documents that have one or more authors

in common. The intuition is that an author tend to write papers in a same topic and the more authors two papers have in common the more is the chance that they are related to the same topic.

2. **Deriving links from Web pages content and from URLs** – We suggest investigating the creation of links between documents that have two or more words in common that co-occur frequently in documents of a a same class. Other possibility is to link documents which are kept in a same directory of a site. These documents tend to be associated to similar topics and can be identified by a common prefix of their URL.

The use of artificial and explicit links together to derive bibliometric measures can be seen as a method of combining different source of evidence, because artificial links are derived from other sources of evidence not related to explicit links.

In summary, we suggest, as future work, some investigation on *link mining* in other to enrich collections with high quality links, aiming to enhance bibliometric classifiers.

As another future work, we suggest investigations on feature selection applied to links to be used in collections where bibliometric relations among documents exist in abundance. We may use some feature selection technique to eliminate parent or children documents that link to or are linked by many documents of distinct classes.

Bibliography

[1] The acm computing classification system - 1998 version. `http://www.acm.org/class/1998/ccs98.html` (visited September 20th 2007), 1998.

[2] Brian Amento, Loren Terveen, and Will Hill. Does "authority" mean quality? predicting expert quality ratings of web documents. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 296–303, Athens, Greece, July 2000.

[3] Robert Amsler. Application of citation-based automatic classification. Technical report, The University of Texas at Austin, Linguistics Research Center, December 1972.

[4] Krishna Bharat and Monica R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 104–111, Melbourne, Australia, August 1998.

[5] Julie Bichtler and Edward A. Eaton III. The combined use of bibliographic coupling and cocitation for document retrieval. *Journal of the American Society for Information Science*, 31(4):278–282, July 1980.

[6] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas. Link analysis ranking: Algorithms, theory, and experiments. *ACM Transactions on Internet Technology*, 5(1):231–297, 2005.

[7] T. Bray. Measuring the web. In *Proceedings of the 5th International World Wide Web Conference on Computer Networks and ISDN Systems*, pages 993–1005, Paris, France, 1996.

[8] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. volume 30, pages 107–117, Amsterdam, The Netherlands, The Netherlands, 1998. Elsevier Science Publishers B. V.

[9] Pável Calado, Marco Cristo, Marcos André Gonçalves, Edleno S. de Moura, Berthier Ribeiro-Neto, and Nivio Ziviani. Link-based similarity measures for the classification of web documents. *Journal of the American Society for Information Science and Technology*, 57(2):208–221, 2006.

[10] Pável Pereira Calado, E. S. de Moura, Berthier Ribeiro-Neto, Ilmério Silva, and Nivio Ziviani. Local versus global link information in the web. *ACM Transactions on Information Systems (TOIS)*, 21(1):42–63, 2003.

[11] S. Jeromy Carrière and Rick Kazman. Webquery: searching and visualizing the web through connectivity. volume 29, pages 1257–1267, Amsterdam, The Netherlands, The Netherlands, 1997. Elsevier Science Publishers B. V.

[12] Soumen Chakrabarti. *Mining the Web - Discovering Knowledge from Hypertext Data*. Morgan Kaufmann Publishers, San Francisco, CA, USA, 2003.

[13] Soumen Chakrabarti, Byron Dom, and Piotr Indyk. Enhanced hypertext categorization using hyperlinks. In *SIGMOD'98: Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, pages 307–318, Seattle, Washington, USA, 1998.

[14] Soumen Chakrabarti, Byron Dom, Prabhakar Raghavan, Sridhar Rajagopalan, David Gibson, and Jon Kleinberg. Automatic resource compilation by analyzing hyperlink structure and associated text. In *Proceedings of the 7th International World Wide Web Conference*, pages 65–74, Brisbane, Australia, April 1998.

[15] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Available at `http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf`.

[16] Hao Chen and Susan T. Dumais. Bringing order to the Web: Automatically categorizing search results. In *Proceedings of the CHI 2000 Conference on Human Factors in Computing Systems*, pages 145–152, Hague, The Netherlands, April 2000.

[17] W. G. Cochran. *Sampling Techniques*. John Wiley & Sons, second edition, 1977.

[18] David Cohn and Thomas Hofmann. The missing link - a probabilistic model of document content and hypertext connectivity. In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 430–436. MIT Press, 2001.

[19] Thierson Couto, Marco Cristo, Marcos André Gonçalves, Pável Calado, Nivio Ziviani, Edleno Moura, and Berthier Ribeiro-Neto. A comparative study of citations and links in document classification. In *JCDL'06: Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 75–84, Chapel Hill, NC, USA, June 2006.

[20] Marco Cristo, Pável Calado, Edleno Moura, and Berthier Ribeiro-Neto Nivio Ziviani. Link information as a similarity measure in web classification. In *10th Symposium On String Processing and Information Retrieval SPIRE 2003*, volume 2857 of *Lecture Notes in Computer Science*, pages 43–55, Manaus, AM, Brazil, october 2003.

[21] Jeffrey Dean and Monika R. Henzinger. Finding related pages in the World Wide Web. *Computer Networks*, 31(11–16):1467–1479, May 1999.

[22] Leo Egghe and Ronald Rousseau. *Intorductions to informetrics: quantitative methods in library, documentation and information science*. Elsevier Science Publishers, North-Holland, Amsterdam, The Netherlands, 1990.

[23] Michelle Fisher and Richard Everson. When are links useful? Experiments in text classification. In *Advances in Information Retrieval, 25th European Conference on IR Research, ECIR 2003*, Lecture Notes in Computer Science, pages 41–56, Pisa, Italy, April 2003.

[24] Johannes Furnkranz. Exploiting structural information for text classification on the WWW. In *Proceedings of the 3rd Symposium on Intelligent Data Analysis (IDA99)*, volume 1642-1999 of *Lecture Notes in Computer Science*, pages 487–497, Amsterdam, The Netherlands, August 1999. Springer Berlin – Heidelberg.

[25] Pinski G. and Narin F. Citation influence for journal aggregates of scientific publications: Theory, whith application to literature of physics. *Information Procesing & Management*, 12:297–312, 1976.

[26] Eugene Garfield. Citation analysis as a tool in journal evaluation. *Science*, 178(4060):471–479, 1972.

[27] N. Geller. On the citation influence methodology of pinski and narin. *Information Processing & Management.*, 14:93–95, 1978.

[28] David Gibson, Jon M. Kleinberg, and Prabhakar Raghavan. Inferring Web communities from link topology. In *Proceedings of the Ninth ACM Conference on Hypertext and Hypermedia: Links, Objects, Time and Space - Structure in Hypermedia Systems*, pages 225–234, Pittsburgh, PA, USA, June 1998.

[29] Eric J. Glover, Kostas Tsioutsiouliklis, Steve Lawrence, David M. Pennock, and Gary W. Flake. Using web structure for classifying and describing web pages. In *WWW'02: Proceedings of the 11th international conference on World Wide Web*, pages 562–569, Honolulu, Hawaii, USA, 2002.

[30] Xiaofeng He, Hongyuan Zha, Chris H. Q. Ding, and Horst D. Simon. Web document clustering using hyperlink structures. *Computational Statistics & Data Analysis*, 41(1):19–45, November 2002.

[31] Thorsten Joachims. Text categorization with support vector machines: learning with many relevant features. In *Proceedings of ECML-98, 10th European Conference on Machine Learning*, pages 137–142, Chemnitz, Germany, April 1998.

[32] Thorsten Joachims, Nello Cristianini, and John Shawe-Taylor. Composite kernels for hypertext categorisation. In *ICML'01: Proceedings of the Eighteenth International Conference on Machine Learning*, pages 250–257, Williamstown, MA, USA, 2001.

[33] M. M. Kessler. Bibliographic coupling between scientific papers. *American Documentation*, 14(1):10–25, January 1963.

[34] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 668–677, San Francisco, CA, USA, January 1998.

[35] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, Sep. 1999.

[36] Steve Lawrence, C. Lee Giles, and Kurt D. Bollacker. Autonomous citation matching. In Oren Etzioni, Jörg P. Müller, and Jeffrey M. Bradshaw, editors, *Proceedings of the Third Annual Conference on Autonomous Agents (AGENTS-99)*, pages 392–393, Seattle, Washington, USA, May 1–5 1999.

[37] Ronny Lempel and Shlomo Moran. Salsa: the stochastic approach for link-structure analysis. *ACM Transactions on Information Systems*, 19(2):131–160, April 2001.

[38] Tom Mitchell. *Machine Learning*. McGraw-Hill, New York, NY, USA, 1997.

[39] Hyo-Jung Oh, Sung Hyon Myaeng, and Mann-Ho Lee. A practical hypertext catergorization method using links and incrementally available class information. In *Proceedings of the 23rd annual Intl. ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 264–271, Athens, Greece, July 2000.

[40] Doreian P. A measure of standing for citation networks within a wider environment. *Information Processing & Management*, 30:21–31, 1994.

[41] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the Web. Technical report, Stanford Digital Library Technologies Project, 1998.

[42] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of plausible inference*. Morgan Kaufmann Publishers, San Francisco, CA, USA, 2nd edition, 1988.

[43] Berthier Ribeiro-Neto and Richard Muntz. A belief network model for IR. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 253–260, Zurich, Switzerland, August 1996.

[44] Berthier Ribeiro-Neto, Ilmério Silva, and Richard Muntz. *Soft Computing in Information Retrieval: Techniques and Applications*, chapter 11—Bayesian Network Models for IR, pages 259–291. Springer Verlag, 2000.

[45] Gerard Salton. Associative document retrieval techniques using bibliographic information. *Journal of the ACM*, 10(4):440–457, October 1963.

[46] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5):513–523, 1988.

[47] Gerard Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, NY, USA, 1983.

[48] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, March 2002.

[49] Ilmério Silva, Berthier Ribeiro-Neto, Pável Calado, Edleno Moura, and Nívio Ziviani. Link-based and content-based evidential information in a belief network model. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 96–103, Athens, Greece, July 2000.

[50] Henry G. Small. Co-citation in the scientific literature: A new measure of relationship between two documents. *Journal of the American Society for Information Science*, 24(4):265–269, July 1973.

[51] Henry G. Small and Michael E. D. Koenig. Journal clustering using a bibliographic coupling method. *Information Processing & Management*, 13(5):277–288, 1977.

[52] Aixin Sun, Ee-Peng Lim, and Wee-Keong Ng. Web classification using support vector machine. In *Proceedings of the Fourth International Workshop on Web Information and Data Management*, pages 96–99, McLean, Virginia, USA, November 2002.

[53] Loren Terveen, Will Hill, and Brian Amento. Constructing, organizing, and visualizing collections of topically related Web resources. *ACM Transactions on Computer-Human Interaction*, 6(1):67–94, March 1999.

[54] Howard Turtle and W. Bruce Croft. Inference networks for document retrieval. In *Proceedings of the 13th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1–24, Brussels, Belgium, September 1990.

[55] T. Upstill, N. Craswell, and D. Hawking. Query-independent evidence in home page finding. *ACM Transactions on Information Systems*, 21(3):286–313, 2003.

[56] V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Velag, New York, NY, USA, 1995.

[57] T. Westerveld, W. Kraaij, and D. Hiemstra. Retrieving Web pages using content, links, URLs and anchors. In *10th Text Retrieval Conference*, pages 663–672, Gaithersburg, Maryland, USA, November 2001.

[58] Ian H. Witten, Alistair Moffat, and Timothy C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann Publishers, 2nd edition, 1999.

[59] Yiming Yang. Expert network: Effective and efficient learning from human decisions in text categorization and retrieval. In *Proceedings of the 17rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 13–22, Dublin, Ireland, July 1994.

[60] Yiming Yang and Xin Liu. A re-examination of text categorization methods. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 42–49, Berkeley, CA, USA, August 1999.

[61] Yiming Yang, Seán Slattery, and Rayid Ghani. A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems*, 18(2):219–241, March 2002.