Pável Pereira Calado

Orientador - Professor Berthier Ribeiro-Neto

Utilização da Estrutura de Ligações da Web em Problemas de Recuperação de Informação

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

Belo Horizonte

3 de Março de 2004

À Lu

À família

Aos amigos

À vida

...

Uma mosca sem valor,
Pousa co'a mesma alegria
Na careca de um doutor
Como em qualquer porcaria

António Aleixo

Agradecimentos

No principio, se deixado à lei da inércia, a opinião cresceria neutra, o interesse seria mesquinho, a visão curta, a vontade fraca e crenças e futilidades guiariam o meu caminho. Um grupo de pessoas impediu que tal acontecesse.

Em época de atirar pedras às janelas e de roubar cigarros, um momento de distracção levar-me-ia a fugas desastradas, saltos suicidas, choques de frente e de surpresa, espancamentos injustos, cuspo, raiva vã, olhos inchados e ignorância. Um grupo de pessoas impediu que tal acontecesse.

Ao decidir o valor das próprias decisões, uma escolha mal pesada e pilhas de trabalho inútil acumular-se-iam à minha frente, gente de cariz neutro e aborrecido atravessaria o meu caminho, os jantares seriam sóbrios e o caminho para o barbeiro, para a mercearia e para o café seriam uma parte importante da minha memória. Um grupo de pessoas impediu que tal acontecesse.

Quando tomei conta de mim mesmo, passos mal dados trariam cansaço, frustração, desistência e medo, os dias seriam passados em casa, a vida seria um tubo de ensaio, a conversa seria pobre e a experiência limitada. Um grupo de pessoas impediu que tal acontecesse.

Durante todo este percurso, um mundo inteiro girou à minha volta. Gente trabalhou, martelou, pregou, soldou, plantou, escreveu, cantou, gritou, apanhou chuva, apanhou sol, cansou-se, dormiu, acordou e voltou a trabalhar.

Alguns fizeram os meus sapatos, outros o meu computador. Alguns conduziram o autocarro, outros varreram o chão. Alguns vivem no Brasil, outros em Portugal, outros na China, outros na Tailândia.

E agora, que sou doutor, como vos posso agradecer?

Abstract

The popularity and growth of the World Wide Web offer a unique opportunity for large scale experimentation. This has greatly affected research in several scientific areas, most notably, in Information Retrieval (IR). For instance, among the many new techniques created in this area, link analysis has been a strong focus of attention. The main reason is that information from links between Web pages can be used to improve the quality of search engine results.

In this work, we study how information derived from the links between Web pages can be used to solve two distinct problems: (a) document ranking and (b) document classification. To represent both problems, formal models, based on Bayesian networks, are proposed. These models are validated through experiments performed on a collection extracted from the Web. Results show that, in fact, links between Web pages are an important source of evidence, both to rank and to classify Web documents. In both cases, combining link information with information from the content of Web pages yields better results than the use of each source of evidence in isolation. In the document ranking problem, information from Web links produces results of high precision on the top of the ranking. In the classification task, links between the pages were shown to be a more reliable source of information than the content of the pages being classified.

Resumo

A popularidade e o crescimento da *World Wide Web* oferecem uma oportunidade única para a experimentação em larga escala, o que tem afetado de sobremaneira a pesquisa em várias áreas do conhecimento, particularmente, a área de Recuperação de Informação (RI). Por exemplo, entre as muitas novas técnicas criadas no contexto da Web, análise de ligações (*links*) é uma que tem atraído grande atenção. A razão é que informação sobre as ligações entre páginas Web pode ser usada para melhorar a qualidade das respostas de uma consulta do usuário.

Neste trabalho, estudamos como ligações entre páginas Web podem ser aplicadas na resolução de dois problemas distintos: (a) ordenação de respostas a uma consulta e (b) classificação de documentos da Web. Para isso, modelos formais baseados em redes Bayesianas são propostos. Estes modelos são validados através de testes executados numa coleção extraída da Web brasileira. Os resultados mostram que, efetivamente, ligações entre páginas Web são uma fonte de evidência importante, tanto para ordenar como para classificar documentos. Em ambos os casos, combinação de informação de ligações entre páginas Web com informação sobre o conteúdo das páginas produz resultados melhores do que aqueles obtidos com o uso de cada fonte de evidência isoladamente. Para o problema de ordenação das respostas, informação sobre as ligações entre páginas Web produz resultados de alta precisão no topo do conjunto ordenado de documentos. Na tarefa de classificação, as ligações entre as páginas demonstraram ser uma fonte de evidência mais confiável que o próprio texto dos documentos.

Lista de Publicações

Artigos publicados durante o doutorado:

Capítulos de Livro

1. Altigran Silva, Pável Calado, Rodrigo Vieira, Alberto Laender, Berthier Ribeiro-Neto. Keyword-based Queries over Web Databases. *In Becker, S. A. (Org.) Effective Databases for Text & Document Management*, p.74-92, 2003.

Artigos em Periódicos

1. Tatiana Coelho, Pável Calado, Lamarque Souza, Berthier Ribeiro-Neto. Image Retrieval Using Multiple Evidence Ranking. *IEEE Transactions on Knowledge and Data Engineering*, aceite para publicação.

2. Marco Cristo, Pável Calado, Maria de Lourdes da Silveira, Ilmério Silva, Richard Muntz, Berthier Ribeiro-Neto. Bayesian Belief Networks for IR. *International Journal Of Approximate Reasoning*, v.34, n.2–3, p.163–179, November 2003.

3. Pável Calado, Berthier Ribeiro-Neto. An Information Retrieval Approach for Approximate Queries. *IEEE Transactions on Knowledge and Data Engineering*, v.15, n.1, p.236-239, January/February 2003.

4. Pável Calado, Berthier Ribeiro-Neto, Nivio Ziviani, Edleno Moura, Ilmério Silva. Local Versus Global Link Information in the Web. *ACM Transactions On Information Systems*, v.21, n.1, p.42-63, January 2003.

Artigos em Conferências

1. Pável Calado, Marco Cristo, Edleno Moura, Nivio Ziviani, Berthier Ribeiro-Neto, Marcos Gonçalves. Combining Link-Based and Content-Based Methods for Web Document Classification. *Proceedings of the 12th International Conference on Information and Knowledge Management CIKM 2003*, p.394-401, November 2003.

2. Marco Cristo, Pável Calado, Edleno Moura, Nivio Ziviani, Berthier Ribeiro-Neto. Link Information as a Similarity Measure in Web Classification. *10th Symposium On String Processing and Information Retrieval SPIRE 2003*, p.43–55, October 2003.

3. Pável Calado, Altigran Silva, Marcos Gonçalves, Berthier Ribeiro-Neto, Alberto Laender, Edward Fox, Juliano Lage, Davi Reis, Pablo Roberto, Monique Vieira. The Web-DL Environment for Building Digital Libraries from the Web. *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries JCDL 2003*, p.346-357, May 2003.

4. Pável Calado, Altigran Silva, Rodrigo Vieira, Alberto Laender, Berthier Ribeiro-Neto. Searching Web Databases By Structuring Keyword-Based Queries. *Proceedings of the 11th International Conference on Information and Knowledge Management CIKM 2002*, p.26-33, November 2002.

5. Pável Calado, Altigran Silva, Marcos Gonçalves, Berthier Ribeiro-Neto, Alberto Laender, Edward Fox, Juliano Lage, Davi Reis, Pablo Roberto,

Monique Vieira. Web-DL: An Experience In Building Digital Libraries From The Web. *Proceedings of the 11th International Conference on Information and Knowledge Management CIKM 2002*, p.675-677, November 2002.

6. Rodrigo Vieira, Pável Calado, Altigran Silva, Alberto Laender, Berthier Ribeiro-Neto. Consultando Bancos de Dados através da Web usando Palavras-chave. *Annals of the Brazilian Symposium on Databases SBBD 2002*, p.194-208, October 2002.

7. Rodrigo Vieira, Pável Calado, Altigran Silva, Alberto Laender, Berthier Ribeiro-Neto. Structuring Keyword-Based Queries for Web Databases. *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries JCDL 2002*, p.94-95, June 2002.

8. Ilmério Silva, Berthier Ribeiro-Neto, Pável Calado, Edleno Moura, Nivio Ziviani. Link-Based and Content-Based Evidential Information Retrieval in a Belief Network Model. *Proceedings of the 23rd Annual ACM SIGIR Conference on Research and Development in Information Retrieval*, p.96-103, July 2000.

Resumo Estendido

Introdução

A *World Wide Web*, devido à sua enorme popularidade e rápido crescimento, tornou necessária a criação de técnicas de Recuperação de Informação capazes de auxiliar usuários leigos a encontrar informação num tão grande e pouco estruturado repositório. Neste trabalho, a informação inerente à estrutura de ligações entre páginas Web é explorada e aplicada em tais técnicas. Novos modelos são propostos, capazes de usar a informação de ligações para tarefas como ordenação e categorização de documentos na Web.

A informação contida na estrutura hipertextual da Web pode ser usada de dois modos principais: (1) como uma medida de importância das páginas e (2) para determinar conjuntos de páginas sobre um mesmo tema. O primeiro caso segue do fato intuitivo de que quando uma determinada página é apontada por muitas outras podemos considerá-la uma fonte de informação importante. No segundo caso, as ligações entre páginas revelam uma relação entre o tópico por elas abordado, uma vez que é natural que o autor de uma página crie ligações para páginas sobre o mesmo tópico.

Estes dois tipos de informação, implícitos na estrutura de ligações da Web, podem ser usados de diversas maneiras, algumas das quais são abordadas neste trabalho. Em primeiro lugar, uma medida de importância das

páginas é combinada com o seu conteúdo textual para conseguir melhorias na ordenação de documentos. Em segundo lugar, medidas de similaridade extraídas das ligações entre páginas são usadas para melhorar o resultado de classificadores tradicionais.

Ordenação de Documentos usando Informação de Ligações

Para a combinar informação textual e de ligações para a tarefa de ordenação de documentos é proposta uma extensão do modelo de redes Bayesianas apresentado em [61]. A rede resultante é mostrada na Figura 1. Nesta rede, os nós K_i representam os termos na coleção, os nós Dc_j representam a informação textual associada a um documento d_j, os nós Da_j e Dh_j representam a informação de ligações associada a um documento d_j, o nó Q representa a consulta do usuário e os nós D_j representam a combinação final de evidências associada a um documento d_j. A informação de ligações representada pelos nós Da_j e Dh_j é obtida aplicando o algoritmo HITS, descrito em [34, 48]. A cada nó corresponde uma variável aleatória binária, que toma o valor 1 para indicar que a informação correspondente será usada na computação da ordenação dos documentos.

A ordenação de documentos é obtida calculando a probabilidade de cada documento d_j ser relevante para a consulta do usuário. Esta probabilidade é representada por $P(D_j = 1|Q = 1)$ e é definida pela equação[1]:

$$P(d_j|\mathbf{k}) \;=\; 1 - (1 - P(dc_j|\mathbf{k})) \times (1 - P(dh_j|\mathbf{k})) \times (1 - P(da_j|\mathbf{k})) \quad (1)$$

[1]Para simplificar a notação, $P(X = 1)$ é representado como $P(x)$ e $P(X = 0)$ é representado como $P(\overline{x})$.

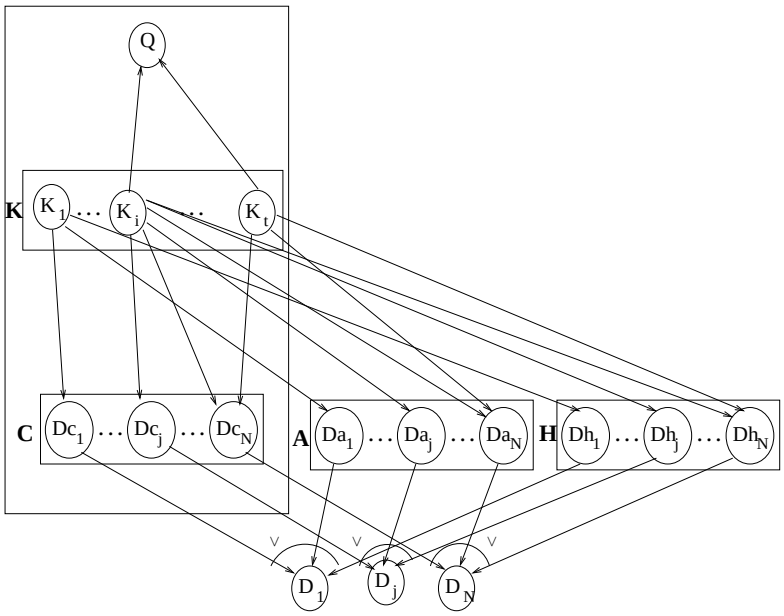


Figura 1: Modelo de combinação de informação para ordenação de documentos Web.

onde **k** representa o estado dos nós K_i em que apenas os nós correspondentes aos termos da consulta estão ativos. O valor da probabilidade $P(dc_j|\mathbf{k})$ é dado pela similaridade vetorial [66] do texto do documento d_j com a consulta do usuário. Os valores das probabilidades $P(dh_j|\mathbf{k})$ e $P(da_j|\mathbf{k})$ são dados pelos valores de *hub* e autoridade retornados pelo algoritmo HITS [48] para o documento d_j, respectivamente.

A informação de ligações fornecida pelo algoritmo HITS pode obtida localmente, denominada *informação local*, ou globalmente, denominada *informação global*. A informação é obtida localmente quando se consideram apenas as ligações entre as páginas relacionadas com a consulta do usuário. A informação é obtida globalmente quando se consideram todas as páginas da coleção. No experimentos que se seguem, ambos os tipos de informação, local e global, foram testados.

Para avaliar a eficácia do modelo de combinação proposto, uma série de experimentos foi executada numa coleção extraída da Web Brasileira [69]. Foram usadas 50 das consultas mais populares submetidas à ferramenta de busca TodoBR (`http://www.todobr.com.br`). A Tabela 1 mostra os resultados obtidos pelo modelo proposto em termos de valores de precisão e revocação [3], usando informação local, para três possíveis combinações de informação de ligações com conteúdo: vetorial+autoridade, vetorial+*hub* e vetorial+autoridade+*hub*. Como base de comparação é mostrado o resultado da ordenação vetorial usada isoladamente. A Tabela 2 mostra os mesmo valores usando informação obtida globalmente.

	Vetorial	Vet-aut		Vet-hub		Vet-hub-aut	
Revocação	Precisão	Precisão	Ganho	Precisão	Ganho	Precisão	Ganho
10	0.610	0.582	-5%	0.701	15%	0.689	13%
20	0.501	0.556	11%	0.647	29%	0.614	23%
30	0.427	0.477	12%	0.613	44%	0.582	36%
40	0.286	0.394	38%	0.511	79%	0.566	98%
50	0.185	0.322	74%	0.378	104%	0.524	183%
60	0.114	0.281	146%	0.295	159%	0.456	300%
70	0.054	0.195	261%	0.187	246%	0.357	561%
80	0.031	0.160	416%	0.150	384%	0.299	865%
90	0.026	0.132	408%	0.134	415%	0.204	685%
100	0.010	0.103	930%	0.106	960%	0.117	1070%
Média	0.267	0.351	31%	0.405	52%	0.466	74%

Tabela 1: Precisão média para a ordenação combinando conteúdo com ligações. A informação de ligações foi obtida localmente. A coluna *Ganho* mostra o ganho em precisão sobre a ordenação vetorial usada isoladamente.

	Vetorial	Vet-aut		Vet-hub		Vet-hub-aut	
Revocação	Precisão	Precisão	Ganho	Precisão	Ganho	Precisão	Ganho
10	0.610	0.659	8%	0.748	23%	0.756	24%
20	0.501	0.549	10%	0.688	37%	0.713	42%
30	0.427	0.472	11%	0.617	44%	0.637	49%
40	0.286	0.364	27%	0.411	44%	0.492	72%
50	0.185	0.201	9%	0.232	25%	0.323	75%
60	0.114	0.112	-2%	0.142	25%	0.150	32%
70	0.054	0.073	35%	0.072	33%	0.092	70%
80	0.031	0.029	-6%	0.035	13%	0.043	39%
90	0.026	0.024	-8%	0.026	0%	0.024	-8%
100	0.010	0.010	0%	0.010	0%	0.010	0%
Média	0.267	0.292	10%	0.345	29%	0.360	35%

Tabela 2: Precisão média para a ordenação combinando conteúdo com ligações. A informação de ligações foi obtida globalmente. A coluna *Ganho* mostra o ganho em precisão sobre a ordenação vetorial usada isoladamente.

Os resultados permitem concluir que a combinação de informação textual com informação de ligações traz melhores resultados que o uso de cada uma isoladamente. Usando informação local, foram obtidos ganhos em precisão de cerca de 74% sobre a ordenação vetorial. Usando informação global os ganhos obtidos foram de 35%.

Embora o uso de informação local tenha trazido ganhos mais altos, informação global foi capaz de obter melhor precisão no topo da ordenação, com um ganho de 28% nos 10 primeiros documentos. Isto é importante pois é reconhecido que em ferramentas de busca reais na Web os usuários estão especialmente interessados nos primeiros documentos retornados. Além disso, informação global é calculada apenas uma vez para toda a coleção, sendo por isso mais eficiente que informação local, que deve ser calculada para cada nova consulta.

Classificação de Documentos usando Informação de Ligações

A segunda parte deste trabalho aborda a questão de classificação de documentos na Web. É reconhecido que classificadores tradicionais, baseados apenas no conteúdo textual das páginas, têm um pobre desempenho na Web [13,37]. Neste trabalho é proposto um modelo de redes Bayesianas capaz de usar informação de ligações para melhorar o desempenho de um classificador tradicional.

Para combinar informação de ligações com o resultado de um classificador tradicional, propomos o modelo mostrado na Figura 2. Neste, os nós D_i representam os documentos de treino para o classificador, os nós T_j representam a informação textual dos documentos de teste, os nós L_j representam

a informação de ligações dos documentos de teste, o nó C representa uma classe e os nós F_j representam a combinação final de evidências relacionadas aos documentos de teste. Tal como na rede anterior, a cada nó corresponde uma variável aleatória binária.

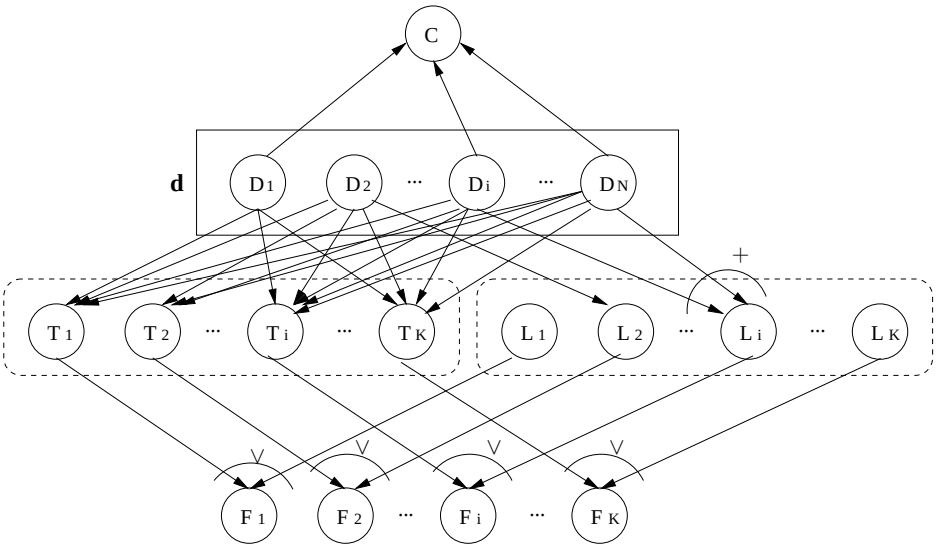


Figura 2: Rede Bayesiana para combinar o resultado de um classificador baseado em conteúdo com informação de ligações.

Pretendemos determinar a probabilidade de um documento d_i pertencer à classe C, ou seja, $P(F_i = 1|C = 1)$. Isto traduz-se na equação:

$$P(f_i|c) = \rho\Big(1-\big(1-W_t\times class(i,\mathcal{C},\overline{\mathcal{C}})\big)\big(1-W_l\times\alpha\sum_{j\in\mathcal{V}(i)\wedge j\in\mathcal{C}} link(i,j)\big)\Big) \quad (2)$$

onde $\rho = P(\mathbf{d})/P(c)$ é uma constante de normalização, $\mathcal{C}$ é o conjunto de documentos de treino pertencentes à classe C, $\mathcal{V}(i)$ é o conjunto de documentos de treino relacionados ao documento d_i através das suas ligações e W_t e W_l são pesos usados para atribuir diferentes importâncias às evidências derivadas do conteúdo textual e das ligações, respectivamente.

A função $class(i,\mathcal{C},\overline{\mathcal{C}})$ representa o valor retornado pelo classificador de conteúdo referente à probabilidade de d_i pertencer à classe C. Neste trabalho,

foram usados três classificadores tradicionais para calcular o valor de *class*: *kNN* [89], Naive Bayes [56] e Support Vector Machines [44].

A função $link(i, j)$ representa a similaridade de ligações entre o documento d_i e o documento d_j. Neste trabalho, esta similaridade foi medida usando co-citação [72], acoplamento bibliográfico [47], similaridade de Amsler [2] e o algoritmo Companion [25].

Para avaliar o desempenho do modelo, foram efetuados testes com duas coleções pré-classificadas extraída do diretório brasileiro Cadê (`http://www.cade.com.br`). A primeira coleção, denominada Cade12, consiste nas 12 categorias do primeiro nível do diretório. A segunda, denominada Cade188, consiste nas 188 categorias do segundo nível do diretório.

A Tabela 3 mostra os resultados da combinação de ambas as fontes de informação. Como medida de desempenho foi usado F_1 [90]. As colunas *Ganho/texto* e *Ganho/lig.* mostram, respectivamente, o ganho da combinação de fontes de informação sobre o uso do classificador de conteúdo e o ganho sobre as similaridades de ligações usadas isoladamente. Para as medidas de similaridade de ligações foi considerado o uso de ligações de, e para, páginas fora da coleção (ligações *externas*), uma vez que o número de ligações entre páginas na coleção (ligações *internas*) se revelou insuficiente. A medida de acoplamento bibliográfico foi também descartada, uma vez que apresentou resultados bastante inferiores às restantes.

Podemos, portanto, concluir que as ligações são uma fonte de informação valiosa para a classificação de páginas Web. A combinação de informação textual e de ligações obtém melhores resultados que o uso de cada uma isoladamente. No entanto, o ganho sobre os classificadores de conteúdo é significativamente mais expressivo que o ganho sobre o uso apenas de ligações. Os melhores resultados foram atingidos pelo algoritmo Companion,

Coleção	Similar.	W_l/W_t	Class.	F_1	Ganho/texto	Ganho/lig.
Cade12	Amsler	10	SVM	74.02	81%	6%
	Co-cit.	20	SVM	74.28	82%	7%
	Compan.	10	SVM	75.51	85%	8%
Cade188	Amsler	10	NB	62.01	154%	2%
	Co-cit.	10	NB	62.29	155%	2%
	Compan.	0.2	SVM	68.10	176%	11%

Tabela 3: Resultados obtidos pela combinação de informação para a classificação de páginas Web.

quando combinado com o classificador SVM, obtendo ganhos de 8% e 11% sobre o melhor classificador e sobre a melhor medida de similaridade usadas isoladamente, respectivamente.

Conclusões

Neste trabalho foi estudado o uso de informação de ligações para duas tarefas de Recuperação de Informação: ordenação e classificação de documentos. Foram propostos modelos de redes Bayesianas para resolver estes problemas combinando informação de ligações com informação textual.

Os experimentos efetuados em coleções extraídas da Web brasileira permitiram tirar as seguintes conclusões:

1. Ligações são, de fato, uma importante fonte de informação;

2. Os modelos propostos foram capazes de combinar eficazmente os diferentes tipos de informação: ligações e conteúdo textual;

3. No caso de ordenação de documentos, informação local traz melhores

resultados, mas informação global é mais apropriada para ferramentas de busca na Web;

4. No caso de classificação, informação de ligações provenientes de páginas fora da coleção é essencial para obter resultados expressivos.

O desenvolvimento deste trabalho deixou ainda algumas questões em aberto, citadas aqui como sugestão para trabalhos futuros. Em primeiro lugar, o refinamento do modelos propostos, para possibilitar a melhoria dos resultados ou a adaptação a outras coleções. Ainda neste tópico, deverão também ser criados algoritmos capazes de automatizar este processo de adaptação a novas coleções. Em segundo lugar, o estudo de técnicas de seleção de ligações, para evitar ligações ruidosas ou que não provêem informação útil. Finalmente, a criação de novos modelos que possam atender a outros problemas de Recuperação de Informação, na Web e não só.

Using Link Structure for Information Retrieval in The World Wide Web

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The field of Information Retrieval has been greatly affected by the appearance and growth of the World Wide Web. Among many new techniques, *link analysis*, i.e., the extraction of information from the Web link structure, has become of great importance. This work concerns the use of information derived from links among Web pages to solve Information Retrieval problems. In this chapter, we develop and discuss the goals and contributions of our project.

1.1 Information Retrieval

In today's world, digital data has become the most common mean for the storage of information. Computer databases, digital libraries and, of course, the Internet, are all regarded as commonplace repositories that provide access to information for users of all kinds, from highly specialized professionals to the general public. The whole set of problems surrounding the storage, access and organization of information in computer systems has become, therefore, the focus of several areas in computer science research, among

these Information Retrieval.

Information Retrieval (IR) focuses on providing users with access to information stored electronically. Unlike data retrieval, which studies solutions for the efficient storage and retrieval of structured data, IR is concerned with the extraction of information from non-structured or semi-structured text data. Application of IR techniques to deal with structured data has yet to be fully explored. One example of this type of approach can be found in [8, 10].

We can interpret the IR problem as composed of three main parts: the user, the IR system, and a digital data repository composed of the documents in a collection. The user has an *information need* that he translates to the IR system as a *query*. Given a user's query, the goal of the IR system is to retrieve from the data repository the documents that satisfy the user's information need, i.e., documents that are *relevant* to the user. Usually, this task consists of retrieving a set of documents and *ranking* them according to the likeliness that they will satisfy the user's information need.

Traditionally, IR was concerned with documents composed only of text. User queries were sets of keywords. Finding documents likely to satisfy a user's need consisted, basically, of finding documents that contained the words in the user's query. Several IR models were proposed based on this general principle [3]. However, with the growth in size and popularity of the *World Wide Web*, the IR field has changed to adapt to a new environment.

1.2 Information Retrieval in the World Wide Web

The World Wide Web has become very popular. Its ease of use and accessibility make it a very important tool, not only for communication, but also for the storage and sharing of information. In fact, from an IR point of view, the Web can be seen as a very large, publicly accessible, data repository, containing documents, or *Web pages*, that are interconnected and contain multimedia elements. The role of a Web IR system is to find, among these documents, information to satisfy the needs of Web users.

The Web, however, has unique characteristics, that impose new challenges to the IR field. Its size, estimated in 2004 at more than 3 billion pages[1], and its continuous growth make it larger than most existing digital repositories. Documents are spread across a large network of small servers, thus making the Web a distributed repository. Most of the stored data is very volatile, with documents constantly being modified, removed or added. Data is also very noisy, containing much erroneous information. Documents contain not only text, but also sound, video, and other media types. Finally, and perhaps most importantly, the majority of Web users are not only non-specialized, with little skills in the use of information systems, but also have a wide range of different interests. User queries are often very vague [74, 75], and the users are frequently unsure of what they are exactly looking for.

In such an environment, traditional text based IR approaches become insufficient. New sources of evidence have to be used to allow determining with greater precision the relevance of the documents with regard to a user

[1] See `http://searchenginewatch.com/reports/sizes.html` and `http://www.searchengineshowdown.com/stats/`.

query. For instance, information extracted from images and other media types, can be combined with text analysis; information on the internal structure of Web documents might be used as an indicative of the importance of the documents content; user logs can be analyzed and used to determine which documents better fit the users needs; the hypertextual nature of the Web can be used to indicate a document's topic and importance. The work here presented involves the use of this last resource, information derived from the link structure among Web documents, to solve Web IR problems.

1.3 Link Analysis

Information is implicitly present in the link structure of the Web. Say a given Web page A has a link to a Web page B. Observing this link, we can make two assumptions. First, the author of page A considers page B to have some importance, since he has inserted a link to it. Thus, the $A \rightarrow B$ link means that the author of A somewhat *endorses* the contents of page B. Second, if pages A and B are linked, we can expect their topics, i.e., the information subjects that the pages cover, to be similar. Thus, pages A and B are expected to contain related information. Observing these assumptions, we can devise two main applications for the information extracted from the Web's link structure: 1) determining the importance of a page, and 2) determining the topic of a page.

Both applications have practical uses in Web IR. Web user queries are often very vague, containing, on average, no more that two words [75]. For instance, a user looking for information on "the launching schedule of the space shuttle Endeavour" will, most often, build a simple query, like "endeavour". Also, many users search for general information, like "guitar tabs", looking

not for a specific document, but for a Web site that has good information regarding the subject. In both cases, the query alone conveys very little useful information on the user's needs. A measure of page importance is, therefore, an important complement for determining which pages are more likely to satisfy the user. Several works have already demonstrated the importance of link analysis for Web page ranking [7,70] and finding site homepages [39,86].

A distinct approach consists of organizing documents in categories, according to their topic, to assist the user with finding the information of his interest [16,78]. These systems are usually called *Web directories*[2]. Due to the size of the Web, it is unfeasible to build such directories manually. In this case, automatically determining the topic of Web pages is an essential task. Web link information has been shown useful to automatically classifying documents in a directory, as discussed in [13,58,71].

In sum, information derived from the analysis of Web links is of importance for Web IR and can be successfully used to complement existing sources of information. In this work, we study the application of link information to: (a) rank the answers to a user query and (b) classify Web documents according to the topics of a directory.

1.4 Objectives and Contributions

This work intends to provide answers to the following research questions:

- How effective is link information to assist with the execution of IR tasks on the Web, such as document ranking and document classification?
- What type of information should be extracted from the Web link structure?

[2] See, for instance, `http://dmoz.org/`.

- How effective is link information used alone, and how effective is its combination with other sources of evidence, such as the Web pages' textual contents?

To achieve the answers, we propose formal models based on Bayesian networks to solve two Web IR problems, ranking and classification. Our models are important because they allow to naturally combine the distinct pieces of evidence. To test these models, experiments were performed on a collection of documents extracted from the Web. The major contributions of this work are, therefore:

- The definition of two formal models that allow combining link-based and content-based information to support the tasks of ranking and classifying Web documents;
- An empirical comparison of existing link analysis algorithms;
- A detailed empirical study of the effects of link information on ranking and classification of Web documents;
- A set of important guidelines on how links should be used to improve the effectiveness of IR tasks such as document ranking and classification.

In addition, the models obtained from this study can also be applied to other Web IR problems, such as document clustering, site homepage finding, information filtering, and to similar problems in other hypertext collections.

1.5 Organization of this Work

The first part of this work, composed of Chapters 1, 2, and 3, gives some background on Web IR problems and introduces link analysis as a potential

solution. In particular, Chapter 2 discusses research related to our work and Chapter 3 introduces basic concepts related to IR, link analysis, and Bayesian networks, essential to the understanding of this work.

In the second part, composed of Chapters 4 and 5, two models to combine link-based and content-based information are presented. Chapter 4 shows how the textual content of Web pages can be combined with Web link structure information, through a Bayesian network model, to improve Web document ranking. Experiments with a Web collection were performed to evaluate the model's effectiveness and compare the use of *global* and *local* link information. Chapter 5 presents a Bayesian network model that also combines content-based and link-based information, this time to classify Web documents according to a directory. Experiments with two collections of Web documents indicate that the model can be successfully used to improve classification methods based only on content.

Finally, in Chapter 6 some conclusions and final analysis of the results are presented. Suggestions are also made regarding future steps to be taken to solve the problems left open by this research.

Chapter 2

Related Work

Citations among documents were first used as a source of information in bibliometric science. Later, these techniques were adapted for the Web, where links among pages take the role of bibliographic citations. In this chapter, Sections 2.1 and 2.2 review some seminal works related to both areas. Since our proposed models use Bayesian networks as a formal basis, important works related to the use of Bayesian networks in IR are also described in Section 2.3.

2.1 Citations

Cross-referencing information was first used in bibliometric science. Citations among scientific papers were used both to find papers on related topics, and to measure the importance of a publication.

In 1963, Kessler introduced the notion of bibliographic coupling [47]. We say that two documents have one unit of bibliographic coupling between them if both reference a same document. This measure can be used to determine documents with similar topics, which we would expect to have a high value

of bibliographic coupling. For this reason, bibliographic coupling was also used to cluster scientific journals [73]. Later, the measure of co-citation was introduced by Small in [72]. The co-citation value of two documents is defined as the frequency with which those two documents are cited together. Co-citation and bibliographic coupling have been used as complementary sources of information for document retrieval and classification [2, 5]. Citations also were suggested as a means to evaluate the importance of scientific journals [33], where the importance of a journal was assumed proportional to the number of citations to its papers, and, in a work as early as Kessler's proposal, Salton also suggested the use of citations for automatic document retrieval [63].

Later works have presented hypertext retrieval systems capable of using citations, or any other type of link, as a complement to full-text searching [6,20,30]. Taking advantage of the network formed by documents and the links between them, Croft et al. used spreading activation techniques [21] for hypertext retrieval. Croft and Turtle have also suggested the use of Bayesian networks as a modeling framework for hypertext [22]. More recently, citations have been used to index and retrieve scientific papers published in the Web [35].

In the Web, links among the documents can take the role of cross-references. This assumption has recently been used to rank documents in Web information systems, as we show in the following section.

2.2 Web Link Information

The ideas used for citations among documents can be naturally transposed to the Web environment. However, several distinctions must be made between

the Web and the domain of scientific publications. Papers are peer reviewed, thus ensuring the referencing of other important papers on the same subject. Web pages, on the other hand, may reference other unimportant pages, pages on unrelated subjects, or they may even refuse to reference each other, even if they are authorities on the same subject (for instance, pages from rival companies, like Sun and Microsoft). Also, links can be used for navigational purposes or even to artificially increase the likelihood that a page be retrieved by a given search engine (spamming).

The application of bibliometric techniques to the Web, and the necessary adaptation of these techniques to the new context, has given rise to algorithms for improving retrieval performance (i.e., quality of results) in the Web, which we now review.

2.2.1 Extracting Link Information

Taking many of the ideas proposed in bibliometric science, Brin and Page [7] propose an algorithm, named PageRank, that uses the Web link structure to derive a measure of popularity for Web pages. Intuitively, a page is considered popular if it is linked by many other pages. To account for the fact that, in the Web, pages can link freely among each other, this measure is refined: a page is recursively defined as popular when it is linked to by many other popular pages.

Kleinberg [34,49] later proposes the HITS (Hyperlink-Induced Topic Search) algorithm, where pages assume two distinct roles: as hubs and as 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. This hub/authority view of the link structure comes from the fact that, even if two authority pages do not link to each other, a third (hub) page might link to both.

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 popularity, what can make HITS classify certain pages as good hubs/authorities when they are not. To avoid this problem, the SALSA algorithm [52,53] computes the degrees of hub and authority for Web pages by examining random walks through the Web graph. Following a different perspective, Cohn and Chang [18] propose a model that is able to group documents in categories and determine the authoritative degree of a document within a category, using factor analysis in the set of links and documents in the collection.

2.2.2 Combining Link and Content Information

Although link structure is a valuable source of evidence, the textual content of a Web also is. For this reason, previous works have proposed combining link-based with content-based information in a single model.

In the algorithm of Chakrabarti et al. [14], the key idea is to use the text surrounding the links to determine a value of importance for each link analyzed. After determining values for all links, a ranking is computed using a weighted version of the algorithm proposed by Kleinberg [49].

Bharat and Henzinger [4] also have studied alternatives for combining link analysis with keyword-based evidence. Their work has two main differences when compared with the algorithm of Chakrabarti et al. [14]. First, although they also use keywords to determine the relevance of the links, this is done

taking all the words in the document. Second, they expand the original user query using the keywords of the documents in the answer. This expansion process improves retrieval performance but can be somewhat expensive, since it greatly increases the number of terms to be processed. Bharat and Henzinger also have modified the algorithm proposed by Kleinberg [49] using heuristics to reduce the weight of some links that degrade the results. An example of such type of links occurs when a set of links from a same site all point to a unique page.

Several works have reported significant results for the task of finding site homepages. In [84] and [32] link-based and content-based document rankings are combined through linear interpolation. In [46], the document's link-based ranking is inserted into the *tf-idf* weights [66] of its terms. In these works, however, improvements of link analysis for the task of document ranking were only marginal.

Finally, in a manner more similar to our proposal, Dumais and Jin [26] build a probabilistic model capable of combining link and content information, that works by iteratively propagating the information through the link structure of the Web.

2.2.3 Link Information for Web Classification

Link information has been previously proposed as a way of finding Web documents related to a same topic. The Companion algorithm [25], for instance, uses links to determine pages related to a given initial page. Its functionality is briefly described in Section 3.3. On a different approach, He et al. [42] propose a clustering algorithm that groups Web pages by operating on the graph defined by their link structure. Co-citation and text similarity measures are used to assign weights to the edges of the graph and partitioning algorithms

are used to split the set of pages into clusters. In [79], three measures of linkage similarity are compared to a human evaluation of similarity between Web pages. The authors come to conclusions quite different from our own, however, mainly due to the collection used—a set of academic sites from the U.K. This collection has a very different link structure where, for instance, many of the pages link to each other, a phenomena that we cannot expect in a Web directory (or in the Web in general [51]).

Differently from simply finding related documents, several other works in the literature have reported the successful use of links as a means to improve classification performance. Using the taxonomy presented in Sun et al. [77], we can summarize these efforts in three main approaches: hypertext, link analysis, and neighborhood.

In the hypertext approach, Web pages are represented by context features, such as terms extracted from linked pages, anchor text describing the links, or text from the paragraphs surrounding the links. As examples of this approach, Furnkranz et al. [31], Glover et al. [36] and Sun et al. [77] achieved good results by using anchor text and the paragraphs and headlines that surround the links. Similarly, Yang et al. [91] show that the use of terms from linked documents works better when neighboring documents are all in the same class.

In the link analysis approach, learning algorithms are applied to handle both the text components of the Web pages and the links between them. Slattery and Mitchell [71], for instance, explore the hyperlink topology using a HITS based algorithm to discover test set regularities. Joachims et al. [45] studied the combination of support vector machine kernel functions representing co-citation and content information. Cohn et al. [19] show that classification performance can be improved by using a combination of link-based

and content-based probabilistic methods. Finally, Fisher and Everson [28] extended this work by showing that link information is useful when the document collection has a sufficiently high density in the linkage matrix and the links are of high quality.

In the neighborhood approach, the document categories are estimated based on category assignments of already classified neighboring pages. The algorithm proposed by Chakrabarti et al. [13] uses the known classes of training documents to estimate the class of the neighboring test documents. Their work shows that co-citation based strategies are better than those using immediate neighbors. Oh et al. [58] improved on this work by using a filtering process to further refine the set of linked documents to be used.

The classification method presented in this work combines the link analysis and neighborhood approaches, differing from previous works in two main issues. First, we analyze several distinct similarity measures derived from links and determine which ones provide the best results for predicting the category of a document. Second, we propose a Bayesian network model that takes advantage of both the information provided by a content-based classifier and the information provided by the document's link structure. This model is independent of the classifier used, thus allowing us to study different classifier/link measure combinations. It also provides a formal and flexible way to test and combine new link-based and content-based algorithms in general.

2.3 Bayesian Networks in Information Retrieval

Bayesian networks, introduced by Pearl in [60], provide a graphical formalism for explicitly representing independencies among the variables of a joint

probability distribution. They were first used in IR problems by Turtle and Croft in [81, 82]. In their model, index terms, documents and user queries are seen as events and are represented as nodes in a Bayesian network. The model takes the viewpoint that the observation of a document induces belief on its set of index terms, and that specification of such terms induces belief in a user query or information need. This model was shown to perform better than traditional probabilistic models and used to effectively combine different sources of information for the task of document ranking.

Later, a second model was proposed by Ribeiro-Neto and Muntz in [61], where the elements of an IR system are formally defined as concepts in a sample space. Their work not only provides a probabilistic justification for the model, but also demonstrates that the combination of evidence from past queries with evidence from the vector space model yields better results than the use of a vector ranking alone.

More recently, Acid et al. [1] presented a third model whose network topology is defined in such way that an exact propagation algorithm, also proposed in their work, can be used to efficiently compute the relevance probabilities of the documents. When compared to Turtle and Croft's work for the task of document ranking, this model shows better performance in four out of five reference collections.

Bayesian networks have also been applied to other IR problems besides ranking as, for instance, relevance feedback [38], automatic construction of hypertext [68], query expansion [23], information filtering [12], assigning structure to database queries [9], document clustering and classification [27], retrieval of juridical information [24], and retrieval of images from the Web [17].

In this paper, we adopt the Bayesian framework proposed by Ribeiro-Neto

and Muntz in [61] for modeling distinct Web IR problems and for combining content-based and link-based information. Early works by Croft and Turtle [22, 80] have already suggested that the combination of links (either derived from citations or artificially created) and content information can be used to improve document retrieval. In their model, however, the addition of link-related evidence is equivalent to adding the terms extracted from the linked documents to the document being ranked. In our proposed models, on the other hand, link evidence is completely independent of the documents' content, and can be derived by any link analysis algorithm.

Chapter 3

Basic Concepts

This chapter introduces basic concepts required for a better understanding of our proposed ranking and classification algorithms. The vector space model, described in Section 3.1, is the most common IR method for ranking documents with regard to a user query and is extensively used throughout our work. To extract link-based information, we use two well known methods of link analysis: PageRank and HITS, which are described in Section 3.2. In Sections 3.3 and 3.4, we discuss the linkage similarity measures and content-based classification algorithms. Since our framework is based on Bayesian networks, we review in Section 3.5 how they are used to represent classic information retrieval ranking algorithms. Finally, Section 3.6 presents some measures commonly used to evaluate the retrieval performance of IR systems, useful to understand the results shown in the following chapters.

3.1 The Vector Space Model

The vector space model is a simple and effective model for retrieving information from a document collection [66]. In it, documents and queries

are represented as vectors in a space composed of index terms, i.e., words extracted from the text of the documents in the collection [85]. This vector representation allows us to use any vector algebra operation to compare queries and documents, or to compare a document to another one.

In the vector space model model, with every term k_i in a document d_j is associated a weight w_{ij}. A document d_j is, thus, represented as a vector of term weights $\vec{d_j} = (w_{1j}, w_{2j}, ..., w_{tj})$, where t is the total number of distinct terms in the entire document collection. Each w_{ij} weight reflects the importance of term k_i in document d_j and is usually computed as:

$$w_{ij} = tf_{ij} \times \log \frac{N}{n_i} \tag{3.1}$$

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 N is the total number of documents in the collection. The factor $\log(N/n_i)$ is called the *inverse document frequency* (IDF) and is used to stress the influence of terms that are more selective because they appear less frequently in the document collection. The expression for w_{ij} is usually referred to as *term frequency-inverse document frequency* (or TF-IDF) weight. Its foundations lie in the observation that a term is more important if it occurs many times in a document and less important if it occurs in many documents in the collection.

In the vector space model, 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:

$$sim(d_j, q) = \frac{\sum_{i=1}^{t} w_{ij} \times w_{iq}}{\sqrt{\sum_{i=1}^{t} w_{ij}^2} \times \sqrt{\sum_{i=1}^{t} w_{iq}^2}} \tag{3.2}$$

By computing the similarities between all the documents in the collection and a given query q, we obtain an ordered set, where documents more likely to satisfy the query have higher similarities. To illustrate, Figure 3.1 shows the vectors corresponding to a document d_j and a query q, with terms k_a and k_b. The similarity between d_j and q is the cosine of angle θ.

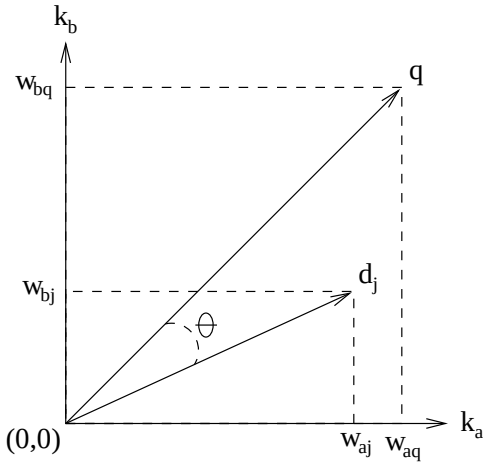


Figure 3.1: Representation of the similarity between document d_j and query q in the vector space model.

In this work, we use the vector space model to obtain the information present in the contents of the documents, both for document ranking, in which documents are compared to user queries to determine the likeliness of satisfying the user's needs, as for document classification, where documents are compared to each other, to determine their similarity of topic.

3.2 Link Analysis Algorithms

Several algorithms have been proposed to extract information from the link structure of the Web. In a general manner, they all work by viewing the Web as a directed graph, where each node corresponds to a Web page, and each edge corresponds to a link between pages. Importance measures from the Web pages are obtained by finding the eigenvectors of the adjacency matrix representing the Web links. Here, we describe the two most used algorithms: PageRank and HITS.

3.2.1 The PageRank Algorithm

The PageRank algorithm was introduced by Brin and Page in [7], and a detailed description of it can be found in [59]. PageRank assumes that a page is more important, or more popular, if it is linked to by many other important, or popular, pages. This definition gives rise to the following recursive equation:

$$R(u) = \sum_{v \in B_u} \frac{R(v)}{N_v} \tag{3.3}$$

where $R(u)$ is the PageRank value for page u, B_u is the set of pages that link to u, and N_v is the number of outgoing links (out-links) from page v. The values of $R(u)$ are subject to:

$$\sum_{\forall u} R(u) = 1 \tag{3.4}$$

Intuitively, we can think of PageRank as modeling the behavior of a random surfer in the Web. The surfer moves through the Web by randomly jumping from the page it is to one of its neighbor pages. For this, it selects an outgoing link randomly. Equation (3.3) gives the probability that the surfer will visit page u. Figure 3.2 shows an example of a PageRank computation.

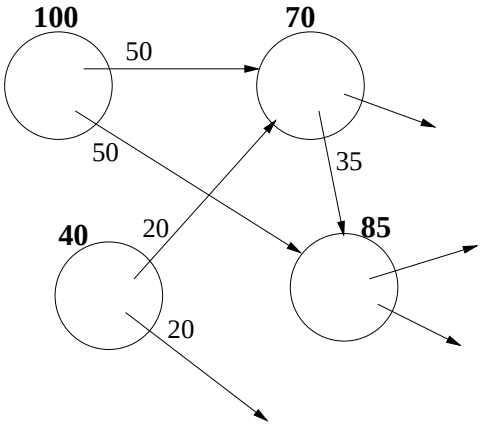


Figure 3.2: A computation of PageRank value.

This definition, however, has a problem. If there is at least a set of pages disconnected from all the rest, i.e., if the transition matrix is reducible, the surfer can get stuck in a cycle. To avoid this, we can assume that the surfer eventually gets bored from following links and jumps to a random page. This translates to the following equation:

$$R(u) = \frac{d}{N} + (1-d) \sum_{v \in B_u} \frac{R(v)}{N_v} \tag{3.5}$$

where d is the so-called *damping factor*, which establishes a probability threshold with which our surfer jumps to a new page at random, independently of link relations. The new page is selected from the set of all pages, whose cardinality is N.

To further examine the meaning of Equation (3.5), let A be the $N \times N$ adjacency matrix representing the link relations in the Web. The A_{ij} entry stands for a link from node i to node j. Each node i represents a Web page. Consider a matrix A', where each entry is defined as:

$$A'_{ij} = \frac{d}{N} + (1-d)\frac{A_{ij}}{N_v} \tag{3.6}$$

Matrix A' represents a finite state positive recurrent Markov chain. Writing the PageRank equation in matrix form, we have:

$$R = (A')^T R \tag{3.7}$$

Together with Equation (3.4), this system of equations gives the steady-state stationary probabilities of the Markov chain represented by A'. Thus, the PageRank of a document is the long-run proportion of times that the random surfer will visit the document on an infinite random walk.

3.2.2 The HITS Algorithm

We cannot always expect an important Web page to link to other important Web pages. For instance, it will be very hard to find a link from the Microsoft Web site to the Sun Web site, or vice-versa, even though they are on the same topic — computer systems. However, we can easily expect a third page, for instance, from a computer systems engineer, to link to both. Thus, we can think of Web pages as having two distinct functions: providing valuable content on a given topic, called an *authority* page; and linking to many authority pages, called a *hub* page. This idea, illustrated in Figure 3.3, led to the algorithm introduced by Kleinberg in [49], later named HITS [34].

Unlike PageRank, which was designed to be used on the whole collection of documents, HITS works on the set of documents related to a user query Q. This set can be found, for instance, by using the vector space model and taking the highest ranked documents. Let R_Q be the set of documents (or Web pages) associated with query Q. We call R_Q the *root set* of documents. To guarantee that there will be enough links among the documents in R_Q, a condition necessary for obtaining valid link information, we expand this set by including documents that link to, or that are linked to by a node in R_Q.

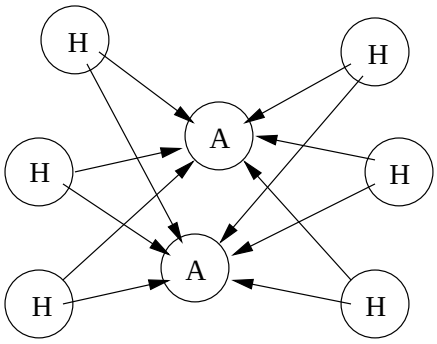


Figure 3.3: Hubs and authorities in an hyperlinked collection.

This expanded set, labelled as B_Q, is called the *base set* of documents. The HITS algorithm is applied to the graph $G = (B_Q, E)$, where the edges in E represent the links between the documents in B_Q. The authority value of a document p is defined as:

$$a(p) = \sum_{d|(d,p)\in E} h(d) \tag{3.8}$$

Recursively, the hub value of a document p is defined as:

$$h(p) = \sum_{d|(p,d)\in E} a(d) \tag{3.9}$$

Thus, a document is considered a good authority if it is linked by many good hubs. Conversely, a document is considered a good hub if it links to many good authorities.

In [49], it is shown that if we iteratively apply Equations (3.8) and (3.9), the values of $a(\cdot)$ and $h(\cdot)$ converge to the principal eigenvectors of A^TA and AA^T, respectively, where A is the adjacency matrix of G. Interestingly, the matrix A^TA is known as the *co-citation* matrix of the collection, whereas AA^T is the *bibliographic coupling* matrix of the collection. Thus, there is a direct relation between a page hub and authority values and the linkage similarity measures described in the following section.

3.3 Linkage Similarity Measures

In our classification experiments, we used five different similarity measures, derived from link structure, to determine a degree of similarity among Web pages: co-citation, bibliographic coupling, Amsler, authority degrees provided by the Companion algorithm, and hub degrees provided by the Companion algorithm. The first three were introduced in bibliometric science, to quantify the relationship between two scientific papers [2, 47, 72]. In this work, we evaluate how they perform when applied to the Web environment, where we assume that links between Web pages have the same role as citations between scientific papers. The Companion algorithm was proposed by Dean and Henzinger [25], as a method to find Web pages related to each other. Here, we use it to provide a value of similarity between documents. We now describe in detail each of the proposed linkage similarity measures.

3.3.1 Co-Citation

Co-citation was first proposed by Small [72] as a similarity measure between scientific papers. Two papers are co-cited if a third paper has citations to both of them. This reflects the assumption that the author of a scientific paper will cite only papers related to his own work. Although Web links have many differences from citations, we can assume that many of them have the same meaning, i.e., a Web page author will insert links to pages *related* to his own page. In this case, we can apply co-citation to Web documents by treating links as citations. As illustrated in Figure 3.4, we say that two pages are co-cited if a third page has links to both of them.

To further refine this idea, let d be a Web page and let P_d be the set of pages that link to d, called the *parents* of d. The co-citation similarity

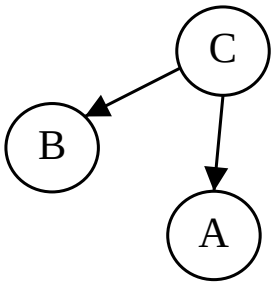


Figure 3.4: Pages A and B share one unit of co-citation, since they are both linked to by page C.

between two pages d_1 and d_2 is defined as:

$$cocitation(d_1, d_2) = \frac{|P_{d_1} \cap P_{d_2}|}{|P_{d_1} \cup P_{d_2}|} \tag{3.10}$$

Equation (3.10) tells us 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.

3.3.2 Bibliographic Coupling

Also with the goal of determining the similarity between papers, Kessler [47] introduced the measure of bibliographic coupling. Two documents share one unit of bibliographic coupling if both cite a same paper. The idea is based on the notion that paper authors who work on the same subject tend to cite the same papers. As for co-citation, we can apply this principle to the Web. We assume that two authors of Web pages on the same subject tend to insert links to the same pages. Thus, we say that two pages have one unit of bibliographic coupling between them if they link to the same page, as shown in Figure 3.5.

More formally, let d be a Web page. We define C_d as the set of pages that

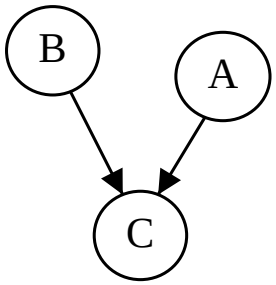


Figure 3.5: Pages A and B share one unit of bibliographic coupling, since they both link to page C.

d links to, also called the *children* of d. Bibliographic coupling between two pages d_1 and d_2 is defined as:

$$bibcoupling(d_1, d_2) = \frac{|C_{d_1} \cap C_{d_2}|}{|C_{d_1} \cup C_{d_2}|} \tag{3.11}$$

According to (3.11), 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.

3.3.3 Amsler

In an attempt to take the most advantage of the information available in citations between papers, Amsler [2] proposed a measure of similarity that combines both co-citation and bibliographic coupling. According to Amsler, two papers A and B are related if (1) A and B are cited by the same paper, (2) A and B cite the same paper, or (3) A cites a third paper C that cites B. As for the previous measures, we can apply the Amsler similarity measure to Web pages, replacing citations by links, as illustrated by Figure 3.6.

Let d be a Web page, let P_d be the set of parents of d, and let C_d be the set of children of d. The Amsler similarity between two pages d_1 and d_2 is

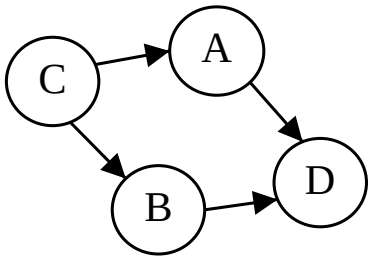


Figure 3.6: Pages A and B share two units of Amsler similarity, since they both link to page D and are both linked to by page C.

defined as:

$$amsler(d_1, d_2) = \frac{|(P_{d_1} \cup C_{d_1}) \cap (P_{d_2} \cup C_{d_2})|}{|(P_{d_1} \cup C_{d_1}) \cup (P_{d_2} \cup C_{d_2})|} \tag{3.12}$$

Equation (3.12) tell us that, the more links (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 neither d_1 nor d_2 have any children or parents, the similarity is defined as zero.

3.3.4 Companion

On a different approach, the Companion algorithm was proposed by Dean and Henzinger in [25]. Given a Web page D, the algorithm finds a set of related pages by examining its link structure, and returns a degree of how related each page is to D. In this work, we proposed that this degree can be used as a similarity measure between D and the remaining pages.

To find a set of pages related to a page D, the Companion algorithm has two main steps:

1. build a Vicinity Graph of D, and

2. compute the degrees of similarity.

In step 1, pages that are linked to D are retrieved. We build the set $\mathcal{V}$, the vicinity of D, that contains the parents of D, the children of the parents of D, the children of D, and the parents of the children of D. This is the set of pages related to D.

In step 2 we compute the degree to which the pages in $\mathcal{V}$ are related to D. To do this, we consider the pages in $\mathcal{V}$ and the links among them as a graph, called the vicinity graph of D, as illustrated in Figure 3.7. This graph is then processed by the HITS algorithm, which returns the degree of *authority* and *hub* of each page in $\mathcal{V}$ (see Section 3.2.2).

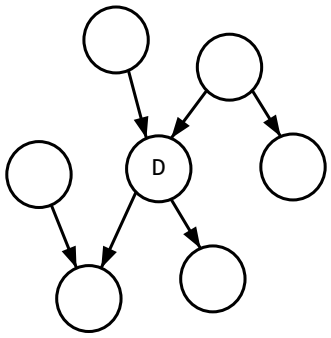


Figure 3.7: Vicinity graph of page D.

Once HITS is applied, we can choose to use the degree of authority or hub (or a combination of both) as a measure of similarity between D and each page in $\mathcal{V}$. We define the similarity between D and any page that is not in $\mathcal{V}$ as zero. In this work we experimented with the Companion algorithm using either the authority or the hub degree in isolation as a similarity measure.

3.4 Content-Based Classifiers

Document classification is the activity of assigning entries from a set of pre-specified categories to a document [67]. Given a set of categories $C =$

$\{c_1, c_2, ..., c_n\}$ and a set of documents $D = \{d_1, d_2, ..., d_n\}$, the classification task consists of determining which elements of C can be assigned to each element of D. The main challenge in this task comes from the fact that the rules to determine whether a document belongs or not to a category are not clearly defined. The most common solution consists of using machine learning algorithms, which are trained with a set of pre-classified documents, and use the knowledge thereby obtained to classify the whole collection.

Automatic classification algorithms are very useful, since there are many practical situations where the amount of data to be classified makes manual classification unfeasible. The most outstanding example is that of the World Wide Web, where we have millions of documents to be classified and where the data is commonly volatile, thus requiring frequent reclassifications. In the Web, the assignment of categories to documents is of great importance, since it can be used to construct on-line directories, such as Yahoo [1], to improve the precision of Web search engines, and even to help in the interactions between user and search systems [16, 78].

In this work, we used three well-known text classifiers: *kNN*, Support Vector Machine (SVM), and Naive Bayes. 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.

3.4.1 The *kNN* Classifier

The *kNN*, or *k nearest neighbors*, is a well known technique that has been widely studied in pattern recognition for over five decades [29]. It works by representing each data element to be classified as a point in an n-dimensional space. To assign a point to a category, its closest neighbors are examined and,

[1] http://www.yahoo.com

in general terms, the object is classified under the category of the majority, as illustrated in Figure 3.8.

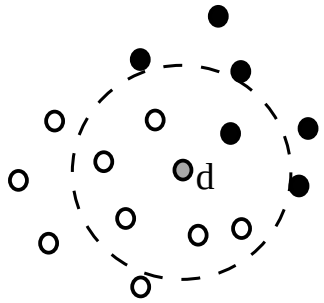


Figure 3.8: The *kNN* classifier. The class of point d is attributed according to the classes of its nearest neighbors. Points in black and white belong to distinct classes.

The most widely used *kNN* algorithm for Information Retrieval was introduced by Yang in [89]. It assigns a category label to a test document based on the categories attributed to the k most similar documents in the training set. More specifically, to a given test document d is assigned a relevance score $s_{c_i,d}$ that associates d to the candidate category c_i. This score is defined as:

$$s_{c_i,d} = \sum_{d' \in \mathcal{N}_k(d)} sim(d,d') f(c_i,d') \tag{3.13}$$

where $\mathcal{N}_k(d)$ are the k nearest neighbors of d, according to a given similarity function sim, and $f(c_i,d')$ is a function that returns 1 if document d' belongs to category c_i and 0 otherwise.

Traditionally, documents are represented by vectors of term weights and the similarity between two documents is measured by the cosine of the angle between them. Term weights are computed using one of the conventional TF-IDF schemes [65], in which the weight of a term in a document is defined as in Eq. (3.1). Based on the computed scores, we determine the top ranking

category and assign it to the test document.

In this work, we used the *Bow* implementation of *kNN* [55], available at `http://www.cs.cmu.edu/~mccallum/bow`.

3.4.2 The SVM Classifier

SVM is a relatively new method of classification introduced by Vapnik in [83] and first used in text classification by Joachims in [44]. 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 the hyperplane.

Figure 3.9 illustrates a space where points of different classes are linearly separable. The solid line represents a possible hyperplane separating both classes. This hyperplane can be described by:

$$(\vec{w} \cdot \vec{x}) + b = 0, \tag{3.14}$$

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 learned from a training set of linearly separable data. Classifying a vector is achieved by applying the decision function

$$f(\vec{x}) = \text{sign}((\vec{w} \cdot \vec{x}) + b) \tag{3.15}$$

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

In Figure 3.9, the dashed lines represent how much the hyperplane can be moved while still separating the classes. The SVM classifier tries maximize the margin between the hyperplane and the points in 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

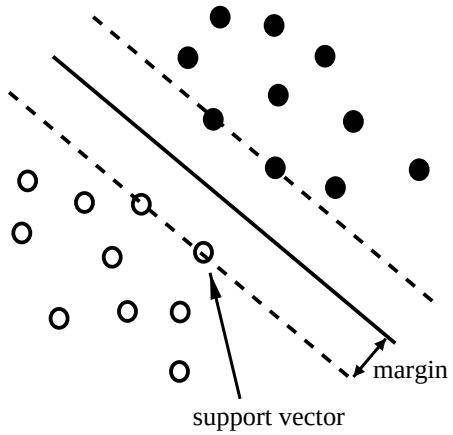


Figure 3.9: The SVM classifier. An separating hyperplane is found by maximizing the margin between both classes.

marginal planes of the classes, the *support vectors*, and is of the form:

$$\vec{w} = \sum_i v_i \vec{x}_i \tag{3.16}$$

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

$$f(\vec{x}) = \text{sign}(\sum_i v_i (\vec{x} \cdot \vec{x}_i) + b) \tag{3.17}$$

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:

$$f(\vec{x}) = \text{sign}(\sum_i v_i \, \text{k}(\vec{x} \cdot \vec{x}_i) + b) \tag{3.18}$$

where $k(\cdot, \cdot)$ is the kernel function.

Support vector machines only take binary decisions: a document belongs or not to a given class. In a multiple class setting, such as the one of this work, a different classifier needs to be learned for each class. To make the final decision, each classifier can be compared to all the others and a voting scheme can be used in which the class of the classifier with the more votes is chosen [43].

In this work, we used the *LIBSVM* implementation of support vector machines [15], available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

3.4.3 The Naive Bayes Classifier

The Naive Bayes method of classification uses Bayes theorem to determine the probability of a category, given the data to be classified. Its underlying assumption is that the attributes that characterize the data points are independent if their category is known, hence the name *naive* Bayes. In Information Retrieval, Naive Bayes classification uses the probabilities of words and categories to estimate the probability of a category given a document [56]. It is assumed that the presence of each word in a document is independent of all words in the document, given its class.

Using this term independence hypothesis, the Naive Bayes classifier assumes that documents are generated from a distribution parameterized by θ. The likelihood of a document d_i being generated is defined as:

$$P(d_i|\theta) = \sum_{j=1}^{C} P(c_j|\theta)P(d_i|c_j,\theta) \tag{3.19}$$

where each c_j represents a class and C is the number of classes. Eq. (3.19) states that a document is generated by (1) selecting one of the classes, with probability $P(c_j|\theta)$, and (2) selecting a document from the class, with probability $P(d_i|c_j,\theta)$. The probability of selecting a class is defined by the pro-

portion of documents belonging to the class, i.e.:

$$P(c_j|\theta) = \frac{\sum_{i=1}^{N} P(c_j|d_i)}{N} \tag{3.20}$$

where N is the total number of documents in the training set and $P(c_j|d_i)$ is defined as 1 if document d_i belongs to class c_j and 0 if otherwise.

In the model used in this work, each term in a document is seen as an event. Each term is represented by a random variable that takes a value from 1 to V, where V is the number of terms in the collection. A document of length n can be represented as a sequence of n term events, i.e., a sequence of n multinomial trials, were each trial is independent of the previous (due to the naive Bayes assumption). The probability of generating a document d_i given a class c_j is, therefore, defined as the multinomial distribution:

$$P(d_i|c_j,\theta) = n_i! \sum_{k=1}^{V} \frac{P(t_k|c_j,\theta)^{f_{ki}}}{f_{ki}!} \tag{3.21}$$

where n_i is the length of document d_i and f_{ki} is the number of times term t_k occurs in document d_i. The probability of a term t_k in a class c_j is defined as:

$$P(t_k|c_j,\theta) = \frac{1 + \sum_{i=1} N f_{ki} P(c_j|d_i)}{V + \sum_{l=1} V \sum_{i=1} N f_{si} P(c_j|d_i)} \tag{3.22}$$

which is the proportion of occurrences of term t_k in the documents of class c_j, normalized to avoid probabilities of 0 or 1.

Finally, once all the parameters are learned, a document can be classified by computing the probability of a class given a document. This is accomplished by applying Bayes' rule, i.e.:

$$P(c_j|d_i,\theta) = \frac{P(c_j|\theta)P(d_i|c_j,\theta)}{P(d_i|\theta)} \tag{3.23}$$

To classify document d_i, the class that maximizes Eq. (3.23) is chosen.

In this work, we used the *Bow* implementation of Naive Bayes [55], available at `http://www.cs.cmu.edu/~mccallum/bow`.

3.5 A Bayesian Network Model for IR

Bayesian networks [60] (also known as *inference networks* or *belief networks*) were chosen as a formal basis for this work since they have been shown to produce good results when applied to IR problems, both for simulating traditional IR models as for combining information from different sources [11, 61, 62, 82]. They also allow a uniform, flexible, intuitive and formally sound view of several link analysis problems.

As in the models by Turtle and Croft [81] and Ribeiro-Neto and Muntz [61], this work takes an epistemological view, as opposed to a frequentist view, of the information retrieval problem, interpreting probabilities as degrees of belief devoid of experimentation. In the following explanation, we will adopt the Ribeiro-Neto and Muntz model [61].

3.5.1 Basic Concepts

Bayesian networks provide a graphical formalism for explicitly representing independencies among the variables of a joint probability distribution. 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 $\mathbf{P}$ be the set of all parent nodes of a node X, $\mathbf{p}$ be a set of

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

$$\sum_{x \in D_X} \mathcal{F}(x, \mathbf{p}) = 1 \tag{3.24}$$

$$0 \leq \mathcal{F}(x, \mathbf{p}) \leq 1. \tag{3.25}$$

where D_X is the set of possible values for variable X. The function $\mathcal{F}(x, \mathbf{p})$ provides a numerical quantification for the conditional probability $P(x|\mathbf{p})$.

To illustrate, Figure 3.10 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. Due to the independencies declared in Figure 3.10, the joint probability distribution can be computed as:

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

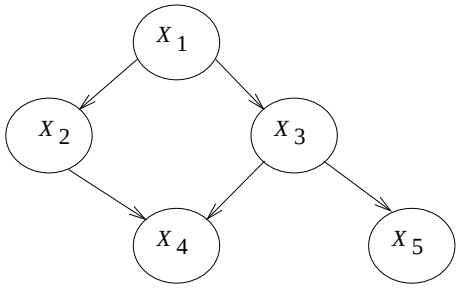


Figure 3.10: Example of a Bayesian network.

The formalism and flexibility of Bayesian networks can be used to solve IR problems, as we see in the following.

3.5.2 The Belief Network Model for IR

In a document retrieval problem, we can think of (at least) three basic components: keywords (or terms), documents, and a query. In a probabilistic model, each one can be seen as an event. These events are not independent, since, for instance, the occurrence of a term will influence the occurrence of a document. Using a Bayesian network, we can model these events and their interdependencies.

We start by associating with each term k_i, a binary random variable, denoted by K_i. This variable is 1 to indicate that an event associated with the term k_i has occurred. In the case of the IR problem, the event is often the *observation* of term k_i. This is so because we select terms to match both documents on the user query. To simplify our notation, we will write $P(k_i)$ as short for $P(K_i = 1)$, and $P(\overline{k_i})$ as short for $P(K_i = 0)$. Analogously, we associate a random variable D_j with each document d_j, and a random variable Q with the user query q. Considering the fact that both documents and queries are composed of keywords, we can model the document retrieval problem using the network in Figure 3.11.

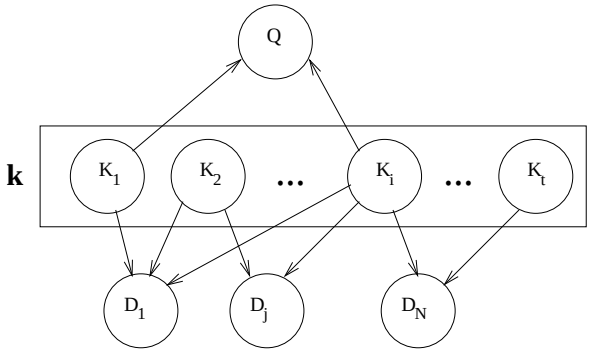


Figure 3.11: Belief network for a query q composed of the keywords k_1 and k_i.

In this network, each node D_j models a document, the node Q models the user query, and the K_i nodes model the terms in the collection. The vector $\mathbf{k}$ is used to refer to any of the possible states of the K_i root nodes. Instantiation of the root nodes *separates* the document nodes from the query node, making them mutually independent.

The similarity between a document d_j and the query q can be interpreted as the probability of document d_j being observed given that query q was observed. Thus, using Bayes' law and the rule of total probabilities, we compute the similarity $P(d_j|q)$ as follows:

$$P(d_j|q) = \eta \sum_{\mathbf{k}} P(d_j|\mathbf{k})\ P(q|\mathbf{k})\ P(\mathbf{k}) \tag{3.26}$$

where $\eta = 1/P(q)$ is a normalizing constant. Equation (3.26) is the generic expression for the ranking of a document d_j with regard to a query q, in the belief network model.

To represent any of the traditional IR models, using the network in Figure 3.11, we need only to define the probabilities $P(d_j|\mathbf{k})$, $P(q|\mathbf{k})$, and $P(\mathbf{k})$ appropriately. As an example, we now show how to represent the vector space model, described in Section 3.1.

3.5.3 A Belief Network for the Vector Space Model

A belief network can be used to compute a vector space ranking by making Equation (3.26) equivalent to Equation (3.2). This is accomplished through proper specification of the conditional probabilities. We start by defining $P(q|\mathbf{k})$ as follows:

$$P(q|\mathbf{k}) = \begin{cases} 1 & \text{if } \forall i,\ K_i = 1 \text{ iff term } K_i \text{ is in query } Q \\ 0 & \text{otherwise} \end{cases} \tag{3.27}$$

Equation (3.27) restricts the computation to the state $\mathbf{k}$ where the only observed terms are those in query q. We also define $P(\overline{q}|\mathbf{k}) = 1 - P(q|\mathbf{k})$.

The conditional probability $P(d_j|\mathbf{k})$ is defined as:

$$P(d_j|\mathbf{k}) = \frac{\sum_{i=1}^{t} w_{ij} \cdot w_{i\mathbf{k}}}{\sqrt{\sum_{i=1}^{t} w_{ij}^2} \sqrt{\sum_{i=1}^{t} w_{i\mathbf{k}}^2}} \tag{3.28}$$

where $w_{i\mathbf{k}}$ and w_{ij} are the *tf-idf* weights used in the vector space model (see Equation (3.1)), and t is the total number of distinct terms in the collection. Equation (3.28) can be recognized as the cosine similarity, defined in Section 3.1. If we define $P(\overline{d_j}|\mathbf{k}) = 1 - P(d_j|\mathbf{k})$, this specification is valid and consistent since $P(d_j|\mathbf{k})$ measures the cosine of the angle between two vectors, which is a number between 0 and 1.

Finally, the *a priori* probability $P(\mathbf{k})$ can be defined as constant for all $\mathbf{k}$:

$$P(\mathbf{k}) = \frac{1}{2^t} \tag{3.29}$$

Let $\mathbf{k}_q$ be the state where only the terms belonging to query Q are active. Applying Equations (3.27), (3.28), and (3.29) to Equation (3.26), we get:

$$P(d_j|q) = \alpha \times \frac{\sum_{i=1}^{t} w_{ij} \cdot w_{iq}}{\sqrt{\sum_{i=1}^{t} w_{ij}^2} \sqrt{\sum_{i=1}^{t} w_{iq}^2}} \tag{3.30}$$

where α is a constant that combines η and $P(\mathbf{k}_q)$. Thus, Equation (3.30) tells us that the ranking of documents defined by $P(d_j|q)$ coincides with the ranking of documents defined by the vector space model.

In a similar fashion, Bayesian network models can be built to take advantage of link information, as we demonstrate in the following chapters.

3.6 Evaluation Measures

To test the models proposed in this work, we ran experiments and evaluated the results using evaluation metrics. Here, we introduce the most common ones which are necessary for understanding the results shown in the following chapters.

3.6.1 Precision and Recall

To evaluate document ranking results we often use *precision* and *recall* figures [3, 54]. These metrics consider that, for each test query, a set of relevant documents has been defined. These relevance judgments are accomplished through human evaluations, made by specialists in the query topic, or by a group of system users.

Given a query Q and a set R of relevant documents for Q, precision and recall figures can be used to evaluate the quality of a retrieval method. Using the retrieval method, we obtain a set A of documents as answers to the query Q. This set is then compared to the set R of relevant documents. The higher the overlap between them, the better is considered the result. Precision and recall are defined as a means to characterize this overlap, as follows.

Precision, p, is the fraction of all answers in A that are correct, i.e:

$$p = \frac{|A \cap R|}{|A|}$$

Precision is defined 1 if no documents where retrieved, i.e., if $|A| = 0$.

Recall, r, is the fraction of correct answers that were properly retrieved in A, i.e.:

$$r = \frac{|A \cap R|}{|R|}$$

Recall is defined as 1 if there are no relevant documents, i.e., if $|R| = 0$.

Frequently, we want to evaluate average precision at given recall levels. The standard 10-point average precision measure returns precision at 10%, 20%, ..., 100% of recall. For instance, precision at 10% recall is the precision when 10% of the relevant documents in the set R have been seen in the ranking, starting from the top. Average precision at 10% recall is the average precision for all test queries, taken at 10% recall. Plotting the precision at the 10 standard recall points allows us to easily evaluate and compare the quality of ranking algorithms.

3.6.2 The F-measure

In classification tasks precision and recall are taken for every class. This yields a great number of values, making the tasks of comparing and evaluating algorithms more difficult. In these cases, 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* [90].

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

$$F_\alpha = \frac{(\alpha^2 + 1)pr}{\alpha^2 p + r} \tag{3.31}$$

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.

In our classification experiments, we assign equal weights to precision and recall by defining $\alpha = 1$. This yields the so called F_1 measure, defined as:

$$F_1 = \frac{2rp}{p + r} \tag{3.32}$$

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

classes, and can also be averaged over the classes to quantify overall retrieval performance.

Chapter 4

Ranking Web Documents by Combining Link-Based and Content-Based Information

When searching for documents in the Web, textual content alone (i.e., the text of the documents) may be insufficient to provide good results. A nowadays popular approach for improving search effectiveness is to use link information. In this chapter, we introduce a Bayesian network model that allows combining content-based and link-based evidence for ranking Web documents. Taking advantage of the model, we also study how the use of local link information, i.e., information extracted from the set of documents in the answer to a query, compares to the use of global link information, i.e., information extracted from the whole collection of documents.

4.1 The Evidence Combination Model

To combine link-based and content-based information for document ranking, we expand the belief network model discussed in Section 3.5.2 to include evidential information extracted from the link structure of the collection. This is accomplished by adding new edges, nodes, and probabilities to the original network. The resulting network is shown in Figure 4.1. This strategy allows us to combine, in a natural and convenient way, the keyword-based evidence associated with the content of the documents with the link-based evidence obtained from the surrounding hypertextual environment.

In the belief network of Figure 4.1, the left hand side represents the original network of Figure 3.11, where each document node D_j is renamed as Dc_j, indicating that it represents content-based information. The right hand side models the link-based sources of evidence. These can be obtained either from the link structure associated with the set of documents in the answer set to a query, thus being sources of *local evidence*, or from the link structure associated with the whole collection of documents, thus being sources of *global evidence*. In Section 4.2 we detail the differences between both sources.

To represent link-based evidential knowledge in the network, we associate two new nodes Dh_j and Da_j with each document d_j in the answer set for query q. We associate a binary random variable Dh_j with the node Dh_j to model evidence associated with the document d_j as a hub. This evidence is computed from the link structure associated with the set of documents, using the HITS algorithm presented in Section 3.2.2. Hub values are represented in our network as the conditional probability of Dh_j being active given the keywords in the query q and given an implicit knowledge of the surrounding link structure. Analogously, we associate a binary random variable Da_j with the node Da_j to model evidence associated with the document d_j as

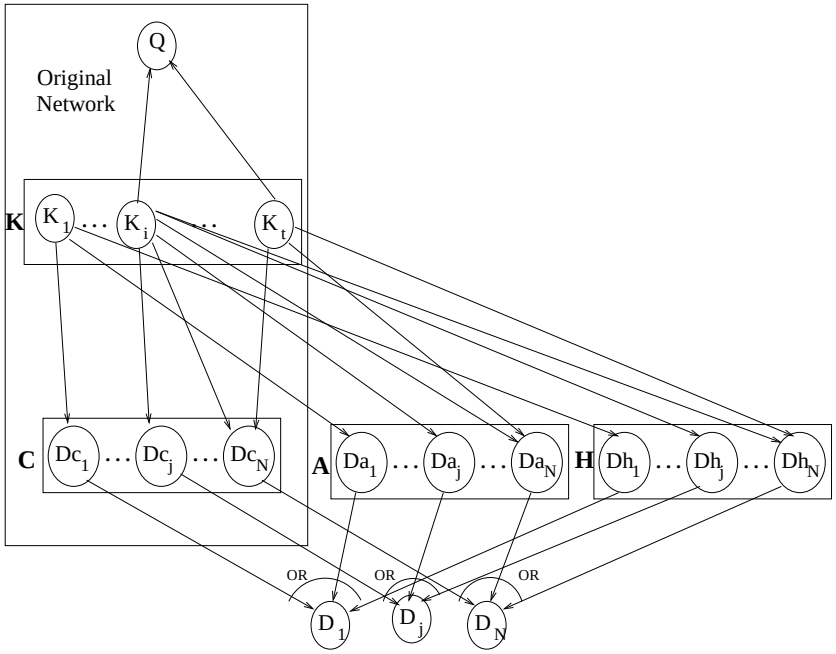


Figure 4.1: Belief network model, expanded with link-based evidence.

an authority. Thus, we now have three sets of nodes representing evidential knowledge associated with the documents in the network: the set **H**, composed of nodes representing hub evidence, the set **A**, composed of nodes representing authority evidence, and the set **C**, composed of nodes representing content-based evidence. The state of the associated random variables is given by **h**, **a**, and **c**, respectively.

The set of nodes **K** is used to model the occurrence of keywords in the query q and, once instantiated, induces believes on each of the nodes in the sets **C**, **H** and **A**. The propagation of these believes in the network is done according to the conditional probabilities governing the relationships between them. The specification of the conditional probabilities is based on the vector space model and on the HITS algorithm, as we later discuss.

The binary random variable Dh_j associated with each node Dh_j of **H** is

1 to indicate that the hub evidence associated with the document d_j is to be considered in the ranking computation. Also, the binary random variable Da_j associated with each node Da_j of $\mathbf{A}$ is set to 1 to indicate that the authority evidence associated with the document d_j is to be considered in the ranking computation. The node d_j represents the combination of content-based and link-based evidential knowledge, from the left and right hand sides of the network. The conditional probabilities, discussed below, define how these evidences are combined.

4.1.1 General Equation for Ranking Computation

As for the original network (see Section 3.5.2), the ranking of a document is computed by the network in Figure 4.1 as the probability $P(d_j|q)$, as follows:

$$P(d_j|q) = \eta \sum_{\mathbf{k}} P(d_j|\mathbf{k})\ P(q|\mathbf{k})\ P(\mathbf{k}) \tag{4.1}$$

where $\eta = 1/P(q)$ is a normalizing constant. However, the conditional probability $P(d_j|\mathbf{k})$ now depends on link-based and content-based pieces of evidence, combined through a disjunctive operator. This is accomplished as follows:

$$P(d_j|\mathbf{k}) \;=\; 1 - (1 - P(dc_j|\mathbf{k})) \times (1 - P(dh_j|\mathbf{k})) \times (1 - P(da_j|\mathbf{k})) \tag{4.2}$$

'This disjunction reflects the fact that it is enough that one evidence is observed for the final evidence to be observed. Substituting Equation (4.2) into Equation (4.1), we can write:

$$\begin{aligned} P(d_j|q) \;=\; \eta \sum_{\mathbf{k}} & [1 - (1 - P(dc_j|\mathbf{k})) \times (1 - P(dh_j|\mathbf{k})) \times \\ & (1 - P(da_j|\mathbf{k}))] \times P(q|\mathbf{k}) \times P(\mathbf{k}) \end{aligned} \tag{4.3}$$

The computation of the probability $P(d_j|\mathbf{k})$ depends on the states of the nodes Dc_j, Da_j, and Dh_j. The probabilities $P(dc_j|\mathbf{k})$, $P(da_j|\mathbf{k})$, and

$P(dh_j|\mathbf{k})$ can now be defined, establishing interesting alternatives for computing the rank of a document d_j with regard to a query q.

4.1.2 Ranking Computation

The belief network model can represent the vector model through proper specification of the conditional probabilities in the network, as discussed in Section 3.5.3. To simplify our notation, let R_{jq} be a reference to the vectorial rank of the document d_j with regard to a query q, computed according to our network model using Equation (3.28):

$$R_{jq} = \frac{\sum_{i=1}^{t} w_{ij} \cdot w_{iq}}{\sqrt{\sum_{i=1}^{t} w_{ij}^2} \sqrt{\sum_{i=1}^{t} w_{iq}^2}} \tag{4.4}$$

Further, let H_{jq} and A_{jq} be the hub and authority values, respectively, associated with document d_j, computed by the HITS algorithm presented in Section 3.2.2.

Case 1: Content-Based Ranking

For representing a ranking based solely on document content, we ignore the knowledge derived from the link structure. This is accomplished in our network model by defining:

$$P(dc_j|\mathbf{k}) = R_{jq};\ P(dh_j|\mathbf{k}) = 0;\ P(da_j|\mathbf{k}) = 0 \tag{4.5}$$

Applying Equations (3.27), (3.28), (3.29), and (4.5) into Equation (4.3), we obtain:

$$P(d_j|q) = \alpha \times R_{jq} \tag{4.6}$$

Therefore, the general network of Figure 4.1 naturally subsumes a ranking dictated by the vector space model.

Case 2: Ranking Based on Hub Evidential Knowledge

To represent a ranking that depends only on hub-based knowledge, we redefine the probabilities as follows:

$$P(dc_j|\mathbf{k}) = 0;\ P(dh_j|\mathbf{k}) = H_{jq};\ P(da_j|\mathbf{k}) = 0 \tag{4.7}$$

which allows ignoring information associated with content-based and authority-based evidence. Notice that the hub evidence associated with d_j is modeled as the conditional probability $P(dh_j|\mathbf{k})$, whose value is set to H_{jq}—the hub value of the document d_j with regard to the query q.

Applying Equations (3.27), (3.29), and (4.7) into Equation (4.3), we obtain

$$P(d_j|q) = \alpha \times H_{jq} \tag{4.8}$$

In this case, our network simply reproduces a ranking based on local hub values.

Case 3: Ranking Based on Authority Evidential Knowledge

In this case, we write:

$$P(dc_j|\mathbf{k}) = 0;\ P(dh_j|\mathbf{k}) = 0;\ P(da_j|\mathbf{k}) = A_{jq} \tag{4.9}$$

where A_{jq} is the authority value of the document d_j with regard to the query q.

Applying Equations (3.27), (3.29), and (4.9) into Equation (4.3) we obtain

$$P(d_j|q) = \eta \times A_{jq} \tag{4.10}$$

As a result, our network simply reproduces a ranking based on authority values.

Case 4: Combining Content-Based and Hub-Based Pieces of Evidence

We now discuss how our network model can be used to naturally combine keyword-based evidential knowledge with link-based evidential knowledge.

Making $P(dc_j|\mathbf{k}) = R_{jq}$, $P(dh_j|\mathbf{k}) = H_{jq}$, and $P(da_j|\mathbf{k}) = 0$ and applying Equations (3.27) and (3.29) into Equation (4.3), we obtain:

$$P(d_j|q) = \eta \times [1 - (1 - R_{jq}) \times (1 - H_{jq})] \tag{4.11}$$

Case 5: Combining Content-Based and Authority-Based Pieces of Evidence

Making $P(dc_j|\mathbf{k}) = R_{jq}$, $P(dh_j|\mathbf{k}) = 0$, and $P(da_j|\mathbf{k}) = A_{jq}$ and applying Equations (3.27) and (3.29) into Equation (4.3), we obtain:

$$P(d_j|q) = \eta \times [1 - (1 - R_{jq}) \times (1 - A_{jq})] \tag{4.12}$$

Case 6: Combining Content-Based, Hub-Based and Authority-Based Pieces of Evidence

Making $P(dc_j|\mathbf{k}) = R_{jq}$, $P(dh_j|\mathbf{k}) = H_{jq}$, and $P(da_j|\mathbf{k}) = A_{jq}$ and applying Equations (3.27) and (3.29) into Equation (4.3), we obtain:

$$P(d_j|q) = \eta \times [1 - (1 - R_{jq}) \times (1 - H_{jq}) \times (1 - A_{jq})] \tag{4.13}$$

4.1.3 Summary of Ranking Alternatives

Table 4.1 summarizes the six alternative rankings modeled in our network. In all cases, the values H_{jq} and A_{jq} might be derived either from local or global information. Notice that we do not consider the combination of only authority-based and hub-based pieces of evidence, since experimental results

indicated that this combination, without the use of content-based evidential information, is not promising.

Case	Ranking	$P(d_j\|q)$
1	Vector	$\eta \times R_{jq}$
2	Hub	$\eta \times H_{jq}$
3	Authority	$\eta \times A_{jq}$
4	Vector-Hub	$\eta \times [1 - (1 - R_{jq}) \times (1 - H_{jq})]$
5	Vector-Authority	$\eta \times [1 - (1 - R_{jq}) \times (1 - A_{jq})]$
6	Vector-Hub-Authority	$\eta \times [1 - (1 - R_{jq}) \times (1 - H_{jq}) \times (1 - A_{jq})]$

Table 4.1: Alternative rankings modeled in our belief network model.

It is interesting to note that, using our Bayesian Network model, a weighted combination of evidences can easily be accomplished. For instance, a Noisy-OR model [60] can be used, instead of a standard disjunction. The noise parameters would serve as weights for each evidence. Weight adjusting can then be used to refine the model, in order to increase the quality of the results, or to adapt it to the needs of specific users.

By using our Bayesian network model, we were able to highly improve Web document retrieval. The combination of the three sources of evidence, content similarity, and hub and authority values, provides better results, in terms of average precision, than the use of each one of them in isolation.

4.2 Using Local and Global Link Information

Besides the improvement of document retrieval, our Bayesian network model allows the study of the effects of using *local link information* versus *global*

link information for document ranking. As proposed in [87], local information refers to the information in the set of documents returned as answers to a user query, whereas global information refers to information extracted from all the documents in the collection. The HITS algorithm was originally restricted to local information [49], whereas the PageRank algorithm was originally used with global information [7].

Local information is topic dependent, i.e., it depends on the topic of the query. For this reason, algorithms that used local information may be free of the noise caused by documents that are irrelevant to the user's needs. On the other hand, global information techniques may capture some global characteristic of the collection, and need to be computed only once for the whole collection, which reduces query processing time. Each approach has its own advantages and disadvantages and it is, therefore, important to evaluate when each of them can be applied. Comparisons on the use of local and global information have been performed in [64, 87, 88], using only the textual content of documents. Here, we compare the use of global link information with the use of local link information.

To capture local information, we used the HITS algorithm just as described in Section 3.2.2. We applied the vector space model to a collection of Web pages, and used the 200 highest ranked documents as the root set for the algorithm. We call the values thereby computed the *local authority* and *local hub values*. To capture global information, we applied HITS to the whole collection. In this case, the root set and the base set are the same—the whole set of pages in the collection. We call the values thereby computed the *global authority* and *global hub* values.

For global information, we also used the PageRank algorithm. Unlike HITS, PageRank computes a single value for each page. Although it is im-

portant to note that there are fundamental differences between the values computed by PageRank and HITS (see Sections 3.2.1 and 3.2.2), the PageRank values can easily be inserted into our Bayesian network model. By replacing the global authority value of a page with its PageRank value, we can insert it in our model through the Vector-Authority combination equation (Equation (4.12)), or through the Vector-Hub-Authority combination equation (Equation (4.13)), where it can be combined with the HITS global hub value.

As seen in the following sections, local link information can provide a significant improvement in average precision figures and that global link information can provide a great increase in average precision at low recall levels. Also, the use of both HITS and PageRank allowed for an interesting comparison of the algorithms.

4.3 Experimental Results

In this section we evaluate each of the six ranking alternatives presented in Table 4.1, considering the adoption of either global or local hub and authority values. Our experiments are based on a collection of documents extracted from the World Wide Web.

4.3.1 The Reference Collection

Our reference collection is composed of a database of Web pages, a set of example Web queries, and a set of relevant documents associated with each example query. The database is composed of 5,939,061 pages of the Brazilian Web, under the domain ".br". The pages were automatically collected by the document collector described in [69], and indexed using inverted lists [85].

A total of 50 example queries were selected from a log of 100,000 queries submitted to the *TodoBR* search engine[1]. The queries selected were the 50 most frequent ones. Some frequent queries related to sex were not considered. The mean number of keywords per query is 1.78. Of the selected queries, 28 were quite general, like "tabs", "movies", or "mp3". Following, 14 queries were more specific, but still on a general topic, like "transgenic food", or "electronic commerce". Finally, 8 queries were quite specific, consisting mainly of music band names. Similarly to [39], for all queries, a description of what documents should be considered relevant was shown to the users. For instance, for query "employment", users were instructed to consider relevant only sites dedicated to employment adds.

Although this is a very small percentage of the total number of queries submitted to the TodoBR search engine, and although the set of most frequent queries may change with time due to shifts in the users' interest, the selected set does represent typical Web queries, in the sense that they are short, imprecise, and cover a wide variety of topics, as reported in [74].

For each of our 50 example queries we composed a query pool formed by the highest ranked 20 documents, as given by each of our 6 types of network ranking. Hub and authority values were computed using either global or local information, which led to a total of 11 different ranking strategies (the vector ranking is unaffected by the type of information considered). The characteristics of the database used are summarized in Table 4.2. Each query pool contained an average of 83.26 pages. All documents in each query pool were submitted to a manual evaluation by a group of 29 users, all of them familiar with Web searching. Users were allowed to follow links and evaluated the pages according not only to their textual content, but also to their linked

[1]Available at `http://www.todobr.com.br`.

pages and graphical content (flash, or dynamic HTML animations). Each user evaluated 3 different queries. The average number of relevant pages per query pool is 36. We adopted the same pooling method used for the Web-based collection of TREC [40, 41].

In our experiments, the local answer sets used consisted of the first 200 documents obtained with the vectorial ranking. For the local HITS algorithm, this set was expanded with its neighboring documents, as explained in Section 3.2.2. Also, to avoid self-reinforcement problems (i.e. a page getting high hub and authority values because it is linked with many pages within the same site, as discussed in [4, 52]), only links to pages belonging to outside sites are considered.

Number of pages	5,939,061
Number of distinct words	2,669,965
Number of words per page	413.5
Number of queries	50
Number of words per query	1.78
Number of pages per query pool	83.26
Number of relevant pages per query pool	36

Table 4.2: Characteristics of the database.

The reference collection has a total of 40,871,504 links, which yields an average of 6.9 links per page. The link distribution is very skewed, however. If we do not consider links between pages of the same site, about 97% of the pages have no links at all. Figure 4.2 shows the link distribution of the collection, on a logarithmic scale. It should be noted that, for those pages that do have links, the number of in-links is generally larger.

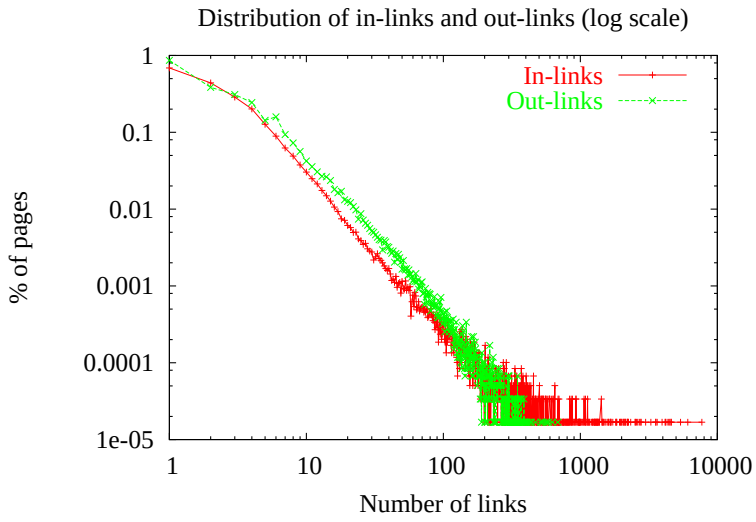


Figure 4.2: Distribution of links for the TodoBR Web collection, considering only links between pages on different sites. Although not shown in the figure, 97.7% of the pages have no in-links, while 97.2% of the pages have no out-links.

4.3.2 Ranking Using Local Link Information

We start by comparing and evaluating the 6 alternative rankings in Table 4.1, considering only local link information. All comparisons were made in terms of precision-recall figures [3]. Note that the recall values are relative to the set of evaluated documents, since we are not able to evaluate the entire collection.

Figure 4.3 illustrates the retrieval performance for the vector, hub, and authority rankings. We observe that the hub ranking is superior for our set of queries. This happens because good hubs are generally pages with a great number of links to other pages covering a particular subject, like pages from Web directories. Our test users considered these pages relevant when searching for information on a given subject. Thus, almost all pages highly ranked as hubs were taken as relevant pages. We also observe that the vector and authority rankings both have good precision values. This indicates that they should not be disregarded as important sources of evidence when ranking Web pages.

In Figure 4.4 we investigate the impact of combining the vector and authority rankings in our belief network model. The results indicate that this combination yields precision figures that are superior to those provided by each ranking in isolation. At low recall levels, the vector-authority ranking shows a small decrease in precision, when compared to the vector ranking. This happens because some pages, although pointed by many others (thus being good authorities), are unrelated to the query topic and therefore not relevant to the users. Nevertheless, the authority ranking contributes to improve the overall precision for all recall levels above 20%, where the vector ranking is not as good as it is at lower recall levels.

Figure 4.5 shows the impact of combining the vector and hub rankings

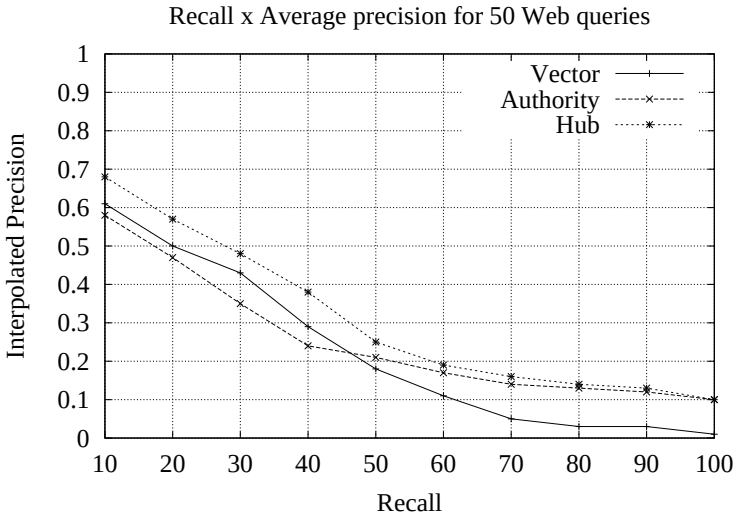


Figure 4.3: Average precision figures for vector, authority and hub rankings, using local link information.

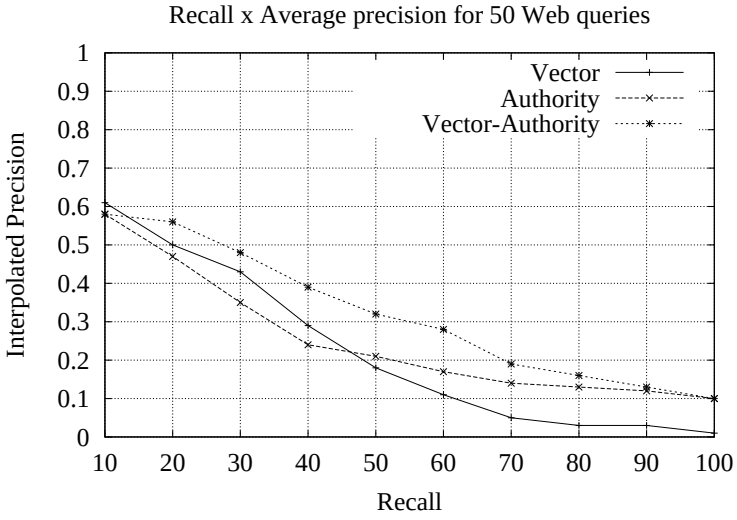


Figure 4.4: Average precision figures for vector, authority, and vector-authority network rankings, using local link information.

in our belief network model. Again we observe that this combination yields higher precision figures than those obtained by each ranking in isolation.

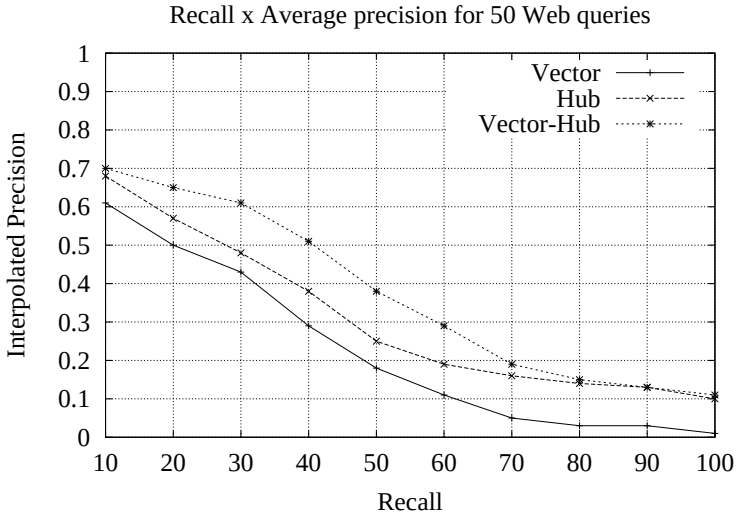


Figure 4.5: Average precision figures for vector, hub, and vector-hub network rankings, using local link information.

Finally, in Figure 4.6, we show the impact of combining the vector, authority, and hub rankings. This three-way combination of evidences yields superior results. At recall levels below 30%, the vector-hub combination has a slightly better performance, due to high relevance given by the users to pages with many links. At middle and high recall levels the vector-hub-authority combination shows a large improvement over the remaining rankings, showing that both hub and authority values are useful for determining the documents' relevancy.

These results confirm the preliminary results presented in [70], but now considering a much larger test collection. Further, they demonstrate that the belief network model is able to take advantage of the distinct nature of each of our three types of evidential knowledge to provide improved overall

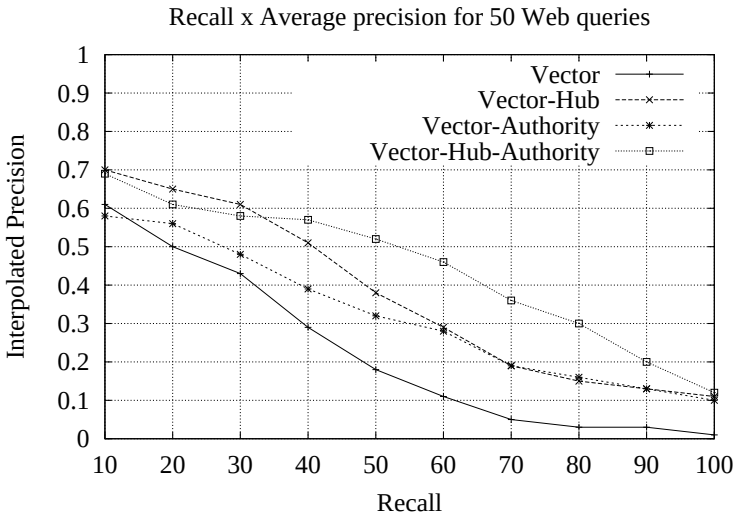


Figure 4.6: Average precision figures for vector, vector-hub, vector-authority, and vector-hub-authority network rankings, using local link information.

retrieval performance. This is an interesting result that indicates the strength of belief networks as a framework for consistently combining distinct pieces of evidence in support of a relevance ranking, a characteristic also observed in [61, 82], in distinct scenarios.

4.3.3 Local versus Global Link Information

We now examine the use of global link information and compare its results to those obtained by the use of local link information. Global link information has two main advantages regarding the use of link-based evidence to rank Web pages. First, authoritative evidence can be naturally seen as information of global nature, since pages are linked from other pages anywhere on the World Wide Web, independently of their topic. Second, global information can be computed once for the whole collection of documents, thus improving

time efficiency in answering user queries.

Figure 4.7 shows results for the vector-hub-authority ranking, when global and local information are considered. We see that the global ranking has a gain in precision, at low recall levels, superior to that of the local ranking. At high recall, precision for the global ranking drops below that of the local ranking, approaching the performance obtained by the vector space model.

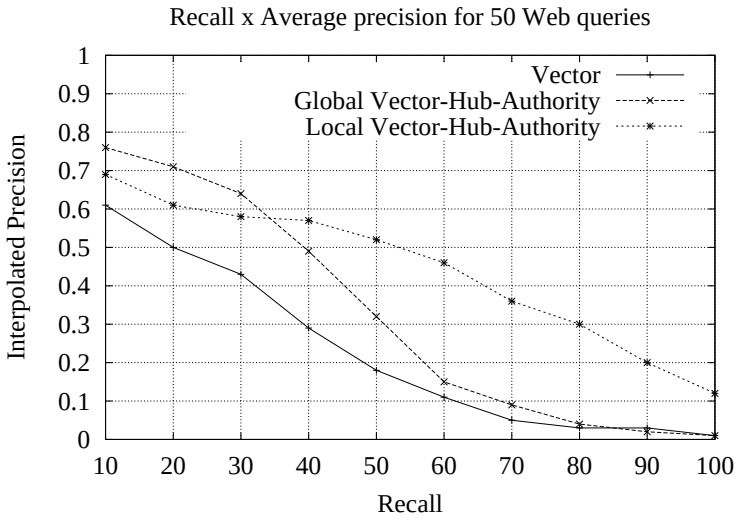


Figure 4.7: Average precision figures for vector, global vector-hub-authority, and local vector-hub-authority rankings.

The high precision at low recall levels shows that global information can be effectively used to improve the relevance of answers provided by Web search engines, whose users are mostly interested in the first 10 answers. On the other hand, due to the nature of the HITS algorithm, hub and authority values are somewhat divided among all pages in the collection. Therefore, for large collections, many pages get equal values that are very close to zero. At high recall levels, this makes the final combined global vector-hub-authority value dependent mainly on the vector part of the combination. The combined

ranking becomes, thus, very similar to the vectorial ranking, which explains the lower precision at high recall values.

Another problem happens if the collection consists of isolated groups of pages. In this case, HITS will assign zero to all but the main group. In our collection, HITS attributed hub and authority values to about 170,000 pages. Although a small percentage of the total collection, this value is enough to give significant improvements in the precision of the results. Since a small minority of pages (about 1%) has the great majority of in- and out-links (about 80%), the most important pages (or the pages corresponding to the most frequent queries) will probably be represented in this group. Notice that this distribution is not unusual in the Web, as can be seen in [51].

Local information retrieves a greater number of relevant documents, thus being ideal for searches were high recall is needed. Since more documents are introduced by the HITS algorithm during the construction of the base set (see Section 3.2.2), more noise is also introduced at the top of the ranking. Unlike global information, relevant documents are more evenly distributed throughout the answer set, causing a flip in the precision curve at 30% recall.

Table 4.3 shows the average gain in precision, compared to the vector space model, for the vector-hub-authority combination, measured at the 10, 20 and 30 first documents in the ranking. We observe that the vector-hub-authority combination using global information provides more relevant documents within the first 20 documents. For the first 10 documents, the gain in average precision obtained with the use of global link information is much higher than the gain obtained with the use of local link information. Therefore, the use of global evidence is useful mainly at low recall values. Also, unlike local information, global information has the advantage of not requiring any extra computation at query time. This makes it an interesting

alternative for systems where high precision is especially important for the first documents in the ranking and where query answer time is critical, such as the Web search engines.

Number of pages	Vector Precision	Local Information		Global Information	
		Precision	Gain	Precision	Gain
10	0.541	0.582	8%	0.671	24%
20	0.403	0.566	40%	0.617	53%
30	0.297	0.523	76%	0.478	61%

Table 4.3: Average precision figures for the top 10, 20, and 30 documents, when local link and global link information are considered.

4.3.4 Comparison with PageRank

Since PageRank is a popular method of link analysis using global information, it is interesting to compare its performance with that of the HITS algorithm. The PageRank score of a document was combined with its vectorial score through Equations (4.11) and (4.13), used for the Vector-Authority and Vector-Hub-Authority evidence combinations, respectively. For the later, global hub values were obtained by HITS. Figure 4.8 shows the resulting precision/recall curves.

Interestingly, the PageRank algorithm performs quite similarly to the HITS algorithm. When combined only with content-based information, its performance is very close to the performance of HITS, when only authorities and content are considered as evidence. By combining PageRank with hub evidence, the curve approximates that of the vector-hub-authority combination for the HITS algorithm. In fact, observing the documents returned by

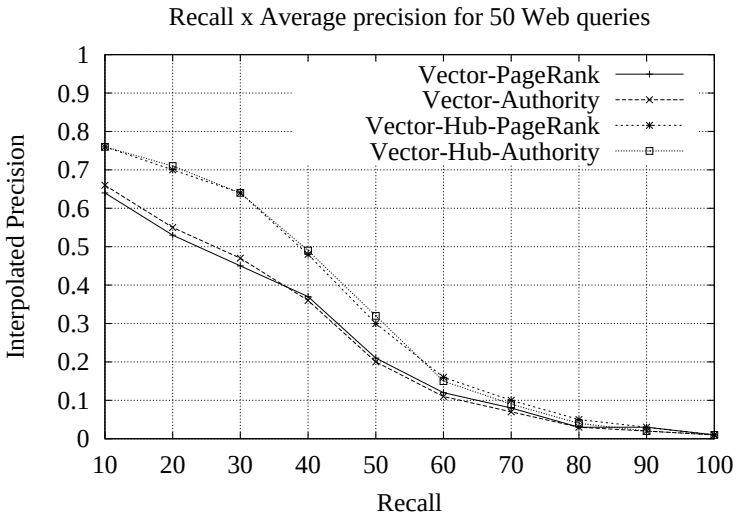


Figure 4.8: Average precision figures for the PageRank algorithm, compared to the HITS algorithm.

both algorithms, we see that most of the documents returned as authorities by HITS, are also present in the PageRank results, although in different positions.

This behavior can be explained. We can expect that many of the pages indicated as good authorities by HITS to have a high degree of in-links. This would also make them very likely to be visited by a surfer randomly following links, much more so on a collection with very skewed distribution of links, such as the Web. PageRank introduces random jumps to avoid circular links, where the random surfer would get caught in an endless cycle. However, in our collection, we removed links between pages within the same site, which greatly reduces the number of circular linking. Thus, we can expect PageRank and HITS to perform similarly.

4.3.5 Summary of Evaluation Results

Table 4.4 summarizes the results of our experiments using local sources of evidence. The table shows the gains in precision of the combined rankings relative to the vector space model. While the vector-authority ranking provides a gain of 31% in average precision, the vector-hub ranking yields a gain of 52% (this is because the test users considered hub pages highly relevant). Further, the combined vector-hub-authority ranking leads to a higher gain in average precision, close to 74%.

	Vector	Vec-aut		Vec-hub		Vec-hub-aut	
Recall	Precision	Precision	Gain	Precision	Gain	Precision	Gain
10	0.610	0.582	-5%	0.701	15%	0.689	13%
20	0.501	0.556	11%	0.647	29%	0.614	23%
30	0.427	0.477	12%	0.613	44%	0.582	36%
40	0.286	0.394	38%	0.511	79%	0.566	98%
50	0.185	0.322	74%	0.378	104%	0.524	183%
60	0.114	0.281	146%	0.295	159%	0.456	300%
70	0.054	0.195	261%	0.187	246%	0.357	561%
80	0.031	0.160	416%	0.150	384%	0.299	865%
90	0.026	0.132	408%	0.134	415%	0.204	685%
100	0.010	0.103	930%	0.106	960%	0.117	1070%
Average	0.267	0.351	31%	0.405	52%	0.466	74%

Table 4.4: Average precision figures for the vector, vector-authority, vector-hub, and vector-hub-authority network rankings, using local information.

Table 4.5 summarizes the results of our experiments using global sources of evidence. The largest gain in average precision is now 35%, obtained by

the vector-hub-authority combination. The vector-authority and vector-hub combinations yield gains in average precision of 10% and 29%, respectively. Even though average values are smaller than those obtained using local information, we see an improvement in precision for recall values below 40%. As discussed before, this means that relevant pages retrieved using global information are more concentrated at the top of the ranking, even if the answer set contains a smaller number of relevant pages.

	Vector	Vec-aut		Vec-hub		Vec-hub-aut	
Recall	Precision	Precision	Gain	Precision	Gain	Precision	Gain
10	0.610	0.659	8%	0.748	23%	0.756	24%
20	0.501	0.549	10%	0.688	37%	0.713	42%
30	0.427	0.472	11%	0.617	44%	0.637	49%
40	0.286	0.364	27%	0.411	44%	0.492	72%
50	0.185	0.201	9%	0.232	25%	0.323	75%
60	0.114	0.112	-2%	0.142	25%	0.150	32%
70	0.054	0.073	35%	0.072	33%	0.092	70%
80	0.031	0.029	-6%	0.035	13%	0.043	39%
90	0.026	0.024	-8%	0.026	0%	0.024	-8%
100	0.010	0.010	0%	0.010	0%	0.010	0%
Average	0.267	0.292	10%	0.345	29%	0.360	35%

Table 4.5: Average precision figures for the vector, vector-authority, vector-hub, and vector-hub-authority network rankings, using global information.

In conclusion, our results show that combining content-based and link-based sources of evidence yields better retrieval results than using any of them separately. In fact, local link information yielded a gain in precision of 74% when compared with the results of the vector space model. Global

link information yielded a gain in precision of 35% for our test collection. These results suggest that, in general, the use of local link information is more promising.

Interestingly, global information was shown to be useful in improving retrieval results at low recall values. For the first 10 documents in the ranking, the use of global link information produced an average gain in precision of 28%, whereas the use of local information showed a gain of only 8%. Also, global information required no extra processing at query time. These characteristics make the use of global sources of evidence a valuable alternative whenever high precision at low recall is important and query processing efficiency is essential, such as in Web search engines.

It is important to note that previous results in the TREC Web track [39] seem to indicate that the use of link analysis brings little gains to the task of document ranking. However, several fundamental points in the TREC experiments make the testing environment different from ours. First, documents in the TREC collection were judged only according to their text, and judges were not allowed to follow links. In our collection, on the other hand, the judges had access to the real site, including its multimedia content, and were allowed to follow links. Thus, most pages classified as relevant were hub pages. Site homepages constituted only 36% of all the relevant documents. Second, link analysis algorithms are expected to work better for more general queries. Since the most frequent queries in the Web are, usually, quite general, we can expect good results from link-based retrieval. TREC Web queries, however, tend to be very specific, thus harming the effectiveness of such approach. Third, our collection was collected by Web crawlers, whereas TREC was constructed as a subset of a larger collection. Although care was taken in TREC to assure good linkage among pages, it is very likely that the

link distribution is quite different from the Web, thus causing some disparity in the results.

Chapter 5

Classifying Web Documents by Combining Link-Based and Content-Based Information

Content-based classifiers are known to perform poorly in the Web [13, 37]. Web documents are not only noisy and with little text, but also contain images, scripts and other types of data unusable by text classifiers. Furthermore, they can be created by many different authors, with no coherence in style, language or structure. For this reason, other types of evidence besides textual content should be used in an attempt to improve classification. In this chapter, we discuss how link-based information can be combined with content-based information used by traditional classifiers and explore how this combination can be used to improve classification effectiveness. To allow combining distinct sources of information in a consistent fashion, we adopt the framework of Bayesian networks. Our Bayesian network model provides a modeling tool that is independent of the link measures and of the classification algorithms used.

5.1 Combining Link and Content-Based Information

To combine link-based and content-based information, we propose the use of the Bayesian network model shown in Figure 5.1. The root nodes, labelled D_1 through D_N, represent our prior knowledge about the problem, i.e., a set of classified documents (the training set). Node C represents a category. Since a category is defined by a set of classified documents, there are edges from nodes D_j to node C, representing the fact that observing a set of training documents will influence the observation of a category.

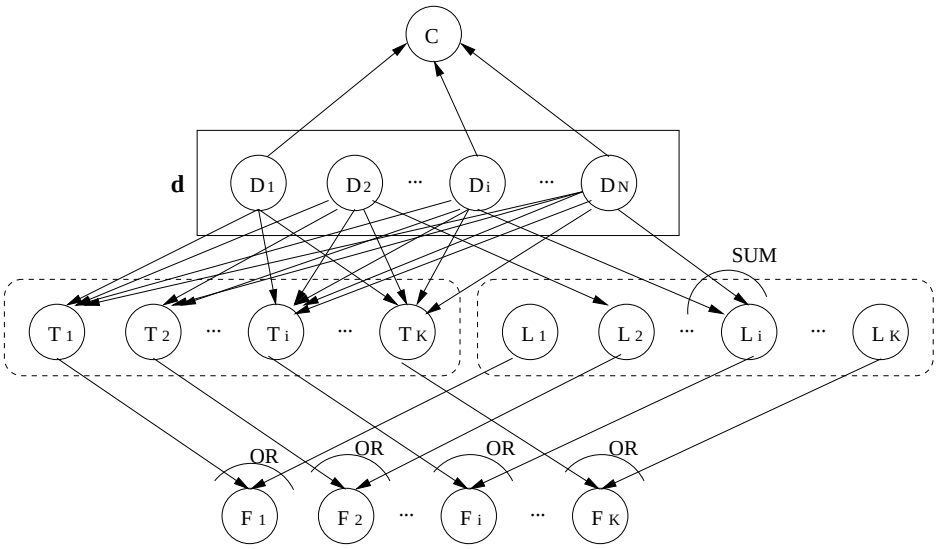


Figure 5.1: Bayesian network model to combine evidence from a content-based classifier with evidence from the link structure.

Nodes T_1 through T_K and L_1 through L_K represent the documents to be classified (the test set) under two different contexts. Each node T_i represents evidence obtained from the content-based classifier, indicating that the test document d_i belongs to category C. Since this evidence depends on the set

of training documents, there are edges from each node D_j to every node T_i. Thus, observing a set of training documents will influence the fact that we observe the test document d_i as belonging to category C.

Similarly, each node L_i represents link-based evidence that document d_i belongs to category C. Each edge from a node D_j to a node L_i represents evidence, obtained using a link-based metric, that the training document d_j is related to the test document d_i. We say that two documents d_i and d_j are related if their linkage similarity is greater than zero, as given by one of the link similarities described in Section 3.3. Our motivation here is as follows. If we observe a subset of training documents that are classified as belonging to category C, then there are grounds to infer that document d_i should be also considered as a candidate for category C.

Finally, nodes F_1 through F_K represent the final evidence that each test document belongs to category C. This evidence depends on both the content-based and the link based evidences, as shown by the incoming edges.

Given these definitions, we can now use the network to determine the probability that a test document d_i belongs to category C, i.e., the probability of observing the final evidence regarding document d_i, given that category C was observed, $P(F_i = 1|C = 1)$. This translates to the following equation:

$$P(f_i|c) = \eta \sum_{\mathbf{d}} P(f_i|\mathbf{d})\ P(c|\mathbf{d})\ P(\mathbf{d}) \tag{5.1}$$

where $\eta = 1/P(c)$ is a normalizing constant and $\mathbf{d}$ is a possible state of all the variables D_j.

The probability $P(f_i|\mathbf{d})$ represents the combination of content-based and link-based evidences. For our experiments, we define $P(f_i|\mathbf{d})$ as a disjunction of the evidences generated by the content-based classification algorithm used and the link-based metrics. This means that, for the final evidence to

be observed, it is enough to observe one of the content-based or link-based evidences. Eq. (5.1) thus becomes:

$$P(f_i|c) = \eta \sum_{\mathbf{d}} \Big(1 - \big(1 - P(t_i|\mathbf{d})\big)\big(1 - P(l_i|\mathbf{d})\big)\Big) P(c|\mathbf{d}) P(\mathbf{d}) \tag{5.2}$$

Eq. (5.2) is the general equation to compute the probability of a document belonging to a given category. We now need to define the probabilities $P(t_i|\mathbf{d})$, $P(l_i|\mathbf{d})$, $P(c|\mathbf{d})$, and $P(\mathbf{d})$.

5.1.1 Computing the Classifications

We start by defining the probability $P(c|\mathbf{d})$, which is used to select only the training documents that belong to the category we want to process. We define $P(c|\mathbf{d})$ as:

$$P(c|\mathbf{d}) = \begin{cases} 1 & \text{if } \forall_i,\ D_i = 1 \text{ iff } d_i \in \mathcal{C} \\ 0 & \text{if otherwise} \end{cases} \tag{5.3}$$

where $\mathcal{C}$ is the set of training documents that belong to category C.

The probability $P(t_i|\mathbf{d})$ that document d_i belongs to category C, given that the set of documents indicated by $\mathbf{d}$ was observed, can now be defined. Let $\mathcal{C}$ be a set of documents labelled as belonging to category C, let $\overline{\mathcal{C}}$ be a set of documents labelled as not belonging to C, and let $class(i, \mathcal{C}, \overline{\mathcal{C}})$ be a function that returns a value of association between document d_i and category C, based on the labelled document sets. We define:

$$P(t_i|\mathbf{d}) = class(i, \mathcal{C}, \overline{\mathcal{C}}) \tag{5.4}$$

The function *class* represents a content-based classifier and, for our experiments, the returned value is given by either the *kNN*, the SVM, or the Naive Bayes algorithms. We assume that this value is normalized such that $0 \leq class(i, \mathcal{C}, \overline{\mathcal{C}}) \leq 1$.

To define the probability $P(l_i|\mathbf{d})$, let $\mathcal{V}(i)$ be the set of training documents related to document d_i (notice that these documents are represented by the parent nodes of node L_i). Let $link(i,j)$ be the similarity between document d_i and document d_j, as given by one of the linkage similarity measures described in Section 3.3. We define:

$$P(l_i|\mathbf{d}) = \alpha \sum_{d_j \in \mathcal{V}(i) \wedge d_j \in \mathcal{C}} link(i,j) \tag{5.5}$$

where α is a normalizing constant used to keep the sum between 0 and 1. Evidence represented by L_i is, therefore, defined as the sum of the values given by the linkage similarity between document d_i and all of its related documents that are indicated by $\mathbf{d}$ as observed. Thus, the more training documents related to d_i belong to a given category, the greater the probability that d_i belongs to the same category.

Finally, since we have no initial preference as to what set of training documents is more probable of being observed, we can regard the *a priori* probability $P(\mathbf{d})$ as a constant. By applying Eqs. (5.4), (5.5), and (5.3) to Eq. (5.2) we obtain the final equation to compute the probability that document d_i belongs to category C:

$$P(f_i|c) = \rho\Big(1 - \big(1 - class(i, \mathcal{C}, \overline{\mathcal{C}})\big)\big(1 - \alpha \sum_{j \in \mathcal{V}(i) \wedge j \in \mathcal{C}} link(i,j)\big)\Big) \tag{5.6}$$

where $\rho = P(\mathbf{d})/P(c)$ is a normalizing constant and $\mathbf{d}$ is the state where only the documents labelled as belonging to class C are active.

5.1.2 Weighted Evidence Combination

So far, the combination model makes no *a priori* assumption about the importance of each source of evidence. The probabilities to be combined depend only on the characteristics of the algorithms used and on the parameters used

to configure them. Thus, when evaluating if a document d_i belongs to a category C, more weight will be given to the source of evidence that yields the highest probability of d_i belonging to C. However, the model can be modified to allow the insertion of user-defined weights, in order to fine tune the classification and provide adaptation to different collections.

The introduction of weights in the model can be accomplished by the use of a noisy-OR combination [60], instead of the simple disjunction defined by the probabilities $P(f_i|\mathbf{d})$ in Eq. (5.2). This yields the final equation:

$$P(f_i|c) = \rho\Big(1-\big(1-W_t\times class(i,\mathcal{C},\overline{\mathcal{C}})\big)\big(1-W_l\times\alpha \sum_{j\in\mathcal{V}(i)\wedge j\in\mathcal{C}} link(i,j)\big)\Big) \quad (5.7)$$

where W_t and W_l are the weights given to the content-based classifier and the link structure evidences, respectively. Notice that, in the noisy-OR model, $1-W_t$ and $1-W_l$ are the so called noise parameters.

5.2 Experiments

To evaluate the effectiveness of the combination model, a set of tasks was performed on a collection of classified Web pages. We now describe the experimental setup and discuss the results achieved.

5.2.1 The Test Collection

We performed experiments using a set of classified Web pages extracted from the Cadê Web directory[1]. This directory points to Brazilian Web pages that were classified by human experts. The Cadê directory contains actually a subset of the TodoBR collection described in Section 4.3.1, which was used to obtain the content of the classified pages.

[1] Available at `http://www.cade.com.br`.

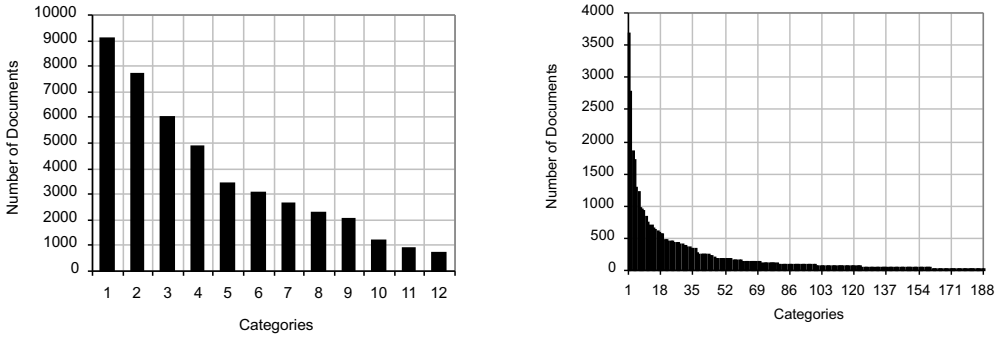
We constructed two sub-collections using the data available on Cadê: Cade12 and Cade188. Cade12 is a set of 44,099 pages labelled using the 12 first level categories of Cadê (Computers, Culture, Education, Health, Internet, News, Recreation, Science, Services, Shopping, Society, and Sports). Cade188 is a subset of Cade12, without the pages classified only in the first level category[2] . Thus, Cade188 corresponds to a set of 42,123 pages relabelled using the 188 second level categories of Cadê (Biology, Chemistry, Dance, Music, Schools, Universities, etc.). Each Web page is classified into only one category.

Figures 5.2(a), 5.2(b), and 5.3 show the category distributions for the Cade12 and Cade188 collections. Notice that both have skewed distributions. In Cade12, the three most popular categories represent more than 50% of all documents. The most popular category, *Services*, has 9,081 documents while the least popular, *Shopping*, has 715 documents. In Cade188, 50% of the documents are in just 10% of the categories. The most popular category, *Society:People*, has 3,675 documents while the least popular, *Internet:Tutorials*, has 24 documents. Cade12 and Cade188 have vocabularies of 191,962 and 168,257 unique words, respectively, after removing stop words.

Information about the links related to the Cadê pages was also extracted from the TodoBR collection. Table 5.1 summarizes the link data obtained. It was divided into two types: the *internal links*, which are links between pages classified by Cadê, and the *external links*, which are links where the target or the source page is in TodoBR, but not in the set of pages classified by Cadê. This distinction is important to verify whether the external information provided by TodoBR can be used to improve the results.

We call *directory pages* those that belong to the Cadê site itself and are

[2] 1,976 documents were classified only in one of the 12 first level categories.



(a) Cade12 (b) Cade188

Figure 5.2: Category distribution for the Cade12 and Cade188 collections.

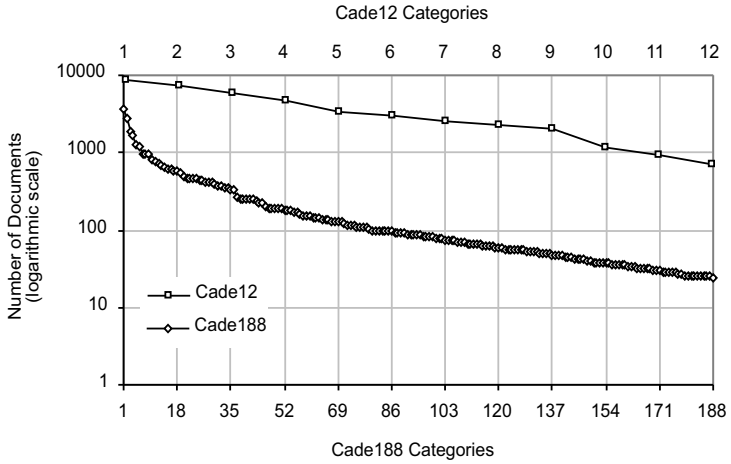


Figure 5.3: Compared distributions for Cade12 and Cade188.

used to compose the directory hierarchy. For instance, the Cadê Science page, which links to science related sites is a directory page. As can be seen, directory pages represent a great part of the internal links in the Cadê collection. Since directory pages provide information on the categories of the remaining pages (for instance, the Science page links only to science related pages), they were not used for calculating the link information measures in our experiments. For the same reason, all the pages found in the TodoBR collection that were *similar* to Cadê pages were removed. A page in TodoBR was considered similar to a page in Cadê if both had more than 70% of the out-links in common. These pages were, usually, illegal copies of the Cadê directory.

Table 5.1 also shows that external pages provide a rich source of link data. About 90% of the Cadê pages are linked to by external pages while less than 4% link to external pages. This was an important reason for using Cadê in our experiments. With Cadê, we can obtain information about external links extracted from TodoBR and verify how useful this information can be for the classification process. This is only possible because Cadê is a subset of TodoBR, which is a large collection containing most of the link information available in Brazilian Web pages. This is not the case with most other classification collections where, in order to obtain more link information, it would be necessary to collect a huge amount of Web pages, or to have access to another search engine database, as we did with TodoBR.

5.2.2 Methodology

To perform the experiments, we used the 10-fold cross validation method [76]. Each dataset was randomly split in ten parts, such that, in each run, a different part was used as a test set while the remaining were used as a training

Statistics	Whole Cadê	Cadê without Directory Pages
Internal Links	45,548	3,830
Links from external pages to Cadê pages	570,404	554,592
Links from Cadê pages to external pages	7,584	5,894
Cadê pages with no in-links	2,556	4,392
Cadê pages with no out-links	40,917	40,723

Table 5.1: Link statistics for the Cadê collection.

set. The split on training and test sets was the same in all experiments. The final results of each experiment represent the average of the ten runs. The 10-fold cross-validation method was chosen because it minimizes the likelihood that the differences in the results of the algorithms being compared is due to a bias in the training set. Since we are only interested in comparing the performance of different classification strategies, this method is appropriate. In a different application, however, other training set selection methods can be used [50].

To make sure that the results are not biased by an inappropriate choice of parameters, several experiments were performed and, in all cases, we report the best results obtained. Thus, for the *kNN* classifier, the value of k was set to 30 and 15,000 features were considered. For the SVM classifier, a linear kernel was used and 10,000 features were considered. For the Naive Bayes classifier, 15,000 features were considered. For all algorithms, the features were selected using the information gain method [57].

The performance of the presented methods was evaluated using the conventional precision, recall and F_1 measures, described in Section 3.6. To compute the final F_1 values, we used macro-averaging and micro-averaging.

For macro-averaging, recall, precision, and F_1 scores were first computed for individual categories and then averaged over all categories. For micro-averaging, the decisions for all categories were counted in a joint pool. We note that, since the datasets used in the experiments have a single label per document, micro-averaged recall, precision and F_1 are the same.

5.2.3 Experimental Results

5.2.4 Evaluation of Each Source of Evidence

We started by analyzing how each source of information, content and links, performs when used in isolation. This can be accomplished by setting the weights W_t or W_l to zero, in Eq. (5.7), according to the evidence to be tested. Table 5.2 shows the micro-averaged F_1 values for the content-based classifiers and linkage similarity measures used in isolation. The linkage similarity measures were computed either using only internal links (marked (i)) and both internal and external links (marked (i+e)). The macro-averaged precision, recall, and F_1 values are shown in Table 5.3. In this case, linkage similarities were computed only using both internal and external links. In both tables, the highest F_1 values for each classifier and similarity measure are shown in bold face.

The content-based classifiers, as expected, show poor results. In terms of micro-averaged F_1, the best results were achieved by SVM on the Cade12 collection and by *kNN* on the Cade188 collection, with values of 40.86 and 24.45, respectively. As seen in Table 5.3, these results are mainly due to very low recall values, indicating that the text of most Web directory pages does not provide enough information for a reliable classification. Both micro-averaged and macro-averaged values were lower for the Cade188 collection because

Source of evidence	Cade12	Cade188
kNN	39.45	**24.45**
SVM	**40.86**	24.31
Naive Bayes	39.38	22.82
Bibliographic coupling (i)	13.61	0.89
Amsler (i)	14.00	1.23
Co-citation (i)	13.84	1.11
Companion (i)	**14.30**	**1.44**
Bibliographic coupling (i+e)	13.70	0.94
Amsler (i+e)	69.53	55.38
Co-citation (i+e)	**69.64**	55.80
Companion (i+e)	67.88	**61.03**

Table 5.2: Micro-averaged F_1 measures obtained using the evidence provided by the content-based classifiers and linkage similarity measures, when used in isolation. For the linkage measures, the mark (i) stands for using only internal links and the mark (i+e) stands for using both internal and external links.

Source of evidence	Cade12			Cade188		
	P	R	F_1	P	R	F_1
kNN	60.81	27.31	32.14	68.04	14.10	**18.00**
SVM	56.29	28.58	33.24	42.14	13.52	15.38
Naive Bayes	47.67	32.59	**35.55**	80.58	8.05	9.49
Bibliographic coupling	16.70	8.81	3.14	70.59	2.01	1.43
Amsler	72.59	70.77	71.08	60.84	63.26	58.01
Co-citation	73.15	70.83	**71.40**	61.52	63.20	58.32
Companion	67.64	70.40	67.24	66.60	65.94	**61.36**

Table 5.3: Macro-averaged precision (P), recall (R), and F_1 measures obtained using the evidence provided by the content-based classifiers and linkage similarity measures, when used in isolation. For the linkage measures both internal and external links were used.

classifiers tend to perform worst in collections where the class distribution is more skewed.

For the linkage similarity measures, when only internal links are available, information is clearly insufficient, yielding very low F_1 values. By considering only internal links much of the link structure information of the collection is lost. In fact, as shown in Table 5.1, about 98% of the link information in the collection comes from external pages.

When links to and from external pages are used, however, link information alone was enough to achieve classification results well above those achieved by the content-based classifiers. In the Cade12 collection, the best results were obtained using the co-citation measure, with 69.64 points in micro-averaged F_1. In the Cade188 collection, the Companion algorithm had the best performance, with 61.03 points in micro-averaged F_1. Unlike

the content-based classifiers, the link-based metrics yielded both high recall and precision, as shown in Table 5.3. It is interesting to note that the linkage similarity measures had higher macro-averaged than micro-averaged F_1 values. This indicates that links are more accurate than text when distributing documents among the classes.

Bibliographic coupling yielded lower F_1 values than the remaining measures. This is not surprising since it relies only on out-link information and, as shown in Section 5.2.1, more than 90% of the pages have no out-links. Since most of the links are *from* external pages *to* pages in the collection, we can expect measures that make use of in-links to perform the best. For this reason, we do not consider bibliographic coupling in the following sections.

5.2.5 Results of Evidence Combination

We now verify the effects of combining both evidences, using our proposed Bayesian Network model. To study how different weights affect the classification results, we applied Eq. (5.7) and varied the weights W_t and W_l for the link-based and content-based evidences, respectively. Since using only internal link information yielded very poor results, we now proceed by using both internal and external links.

Figures 5.4 through 5.6 show the resulting micro-averaged F_1 values for the combination of each linkage similarity measures on the Cade12 collection. In all graphs, the YY axis shows micro-averaged F_1 and the XX axis shows the ratio between the link-based evidence weight and the content-based evidence weight (W_t/W_l), on a logarithmic scale. Curves labeled "*kNN*" , "SVM", and "Naive Bayes" represent the micro-averaged F_1 value for the combination of the indicated similarity measure with each of the content-based classifiers. The line labeled "link baseline" represents the result of

the indicated linkage similarity when used in isolation. The baselines for the content-based classifiers used in isolation are shown with labels "kNN baseline", "SVM baseline", and "NB baseline".

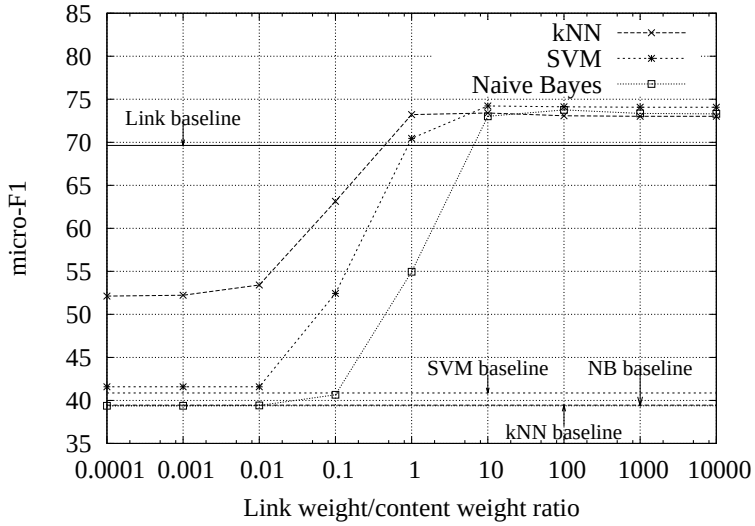


Figure 5.4: Effects of weighted combination for the co-citation similarity in the Cade12 collection.

It can be seen that results improve when the weight given to the link-based evidence is increased. For the co-citation and Amsler similarities F_1 values always show improvements over the link baseline when link weights are more than 10 times greater than content weights. For the Companion measure, this happens at link weights greater than 100 times the content weight. This confirms that links are a more reliable source of evidence than the pages' content. For the Cade188 collection, results are much similar, as shown by the graphs in Figures 5.7 through 5.9. In this case, however, F_1 values above the link baseline are achieved with lower link evidence weights.

It should be noted that, for both collections, the F_1 values for the combination do not converge to the content baseline when the weights given to the

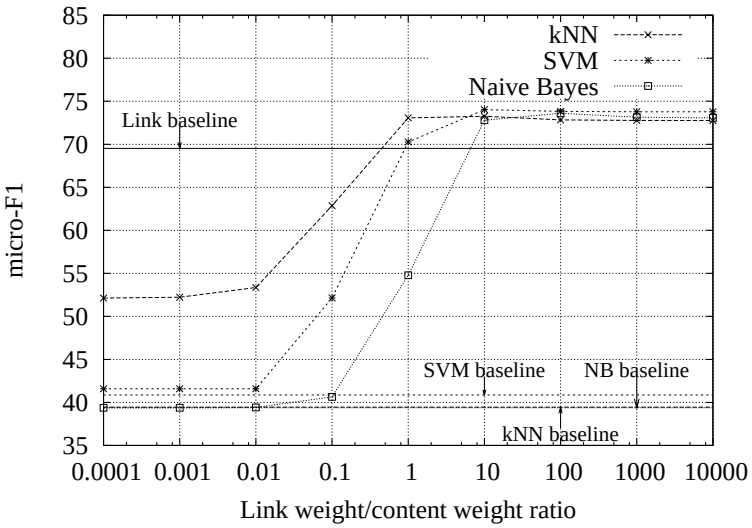


Figure 5.5: Effects of weighted combination for the Amsler, similarity in the Cade12 collection.

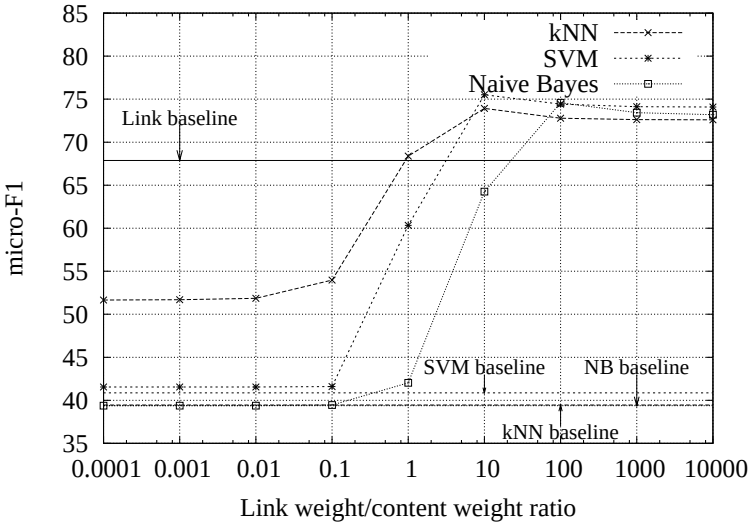


Figure 5.6: Effects of weighted combination for the Companion similarity in the Cade12 collection.

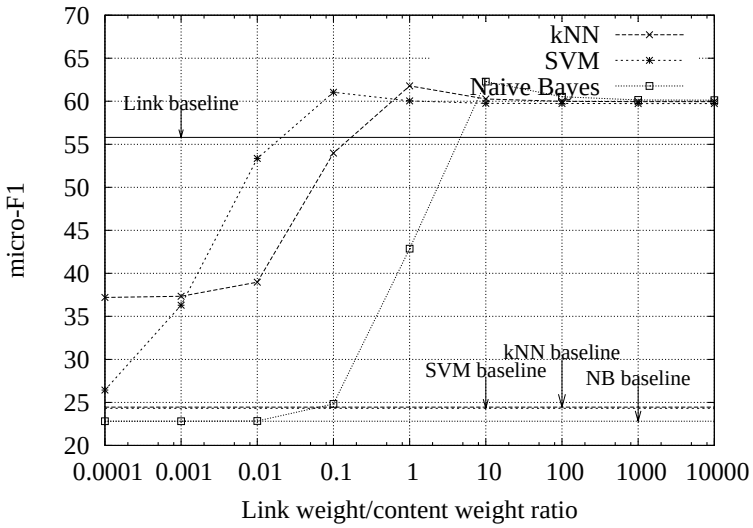


Figure 5.7: Effects of weighted combination for the co-citation similarity in the Cade188 collection.

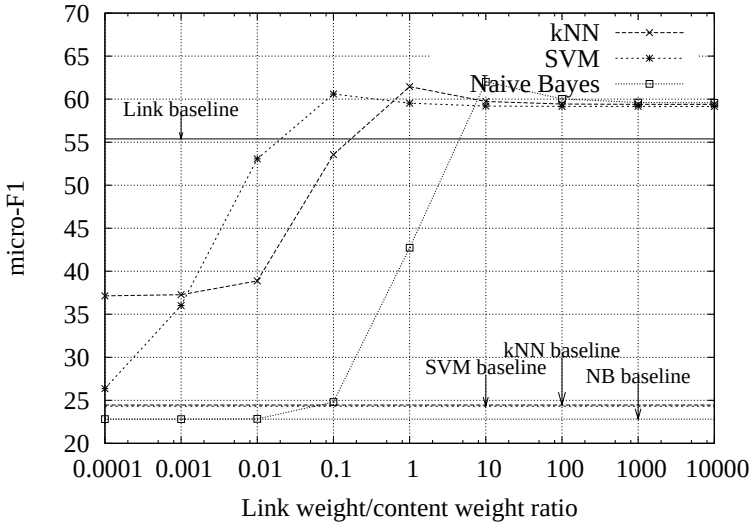


Figure 5.8: Effects of weighted combination for the Amsler similarity in the Cade188 collection.

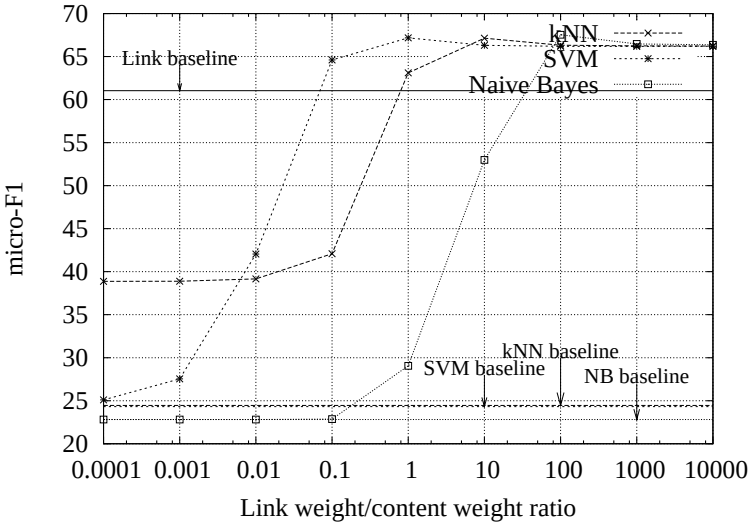


Figure 5.9: Effects of weighted combination for the Companion similarity in the Cade188 collection.

link-based sources evidence are very low. This happens because, when the content-based classifier computes the probability of a document belonging to a class, there is often a tie between the most probable classes. Thus, it is the link similarity measures who solve this tie and choose the correct class for the document, even when the link weight is very low. The same is observed regarding the convergence to the link baseline.

Table 5.4 summarizes the best micro-averaged F_1 values obtained by using weighted evidence combination. The results again indicate that links have more importance than text in the combination, since differences in F_1 using distinct content-based classifiers are very small. Interestingly, in both collections, the best results were achieved not always by combining the best sources of evidence, but by combining the Companion algorithm with the SVM classifier.

All measures were capable of capturing pertinent information regarding

Linkage Similarity Measures	Cade12			Cade188		
	kNN	SVM	NB	*kNN*	SVM	NB
Amsler	73.55	74.02	73.85	61.70	60.60	62.01
Co-citation	73.71	74.28	73.99	61.97	61.05	62.29
Companion	**74.16**	**75.51**	**74.79**	**67.47**	**68.10**	**67.94**
Content baseline	39.45	40.86	39.38	24.45	24.31	22.82

Table 5.4: Best micro-averaged F_1 measures obtained with the three classifiers,in the Cade12 and Cade188 collections, using weighted evidence combination.

the links in the collection. However, the Companion algorithm performed consistently better than the remaining measures, providing the highest gains in precision independently of the classifier used. These gains are more evident in Cade188, where links are a much noisier source of information (about 20% of the links in Cade12 link to documents in the same class, whereas in Cade188 only about 10% of the links link to documents in the same class).

5.2.6 Summary of Evaluation Results

Table 5.5 summarizes the results achieved by the proposed evidence combination model, for the Cade12 and Cade188 collections. For each linkage similarity measure, the table shows the best F_1 values obtained, the corresponding link weight/content weight ratio, and the classifier used. All measures are considered using both internal and external links. Columns *Gain/text* and *Gain/link* show, respectively, the gain in micro-averaged F_1 over the best text classifier and over the best linkage similarity measure used in isolation.

In the Cade12 collection, the best overall results were obtained by com-

Collection	Measures	W_l/W_t	Class.	F_1	Gain/text	Gain/link
Cade12	Amsler	10	SVM	74.02	81%	6%
	Co-citation	20	SVM	74.28	82%	7%
	Companion	10	SVM	75.51	85%	8%
Cade188	Amsler	10	NB	62.01	154%	2%
	Co-citation	10	NB	62.29	155%	2%
	Companion	0.2	SVM	68.10	176%	11%

Table 5.5: Best results achieved by each measure when using weighted evidence combinations on the Cade12 and Cade188 collections.

bining the SVM classifier with the Companion algorithm. The content-based evidence was given 10 times less weight than the link-based evidence. Results were similar for the remaining similarity measures, only with slightly lower F_1 values. Although in all cases the combination yielded a high gain over the best text classifier, improvements over the best linkage similarity measure were much smaller, ranging from 6% to 8%.

In the Cade188 collection, the combination of the Companion algorithm with the SVM classifier also yielded the highest F_1 values and the highest gain over any of the sources of evidence used in isolation. The Companion algorithm showed results superior the other linkage similarity measures, with a gain of 11% in F_1, which is over 5 times higher than the gain obtained by the Amsler and co-citation similarities.

Given these values, we can conclude that links are indeed a valuable source of information for Web classification. However, to achieve expressive results, external links should be used. In the Cade12 collection, the best similarity measure used in isolation was the co-citation similarity, which yielded a gain of 70% over the best content-based classifier (SVM). In the Cade188

collection, the best similarity measure used in isolation was the Companion algorithm, which yielded a gain of 150% over the best content-based classifier (kNN). The bibliographic coupling similarity showed an inferior performance, due to the fact that it uses only out-link information and most pages in the collection have no out-links.

By combining both sources of evidence, further gains can be achieved. Gains over the link-based measures used in isolation, however, are not as expressive as gains over the content-based classifiers. The Companion algorithm seems to provide the most robust similarity measure, showing the highest F_1 values in combination with any of the content-based classifiers. Gains of 8% and 11% over the best similarity measures used in isolation where achieved in the Cade12 and Cade188 collections, respectively.

The fact that link based measures showed a recall as high as, or higher than, precision, whereas the the content-based classifiers showed a recall much lower than precision suggests that: (1) link information can correctly classify a large number of documents, but can also introduce noise; and (2) content-based information can filter out some of the noise, but can also remove documents that would be correctly classified by the linkage similarities. By regulating the weight given to each source of evidence, this effect can be balanced to improve classification accuracy.

Chapter 6

Conclusions and Future Work

In this chapter, we present a brief summary of the achievements of this work. In Section 6.1, a final analysis of the results is presented and some conclusions are drawn. Following, in Section 6.2, some future work is suggested to complement this work and solve problems left open.

6.1 Conclusions

In this work, a study of the application of link information to two different information retrieval problems, ranking and classification in the Web, was performed. Through the use of Bayesian network models, link information was combined with traditional IR techniques, and the effects of such combination were explored.

For the ranking problem, we intended to study how link information can be used to improve the precision of ranking algorithms for Web documents and evaluate the effects of global and local link information on the quality of the results. Two different link-based ranking algorithms, PageRank and HITS, were studied and combined with a traditional vector space ranking

strategy. This combination was achieved through the use of a Bayesian network model.

Experiments performed on a Web collection allowed us to draw four main conclusions:

1. Links are, in fact, a useful source of evidence for the Web ranking problem;

2. The proposed combination model is able to capture information from the two sources of evidence, text and links, and effectively combine them to achieve a ranking of improved quality. In fact, results show that the use of the combined sources of evidence supersedes the use of each source of evidence in isolation;

3. Global link information yields higher precision at the top of the ranking. It is, thus, appropriate for systems like Web search engines; and

4. On average, local link information provides better overall results than global link information.

For the classification problem, we intended to study how link information can be used to improve the results of traditional content-based classifiers. Four different linkage similarity measures, bibliographic coupling, co-citation, Amsler, and Companion, were used in combination with three traditional classifiers, Naive Bayes, *kNN*, and Support Vector Machine. The combination was also achieved through a Bayesian network model.

Experiments were performed in two different collections extracted from a Web directory. Results allowed us to draw the following conclusions:

1. Links provide highly valuable information on the similarity of Web pages. In fact, links alone show better performance on both collections than any of the content-based classifiers;

2. The effectiveness of the combination of link-based information with the results of the content-based classifiers depends on the importance given to each source of evidence. In the collections used, links were a more reliable source of information and, thus, to achieve classification improvements, more weight should be given to the link-based evidence;

3. In Web pages that do have links, in-links are much more common than out-links. For this reason, similarity measures should make use of in-link information. Measures like bibliographic coupling, which do not use in-links, are likely to perform poorly in Web collections.

4. Of the tested similarity measures, co-citation and Companion showed the best performance. The Companion algorithm seems to be more robust, since it uses hub/authority information to eliminate spurious similarity information.

Although these experiments were performed on the Cadê collection, we expect them to be applicable to other Web directories. The most popular pages in the Web should be those with a high number of in-links [7, 49]. These pages will also be the most interesting for Web directories, where it is preferable (and easier) to populate the hierarchy with a reasonable set of highly referenced sites, instead of a huge set of obscure pages. Thus, similarity measures that make use of in-link information are expected to be the most appropriate for building Web directories. This conclusion is reinforced by that fact that, although most Web pages have very few links (or no links at all), those that are highly linked have much more in-links than out-links.

In both the ranking and classification problems, the proposed Bayesian network models were effective in combining the different sources of evidence.

However, due to fundamental differences in the type of problem being solved, their performance was distinct. The main problems in Web ranking are due to a lack of information in the user queries [74,75]. Short and imprecise queries provide little information to a ranking algorithm based simply on text. Links, on the other hand, are widely available and a rich source of information. Further, link structure provides a source of evidence rather independent from the text of the documents. This independence makes text and links very complementary sources of information. Documents retrieved using content-based algorithms are not necessarily retrieved by link-based algorithms, and vice-versa. It can, therefore, be expected that a good combination strategy will profit from using both links and text.

In Web classification, noisy documents, multiple authorship, among other reasons, cause the poor retrieval performance observed in the application of traditional content-based classifiers. Links, on the other hand, are much more useful for classifying Web documents, as shown by the experiments in Section 5. However, it can be observed that many of the documents correctly classified using content information are also correctly classified using link information. This justifies the smaller gains in precision achieved by the proposed Bayesian network model. Sources of evidence are less independent and, thus, have less to gain from each other.

To conclude, the models used and the experiments performed allow us to provide answers, at least partially, to the research questions that motivated this work, as follows:

- Link structure is indeed an effective and valuable source of evidence for solving IR problems on the Web, in particular for document ranking and classification;

- For document ranking, the two most popular link analysis algorithms,

HITS and PageRank, were able to extract useful information from the Web link structure and showed a similar performance;

- For document classification, measures that make use of in-link information are the most appropriate. The co-citation and Companion similarity measures showed the best performance;
- The effectiveness of using link information alone depends on the problem being solved. Link information, in general, yields slightly worst results than the vector space model for document ranking. For document classification, links show results superior to those achieved by traditional content-based classifiers;
- Using the proposed Bayesian network models, links can be effectively combined with textual information, to improve the results of Web document ranking and classification;

Further, the proposed models allow the study and comparison of different algorithms under the same conditions, using a flexible and formally sound framework.

6.2 Future Work

This section shows a list of suggestions for the continuation of the work here presented. The list addresses open questions left by the research findings and new ideas found during the course of the work.

Model refinement

The models presented in Chapters 4 and 5, although already showing good performance as they are, still leave some open room for refinements. The

weighted combination of evidences that was used in classification can be also used to improve the effectiveness and personalize the results of the document ranking model. Also, different types of evidence, like the anchor text or text passages in the Web pages can be used. Finally, different evidence combination functions can be integrated into the model.

Experiments with other collections

All experiments were performed using a sample of the Brazilian Web. However, other hypertext collections may also beneficiate from the use of link-based evidence. Recently, experiments were initiated with the ACM digital library, a collection of scientific papers on computer science, where documents are linked through citations. Preliminary results indicate that citations behave differently from links. Further experimentation is needed to test the effectiveness of the proposed models on such collection, and determine if their use can also result in improvements to text-based algorithms.

Automatically learning combination parameters

Finding the ideal weights for each of the evidences to be combined is an important problem. Although we studied the effects of such weights on the results of the combination for document classification, a different evaluation of when an how much one evidence should be favored over the other must be performed for different reference collections. Methods to automatically determine such weights should, therefore, be investigated. Experiments are already under way to automatically learn combination functions by using genetic programming techniques.

Link selection techniques

Many links provide no valid information. They can exist simply for navigation purposes, Web page authors can create links arbitrarily, with no intention of linking to related or similar pages, and links can also be created artificially to increase the probability a Web page being well ranked on a commercial search engine. For these reasons, the selection of the links in the collection that should be used on link-based solutions is an essential step. In this work, only simple heuristics were considered. However, automatic techniques should be studied that can extract from the collection the links with the best informational content.

Creation of new models

Finally, by further exploring Web IR tasks, we expect new problems to arise that may lead to the creation of new Bayesian network models. Any new models created will, of course, follow the same line of work as those here presented, providing flexible, intuitive, and formally sound solutions to Web IR problems. These problems may include finding related pages, filtering information, or finding site homepages, besides the already seen problems of document ranking and classification.

Bibliography

[1] Silvia Acid, Luis M. de Campos, Juan M. Fernández-Luna, and Juan F. Huete. An information retrieval model based on simple bayesian networks. *International Journal of Intelligent Systems*, 18(2):251–265, January 2003.

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

[3] Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval*. Addison-Wesley-Longman, 1999.

[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] David C. Blair and M. E. Maron. An evaluation of retrieval effectiveness for a full-text document-retrieval system. *Communications of the ACM*, 28(3), March 1985.

[7] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the 7th International World Wide Web Conference*, pages 107–117, April 1998.

[8] Pável Calado. Consultas aproximadas em bancos de dados relacionais. Master's thesis, Federal University of Minas Gerais, Department of Computer Science, 2000.

[9] Pável Calado, Altigran Soares da Silva, Rodrigo C. Vieira, Alberto H. F. Laender, and Berthier A. Ribeiro-Neto. Searching web databases by structuring keyword-based queries. In *Proceedings of the 11th International Conference on Information and Knowledge Management CIKM 2002*, pages 346–357, McLean, VA, USA, November 2002.

[10] Pável Calado and Berthier Ribeiro-Neto. An information retrieval approach for approximate queries. *IEEE Transactions on Knowledge and Data Engineering*, 15(1):236–239, January/February 2003.

[11] Pável Calado, Berthier Ribeiro-Neto, Nivio Ziviani, Edleno Moura, and Ilmério Silva. Local versus global link information in the Web. *ACM Transactions On Information Systems*, 21(1):42–63, January 2003.

[12] James P. Callan. Document filtering with inference networks. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 262–269, Zurich, Switzerland, August 1996.

[13] Soumen Chakrabarti, Byron Dom, and Piotr Indyk. Enhanced hypertext categorization using hyperlinks. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 307–318, Seattle, WA, USA, June 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] Tatiana Coelho, Pável Calado, Lamarque Souza, and Berthier Ribeiro-Neto. Image retrieval using multiple evidence ranking. *IEEE Transactions on Knowledge and Data Engineering*, 16(4):408–417, April 2003.

[18] David Cohn and Huan Chang. Learning to probabilistically identify authoritative documents. In *Proceedings of the 17th International Conference on Machine Learning, ICML-00*, pages 167–174, Stanford, CA, USA, June 2000.

[19] 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.

[20] James H. Coombs. Hypertext, full text, and automatic linking. In *Proceedings of the 13th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 83–98, Brussels, Belgium, September 1990.

[21] W. Bruce Croft, T. J. Lucia, J. Cringean, and P. Willett. Retrieving documents by plausible inference: an experimental study. *Information Processing & Management*, 25(6):599–614, 1989.

[22] W. Bruce Croft and Howard Turtle. A retrieval model for incorporating hypertext links. In *Proceedings of the 2nd Annual ACM Conference on Hypertext*, pages 213–224, Pittsburgh, PA, USA, November 1989.

[23] Luis M. de Campos, Juan M. Fernández-Luna, and Juan F. Huete. Query expansion in information retrieval systems using a bayesian network-based thesaurus. In *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI–98)*, pages 53–60, San Francisco, CA, July 1998.

[24] Maria de Lourdes da Silveira, Berthier Ribeiro-Neto, Rodrigo de Freitas Vale, and Rodrigo Torres Assumpção. Vertical searching in juridical digital libraries. In *Advances in Information Retrieval, 25th European Conference on IR Research, ECIR 2003, Proceedings*, pages 491–501, Pisa, Italy, April 2003.

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

Also in Proceedings of the 8th International World Wide Web Conference.

[26] Susan T. Dumais and Rong Jin. Probabilistic combination of content and links. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 402–403, New Orleans, LA, USA, September 2001.

[27] Susan T. Dumais, John Platt, David Heckerman, and Mehran Sahami. Inductive learning algorithms and representations for text categorization. In *Proceedings of the 7th International Conference on Information and Knowledge Management CIKM 98*, pages 148–155, Bethesda, MD, USA, November 1998.

[28] 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, Proceedings*, pages 41–56, Pisa, Italy, April 2003.

[29] E. Fix and J. L. Hodges. Discriminatory analysis — nonparametric discrimination: Consistency properties. Technical Report Tec. Rep. 4, Proj. N. 21-49-004, USAF School of Aviation Medicine, Randolph Field, TX, USA, 1951.

[30] Mark E. Frisse and Steve B. Cousins. Information retrieval from hypertext: Update on the dynamic medical handbook project. In *Proceedings of the 2nd Annual ACM Conference on Hypertext*, pages 199–212, Pittsburgh, PA, USA, November 1989.

[31] Johannes Furnkranz. Exploiting structural information for text classification on the WWW. In *Proceedings of the 3rd Symposium on Intelligent*

Data Analysis (IDA99), pages 487–498, Amsterdam, The Netherlands, August 1999.

[32] Jianfeng Gao, Guihong Cao, Hongzhao He, Min Zhang, Jian-Yun Nie, Stephen Walker, and Stephen Robertson. TREC-10 web track experiments at MSRA. In *The Tenth Text REtrieval Conference (TREC-2001)*, pages 384–392, Gaithersburg, MD, USA, November 2001.

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

[34] 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.

[35] C. Lee Giles, Kurt D. Bollacker, and Steve Lawrence. CiteSeer: An automatic citation indexing system. In *Proceedings of the 3rd ACM International Conference on Digital Libraries*, pages 89–98, Pittsburgh, PA, USA, June 1998.

[36] Eric J. Glover, Kostas Tsioutsiouliklis, Steve Lawrence, David M. Pennock, and Gary W. Flake. Using Web structure for classifying and describing Web pages. In *Proceedings of the 11th International World Wide Web Conference*, Honolulu, HI, USA, May 2002.

[37] Norbert Gövert, Mounia Lalmas, and Norbert Fuhr. A probabilistic description-oriented approach for categorizing web documents. In *Proceedings of the 8th International Conference on Information and Knowl-*

edge Management CIKM 99, pages 475–482, Kansas City, MO, USA, November 1999.

[38] David Haines and W. Bruce Croft. Relevance feedback and inference networks. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2–11, Pittsburgh, PA, USA, June 1993.

[39] David Hawking and Nick Craswell. Overview of TREC-2001 Web track. In *The Tenth Text REtrieval Conference (TREC-2001)*, pages 61–67, Gaithersburg, MD, USA, November 2001.

[40] David Hawking, Nick Craswell, and Paul B. Thistlewaite. Overview of TREC-7 very large collection track. In *The Seventh Text REtrieval Conference (TREC-7)*, pages 91–104, Gaithersburg, MD, USA, November 1998.

[41] David Hawking, Nick Craswell, Paul B. Thistlewaite, and Donna Harman. Results and challenges in web search evaluation. *Computer Networks*, 31(11–16):1321–1330, May 1999. Also in Proceedings of the 8th International World Wide Web Conference.

[42] 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.

[43] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, 13(2):415–425, March 2002.

[44] 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.

[45] Thorsten Joachims, Nello Cristianini, and John Shawe-Taylor. Composite kernels for hypertext categorisation. In *Proceedings of the 18th International Conference on Machine Learning, ICML-01*, pages 250–257, Williams College, US, June 2001.

[46] Tapas Kanungo and Jason Y. Zien. Integrating link structure and content information for ranking web documents. In *The Tenth Text REtrieval Conference (TREC-2001)*, pages 237–239, Gaithersburg, MD, USA, November 2001.

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

[48] Jon 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.

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

[50] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence*, pages 1137–1143, Montreal, Canada, August 1995.

[51] S. Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, D. Sivakumar, Andrew Tomkins, and Eli Upfal. The Web as a graph. In *Proceedings of the 19th Symposium on Principles of Database Systems*, pages 1–10, Dallas, TX, USA, May 2000.

[52] Ronny Lempel and Shlomo Moran. The stochastic approach for link-structure analysis (SALSA) and the TKC effect. *Computer Networks*, 33(1–6):387–401, June 2000. Also in Proceedings of the 9th International World Wide Web Conference.

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

[54] David D. Lewis. Evaluating and optimizing autonomous text classification systems. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 246–254, Seattle, WA, USA, July 1995.

[55] Andrew K. McCallum. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. Available at `http://www.cs.cmu.edu/~mccallum/bow`, 1996.

[56] Andrew K. McCallum and Kamal Nigam. A comparison of event models for Naive Bayes text classification. In *Proceedings of AAAI/ICML-98 Workshop on Learning for Text Categorization*, pages 41–48, Madison, WI, USA, July 1998.

[57] Tom Mitchell. *Machine Learning.* McGraw-Hill, March 1997.

[58] 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 international ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 264–271, Athens, Greece, July 2000.

[59] 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.

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

[61] 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.

[62] 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.

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

[64] Gerard Salton, J. Allan, and Chris Buckley. Approaches to passage retrieval in full text information systems. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 49–58, Pittsburgh, PA, USA, June 1993.

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

[66] Gerard Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

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

[68] Dongwook Shin, Sejin Nam, and Munseok Kim. Hypertext construction using statistical and semantic similarity. In *Proceedings of the 2nd ACM International Conference on Digital Libraries*, pages 57–63, Philadelphia, PA, USA, July 1997.

[69] Altigran Silva, Eveline Veloso, Paulo Golgher, Berthier Ribeiro-Neto, Alberto Laender, and Nivio Ziviani. CobWeb - a crawler for the brazilian web. In *Proceedings of the String Processing and Information Retrieval Symposium (SPIRE'99)*, pages 184–191, Cancun, Mexico, September 1999.

[70] 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.

[71] Seán Slattery and Tom Mitchell. Discovering test set regularities in relational domains. In *Proceedings of the 17th International Conference on Machine Learning*, pages 895–902, Stanford, CA, USA, June 2000.

[72] 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.

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

[74] Amanda Spink, Bernard J. Jansen, Dietmar Wolfram, and Tefko Saracevic. From e-sex to e-commerce: Web search changes. *IEEE Computer*, 35(3):107–109, April 2002.

[75] Amanda Spink, Dietmar Wolfram, Bernard J. Jansen, and Tefko Saracevic. Searching the web: The public and their queries. *Journal of the American Society for Information Science and Technology*, 52(3):226–234, February 2001.

[76] M. Stone. Cross-validation choices and assessment of statistical predictions. *Journal of the Royal Statistical Society*, B36:111–147, 1974.

[77] 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, VA, USA, November 2002.

[78] 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.

[79] Mike Thelwall and David Wilkinson. Finding similar academic Web sites with links, bibliometric couplings and colinks. *Information Processing & Management*, 2003. (in press).

[80] Howard Turtle. *Inference Networksd for Document Retrieval.* PhD thesis, Graduate School of the University of Massachusetts, February 1991.

[81] 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.

[82] Howard Turtle and W. Bruce Croft. Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems*, 9(3):187–222, July 1991.

[83] V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Velag, March 1995.

[84] Thijs Westerveld, Wessel Kraaij, and Djoerd Hiemstra. Retrieving Web pages using content, links, URLs and anchors. In *The Tenth Text REtrieval Conference (TREC-2001)*, pages 663–672, Gaithersburg, MD, USA, November 2001.

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

[86] Wensi Xi, Edward A. Fox, Roy P. Tan, and Jiang Shu. Machine learning approach for homepage finding task. In *Proceedings of the String Processing and Information Retrieval Symposium (SPIRE'2002)*, pages 145–159, Lisbon, Portugal, September 2002.

[87] Jinxi Xu and W. Bruce Croft. Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 4–11, Zurich, Switzerland, August 1996.

[88] Jinxi Xu and W. Bruce Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM Transactions on Information Systems*, 18(1):79–112, January 2000.

[89] 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.

[90] 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.

[91] 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.