UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação
ISSN 0103-2569

Métricas de Qualidade de Hiperdocumentos: uma análise utilizando Sistemas de Aprendizado de Máquina

Elisandra Aparecida Alves da Silva
Renata Pontin de Mattos Fortes
Maria Carolina Monard

Nº 135

RELATÓRIOS TÉCNICOS

São Carlos – SP
Mar./2001

Métricas de qualidade de hiperdocumentos: uma análise utilizando sistemas de aprendizado de máquina

RELATÓRIO TÉCNICO

Autores:

Elisandra Aparecida Alves da Silva

(e-mail: elisilva@icmc.sc.usp.br)

Profa. Dra. Renata Pontin de Mattos Fortes

(e-mail: renata@icmc.sc.usp.br)

Profa. Dra. Maria Carolina Monard

(e-mail: mcmonard@icmc.sc.usp.br)

Departamento de Computação
Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo – Campus de São Carlos
Caixa Postal 668
13560-970 São Carlos, SP

Índice

Índice de Figuras

Índice de Tabelas

1. INTRODUÇÃO

O avanço da WWW proporcionou um aumento significativo em desenvolvimento de hiperdocumentos e sistemas hipermídia. Entretanto, devido a natureza dinâmica das alterações das informações e dependendo da quantidade de informação a ser processada, o projeto e a manutenção hipermídia podem ser atividades complexas que, se não abordadas sistematicamente e cuidadosamente, podem causar muitos problemas.

Este trabalho enfoca características de um hiperdocumento as quais podem ser consideradas para auxiliar sua qualidade total. A idéia geral é se concentrar em características que possam ser medidas objetivamente em função de dados que possam ser automaticamente coletados e subseqüentemente utilizados para fornecer um diagnóstico sobre aspectos relacionados à qualidade de hiperdocumentos. Obviamente, as métricas propostas não avaliam a qualidade total do hiperdocumento, uma vez que a qualidade é fortemente dependente da organização lógica e da qualidade semântica da informação presente no hiperdocumento. No entanto, as métricas podem ser usadas como um critério para diagnosticar ou para minimizar alguns problemas (Fortes & Nicoletti 99).

De acordo com (Fortes & Nicoletti 97), o projeto de *Web* sites não é um processo exato, e muitos aspectos relacionados com produção de um "bom" hiperdocumento deveriam ser considerados. Visto que o projeto hipermídia possibilita maneiras alternativas de se organizar informações, não existem regras rígidas que possam ser seguidas para executar um bom projeto. A inexistência de alguma regra que pudesse ser considerada para se avaliar o hiperdocumento, durante sua criação, torna a tarefa de autoria uma atividade difícil. Por sua vez, a dificuldade durante a autoria pode ser uma das razões para a existência de tantos hiperdocumentos de pouca qualidade. Algumas propriedades de hiperdocumentos que podem ser usadas para identificar estruturas de textos mal projetadas têm sido apontadas em (Thistlewaite 95): *links* inconsistentes, inconsistência nos critérios para criação de *link*, e dificuldade de manutenção.

Nesse contexto, investigamos as propriedades de *links* de forma que eles possam ser classificados e esta classificação, por sua vez, forneça um forneça *feedback* sobre a estrutura hipertexto, durante os processos de autoria e manutenção de hiperdocumentos.

Neste trabalho são apresentadas diversas métricas, propostas na literatura e outras por nós sugeridas, para analisar a qualidade da estrutura de hiperdocumentos. Essas métricas são analisadas e avaliadas experimentalmente com o suporte de sistemas de aprendizado de máquina supervisionado e não supervisionado.

O trabalho está organizado da seguinte forma, na Seção 2 são descritas as métricas utilizadas bem como a forma geral de implementação dos programas que coletam as informações necessárias para obtenção dos valores dessas métricas. Na Seção 3 é descrito o experimento realizado e os resultados obtidos. Finalmente, são apresentadas as referências bibliográficas.

2. MÉTRICAS DE QUALIDADE DE HIPERDOCUMENTOS

2.1. Considerações Iniciais

Autoria hipermídia tem sido uma área de pesquisa de considerável interesse nos últimos anos. Existem muitas abordagens para efetuar a autoria e, consequentemente, muitas alternativas são oferecidas aos autores hipermídia. Na literatura, diferentes modelos de autoria hipermídia são propostos (Garzotto et al. 91), (Rossi et al. 95), metodologias (Balasubramanian et al. 94), ambientes orientados a modelos (Türing et al. 95), (Nanard & Nanard 95), (Jordan et al. 89), (Andrews et al. 95a), (Marshall et al. 91), (Marshall et al. 95), (Marmann et al. 92), (Duval & Olivié 95), (Catlin & Garrett 91), e ambientes de propósitos gerais (Davis et al. 92), (Meyrowitz 86), (Bernstein et al. 91), (Goldberg et al. 96), (Thimbleby 96), (Andrews et al. 95b).

O processo de autoria hipermídia visa a construção de vários produtos, tais como, uma especificação, um projeto e uma aplicação hipermídia. Entender o processo contribui para controlá-lo e melhorá-lo, e um modo de fazer isso é utilizando métricas.

De acordo com (Basili et al. 94), métricas podem ser usadas para:

i. Suportar o planejamento de projeto
ii. Determinar os pontos fortes e fracos dos processos e produtos atuais
iii. Fornecer princípios para técnicas de adoção/refinamento
iv. Avaliar a qualidade de processos e produtos específicos
v. Avaliar o progresso de um projeto durante seu desenvolvimento
vi. Tomar ações corretivas baseadas na avaliação
vii. Avaliar o impacto da tomada de tal ação

De acordo com (Hatzimanikatis et al. 95), métricas podem ser utilizadas em engenharia hipertexto para os seguintes propósitos:

i. Predizer e planejar as próximas fases de um projeto de autoria hipertexto, especialmente teste e manutenção.
ii. Identificar, durante a fase de autoria, as partes de um hiperdocumento que são muito complexas ou mal estruturadas.
iii. Servir como componente de um modelo de qualidade

Embora muitas pesquisas voltadas ao desenvolvimento de métricas para hipermídia tenham sido realizadas por diversos pesquisadores (Botafogo et al. 92), (Rivlin et al. 94), (Garzotto et al. 94), (Garzotto et al. 95), (Hatzimanikatis et al. 95), (Yamada et al. 95), eles desenvolveram seus trabalhos de uma forma *ad-hoc*, propondo algumas métricas de forma ambígua e limitando a aplicação. Existem decisões que têm que ser tomadas quando da definição de uma métrica. Essas decisões têm que ser tomadas com relação ao objetivo da métrica e definindo um modelo empírico baseado em alguma hipótese (Mendes et al. 98). Infelizmente, muitas métricas propostas na literatura não discutem qual a motivação dessas decisões, tornando difícil o entendimento das considerações feitas.

A Tabela 1 de (Mendes et al. 98) compara métricas propostas considerando as quatro questões que deveriam ser respondidas quando da validação de uma métrica, segundo (Briand et al. 97):

(1) A métrica captura adequadamente o atributo que se propõe a medir?

(2) O atributo é bem definido baseado num modelo empírico?

(3) Existe alguma evidência empírica suportando a hipótese do modelo empírico?

(4) A métrica é útil de uma perspectiva prática?

Tabela 1. Comparação das métricas propostas (Mendes et al. 98)

Questão	Métricas propostas			
	(Botafogo et al. 92)	(Garzotto et al. 94)	(Hatzimanikatis et al. 95)	(Yamada et al. 95)
(1)	Sim	Não	Não	Sim
(2)	Não	Não	Não	Sim
(3)	Não	Não	Não	Sim
(4)	Sim	Sim	Sim	Sim

Todos os autores definiram métricas que são úteis de uma perspectiva prática. Embora apenas nos trabalhos de Botafogo et al. e Yamada et al. a métrica proposta captura adequadamente o atributo que se propõe medir. O único trabalho no qual existe uma evidência empírica suportando a hipótese do modelo empírico é o de Yamada et al.

As primeiras métricas de hiperdocumentos foram propostas por (Botafogo & Shneiderman 92), e posteriormente usadas por (Rivlin et al. 94). Elas se baseiam prioritariamente na aplicação de técnicas baseadas em grafos para obtenção das métricas relativas a Compactação e Estratificação. Compactação indica a conectividade intrínseca do hiperdocumento. Do ponto de vista do leitor, um alto valor de compactação indica que cada nó possui muitos *links*. Estratificação revela o grau de organização de um hiperdocumento; o máximo valor de estratificação ocorre quando o hiperdocumento é linear.

Os fatores de qualidade importantes, identificados como legibilidade e manutenibilidade de um hiperdocumento, bem como os critérios correspondentes foram definidos por (Hatzimanikatis et al. 95). Entretanto, esses dois fatores não podem ser obtidos diretamente. Por isso, esses fatores são decompostos em critérios de nível mais baixo. A maioria dos critérios afeta ambos os fatores, a saber: **tamanho, complexidade do caminho, impureza da árvore, modularidade, complexidade individual do nó, coerência, complexidade do conteúdo dos nós, simplicidade.**

Um outro método de avaliação, conhecido como "orientado a projeto", proposto por (Garzotto et al. 95), considera o seguinte subconjunto de critérios de avaliação: **riqueza, facilidade, auto-evidência, previsibilidade, legibilidade, consistência e reuso**.

Para cada um dos critérios de (Hatzimanikatis et al 95) e (Garzotto et al. 95), apresentados, por sua vez podem ser atribuídas métricas que auxiliem à sua avaliação. De fato, a qualidade pode ser vista como um parâmetro (ou um indicativo) que pode ser estimado por meio de métricas (de um atributo de interesse) (Jalote 97). No entanto, as métricas não são fáceis de se estabelecer, uma vez que são variáveis dependentes de características intrínsecas ao software produto e seu processo.

Na próxima seção são apresentadas as métricas utilizadas e implementadas para a realização do experimento.

2.2. As métricas utilizadas no experimento

As métricas apresentadas nessa seção foram escolhidas considerando-se quais métricas poderiam ser obtidas a partir das informações fornecidas pelo sistema LiOS (*Link-Oriented System*) (Fortes 96). Esse sistema tem como objetivo principal verificar indicativos de qualidade de hiperdocumentos da *Web*, quanto aos critérios de reuso e consistência de *links*. Recentemente, uma nova versão desse sistema foi construída, contando com o suporte de um SGBD Relacional para controlar as versões das informações coletadas dos sites. A nova versão do sistema foi denominada DB-LiOS (*DataBase - Link Oriented System*) (Seraphim & Fortes 2000).

A ferramenta DB-LiOS fornece a base de dados necessária para que as métricas possam ser coletadas e armazenadas. As métricas utilizadas neste trabalho podem ser divididas em duas categorias:

1) Métricas aplicadas a sites.
2) Métricas aplicadas a páginas.

Na primeira categoria as métricas podem ser computadas quando é conhecida a estrutura de todo o hiperdocumento. As métricas da segunda categoria podem ser computadas quando é conhecida a estrutura da página. Nas próximas seções são apresentadas as métricas consideradas neste trabalho.

2.2.1. Métricas aplicadas a sites

Essas métricas são: Compactação, Estratificação, Impureza da Árvore, Nro. de páginas que voltam à página anterior e Nro. de páginas que voltam à homepage. A seguir definimos alguns conceitos relacionados às métricas utilizadas.

2.2.1.1. Compactação

Definição 1. Seja H um hiperdocumento representado pelo grafo G=<N,L>, onde N é o conjunto de nós (|N|=n) e L o conjunto de *links*. A matriz distância de H é a matriz A_{nxn} onde:

$a_{ij}=\infty$ (se o nó i não pode ser alcançado pelo nó j) ou

a_{ij}=min (onde min é o mínimo caminho de i a j)

A **centralidade** de um nó é a soma das distâncias desse nó a cada nó do grafo, dividido pela soma de todas as distâncias na matriz distância.

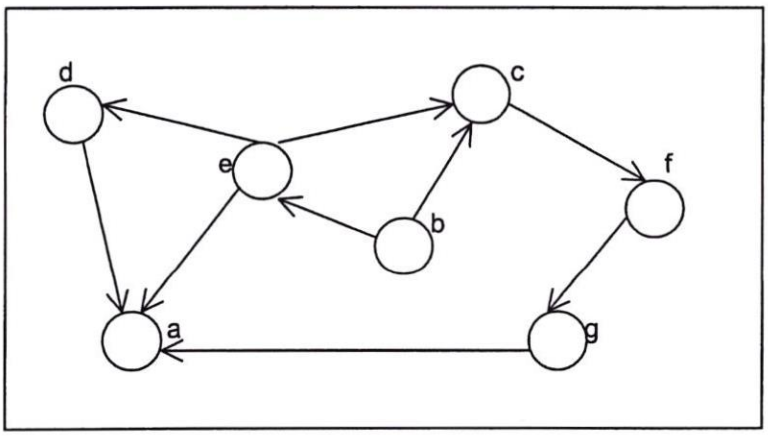


Figura 1. Um grafo representando um hiperdocumento

Na prática, os valores infinitos são substituídos por um número K (constante de conversão), resultando na **matriz distância convertida**. É comum escolher K como o número de nós do

hiperdocumento. Por exemplo, a matriz distância, a matriz convertida e a centralidade de cada nó (c_n) do hiperdocumento representado pelo grafo apresentado na Figura 1 são apresentados na Tabela 2. É possível notar que o nó b é o "mais central", tendo em vista que b apresenta o menor valor de c_n – cada nó do grafo pode ser acessado a partir de b.

Tabela 2. Matriz distância e matriz convertida com o valor somado para a centralidade (c_n) de cada nó (a soma de todas as distâncias na matriz distância convertida é 198)

	a	*b*	*c*	*d*	*e*	*f*	*g*
a	0	∞	∞	∞	∞	∞	∞
b	2	0	1	2	2	2	3
c	3	∞	0	∞	1	1	2
d	1	∞	∞	0	∞	∞	∞
e	1	∞	1	1	0	2	3
f	2	∞	∞	∞	∞	0	1
g	1	∞	∞	∞	∞	∞	0

	a	*b*	*c*	*d*	*e*	*f*	*g*	c_n
a	0	7	7	7	7	7	7	0.212
b	2	0	1	2	2	2	3	0.056
c	3	7	0	7	1	1	2	0.136
d	1	7	7	0	7	7	7	0.182
e	1	7	1	1	0	2	3	0.076
f	2	7	7	7	7	0	1	0.157
g	1	7	7	7	7	7	0	0.182

Definição 2. Seja H um hiperdocumento representado pelo grafo G=<N,L>, onde N é o conjunto de nós (|N|=n), L o conjunto de *links* e sua matriz distância convertida é A'_{nxn}. A **compactação** de H é definida por:

$$\text{Compactação} = (\text{Max} - \text{Soma})/(\text{Max} - \text{Min})$$

onde:

$$\text{Soma} = \sum_{i=1}^{n}\sum_{j=1}^{n} a'_{ij}$$

$$Max=(n^2 - n)*K$$
$$Min=(n^2 - n)$$

a'_{ij}=K (se o nó i não pode ser alcançado pelo nó j) ou

a'_{ij}=min (onde min é o mínimo caminho de i a j)

Quando todo nó em um hiperdocumento é isolado, todo elemento a'_{ij} na matriz distância convertida é igual a K (o máximo valor que um elemento pode assumir na matriz distância convertida) para $i \neq j$. Isso significa que a matriz terá $(n^2 - n)$ elementos iguais a K, consequentemente $Max = (n^2 - n)*K$. O valor mínimo de Soma será atingido quando todo nó é conectado aos outros. Quando isso acontece, todo elemento na matriz distância convertida é igual a 1 e, consequentemente, Min será igual a $(n^2 - n)$. No exemplo anterior,

$$\begin{aligned} \text{Compactação} &= [((7^2 - 7)*7) - 198]/[(7^2 - 7)*7 - (7^2 - 7)] \\ &= (294 - 198)/(294 - 42) \\ &= 96/252 \\ &= 0.381. \end{aligned}$$

Se cada nó do grafo é diretamente conectado a todos os outros, a compactação será 1. Por outro lado, se um grafo possui apenas nós isolados terá compactação igual a 0. Botafogo considera que uma compactação de 0.5 sugere uma estrutura bem conectada mas, obviamente, isso depende do tipo de hiperdocumento que está sendo medido.

2.2.1.2. Estratificação

A estratificação é uma métrica usada para capturar a ordenação linear do hiperdocumento. Essa métrica reflete as escolhas de navegação do usuário enquanto navega no hiperdocumento. Dependendo do modo como o hiperdocumento é construído, os usuários terão acesso a uma estrutura mais ou menos flexível de navegação. Um hiperdocumento estratificado não permite muita flexibilidade durante a navegação, ou seja, fornece um modo estratificado ou hierárquico de navegação. Um hiperdocumento com uma alta estratificação indica que os usuários não tem muita escolha e, consequentemente, têm menor possibilidade de sofrer com o problema de desorientação. Por outro lado, uma baixa estratificação sugere

que o número excessivo de *links* que o usuário possa escolher pode causar desorientação ao usuário (Fortes & Nicoletti 99).

Definição 3. (DeBra 99) Seja H um hiperdocumento representado pelo grafo G=<N, L>, onde N é o conjunto de nós (|N|=n), L é o conjunto de *links*, e seja d(u,v) a distância mínima do nó u para o nó v. Então

A_i é a soma das distâncias finitas d(i,v) para todo v em D. Ou seja, A_i é a soma das entradas finitas nas i-ésimas linhas da matriz distância para D. A_i é chamado "status" do nó i.

B_i é a soma das distâncias finitas d(u,v) para todo u em D. Ou seja, B_i é a soma das entradas finitas nas i-ésimas colunas da matriz distância para D. B_i é chamado "contrastatus" do nó i.

A distância total de D é a soma de todas as distâncias finitas em D.

O prestígio do nó i é dado por $A_i - B_i$. O prestígio total (soma de todos os prestígios) de um hiperdocumento é sempre 0. O prestígio absoluto (ou estratificação absoluta) de D é definido como a soma dos valores absolutos do prestígio de cada nó; seu valor obviamente aumenta com o tamanho do hiperdocumento. A Tabela 3 apresenta esses valores para o grafo da Figura 1. Um modo de normalizar o prestígio absoluto é compará–lo ao prestígio de um documento linear do mesmo tamanho. O prestígio absoluto linear (*Linear Absolute Prestige* - LAP) de um hiperdocumento com n nós é o prestígio absoluto de um hiperdocumento linear com n nós, ou seja, se n é o número de nós, LAP é definido como (Papast 97):

$$LAP = \begin{cases} n^3/4 \text{ se n div } 2=0 \\ (n^3 - n)/4 \text{ se n div } 2 = 1 \end{cases}$$

A estratificação de um hiperdocumento é definida como seu prestígio absoluto dividido por seu LAP.

O prestígio absoluto de D é 110. Seu LAP é $(7^3 - 7)/4 = 84$. Para o exemplo anterior, a estratificação é 110/84 = 1.31. Infelizmente, até agora não se sabe como interpretar os valores obtidos com essas métricas.

Tabela 3. Prestígio e prestígio absoluto para o grafo da Figura 1

	A_i	B_i	$Prestígio_i$	$\|Prestígio_i\|$
a	42	10	32	32
b	11	42	-31	31
c	27	30	-3	3
d	36	31	5	5
e	15	36	-21	21
f	31	26	5	5
g	36	23	13	13

2.2.1.3. Impureza da árvore

Algoritmos para identificação de raízes da árvore e hierarquia foram propostos por (Botafogo & Shneiderman 92).

Em Engenharia de Software, acredita-se que quanto maior o desvio da estrutura de um projeto em relação à estrutura de árvore pura, pior é o projeto (Kan 95). (Fenton 91) apresenta diversas propriedades que uma métrica de impureza de árvore MI(G), para um grafo G, deveria satisfazer. Uma métrica satisfatória é dada pelo número de arestas que excedem a extensão da árvore do grafo G dividido pelo número máximo de arestas que excedem a extensão da árvore.

Sendo **e** o número de arestas (*links*) no grafo G e **n** o número de nós[1] (páginas), então a **medida de impureza (MI)** do grafo G, representando o hiperdocumento, em relação a uma árvore é definida pela seguinte expressão:

$$\mathbf{MI(G)} = \frac{2(e - n + 1)}{(n-1)(n-2)}$$

[1] Consideramos neste trabalho nós como páginas e páginas como *urls*. Embora existam trabalhos que referenciem os termos nó, página e *url* com definições diferentes, para efeito deste trabalho os consideramos de mesmo significado.

De fato, o grafo é comparado a um grafo completo com o mesmo número de nós. Essa métrica caracteriza precisamente a impureza da árvore do projeto de um sistema, bem como do projeto de um hiperdocumento.

Impureza da árvore significa quanto o grafo que representa o hiperdocumento se desvia de uma árvore pura. A estrutura de uma árvore pura lembra a estrutura de documentos impressos, os quais podem ser facilmente entendido por seus leitores e mantenedores.

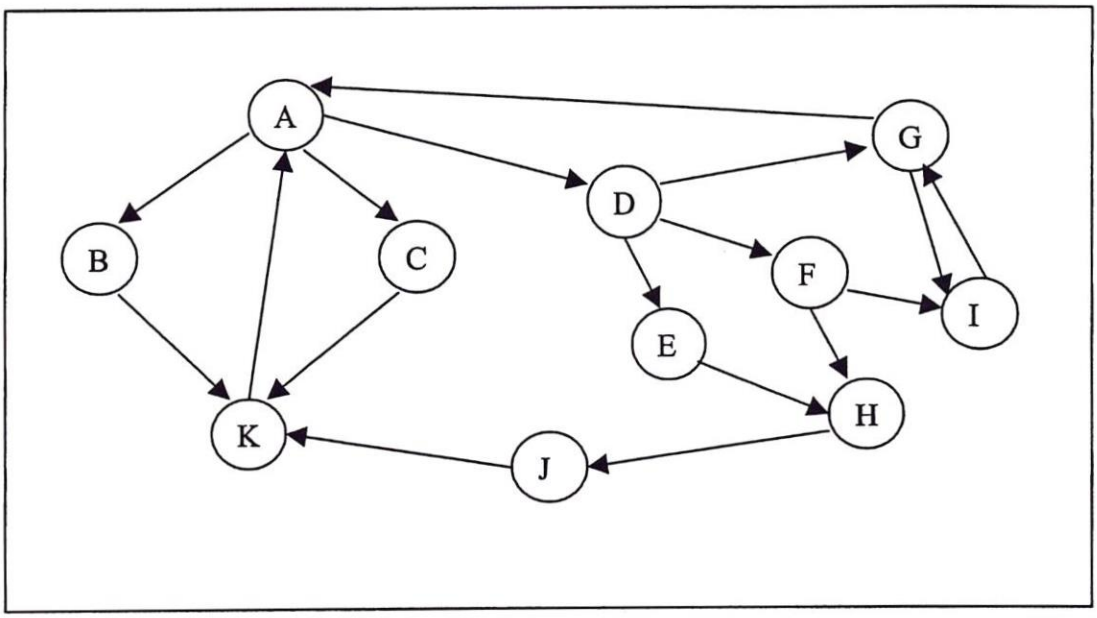


Figura 2. Grafo representando um hiperdocumento para o exemplo de obtenção da Impureza da Árvore

De acordo com (Hatzimanikatis et al. 95), o desvio de um grafo de um hiperdocumento da estrutura de uma árvore influencia na leitura do hiperdocumento. Navegação em hiperdocumentos estruturados como árvore sem referências cruzadas é muito mais fácil do que em hiperdocumentos com muitas referências cruzadas. A estrutura de árvore lembra livros impressos. Um grau de impureza da árvore aceitável depende da aplicação. Os autores acreditam que aplicações tais como manuais *online* e livros têm um pequeno valor para a impureza da árvore.

Por exemplo, para o grafo ilustrado na Figura 2 a Impureza da árvore é:

$$MI(G) = \frac{(2x7)}{(11-1)(11-2)} = 14/90 = 0.155$$

2.2.1.4. Nro. de páginas que voltam à página anterior

Essa métrica define o número de páginas que possuem *links* para páginas que fazem referência a ela, ou seja, se uma página A contém um *link* para uma página B, e se B também contém um *link* para A então o valor para essa métrica é incrementado.

2.2.1.5. Nro. de páginas que voltam à homepage

Essa métrica define o número de páginas que possuem *link* para a página principal do site (homepage). Para a Figura 2, considerando-se que a homepage é o nó A, o valor para essa métrica é 2 pois apenas os nós G e K apontam para o nó A.

Na próxima seção são apresentados os algoritmos e a forma de implementação dessas cinco métricas aplicadas a sites.

2.2.2. Implementação das métricas aplicadas a sites

O sistema DB-LiOS, utilizado neste trabalho para a extração dos *links*, foi desenvolvido em linguagem Delphi. Os programas que coletam as métricas descritas anteriormente também foram desenvolvidos em Delphi, utilizando a mesma base de dados do sistema DB-LiOS.

Para implementação de todas as métricas descritas na seção anterior, cada hiperdocumento é representado utilizando a matriz de adjacência.

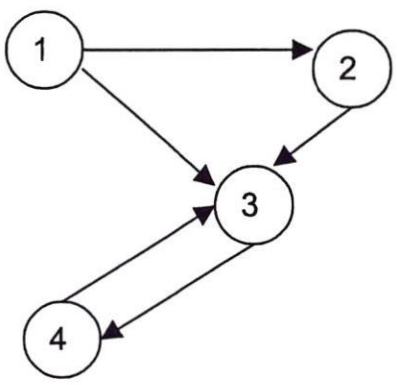
O sistema DB-LiOS nos fornece a informação necessária para obter a matriz de adjacência: para cada site tem-se uma lista com todos os *links*, sendo que esses *links* são representados por nó-fonte, âncora e nó-destino. A matriz de adjacência é gerada observando-se quais páginas são conectadas por *links*.

Os algoritmos apresentados a seguir foram utilizados para a obtenção da matriz de mínimo caminho a partir da matriz de adjacência. A matriz de mínimo caminho é necessária para os cálculos das métricas de Compactação e Estratificação.

Definição 4. (Szwarcfiter 84) Seja V o conjunto de vértices e A o conjunto de arestas, a matriz de adjacência X de um dígrafo D(V,A) é definida como:

$x_{ij} = 1$ se $(v_i, v_j) \in A$

$x_{ij} = 0$ se $(v_i, v_j) \notin A$



$$X = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \end{array}$$

Figura 3. Um dígrafo e sua matriz de adjacência

O algoritmo geral utilizado para obter o caminho de menor custo entre 2 vértices, é dado a seguir:

1. Considere um dígrafo com pesos associados às arestas
2. Substitua na matriz adjacência os valores '1' pelos respectivos pesos
3. Obtenha a matriz dos caminhos de menor custo, MC, tal que

MC_{ij}= custo do menor caminho de i a j

4. Estratégia

   Trocar : 0 por ∞

   e 1 pelo peso na matriz Adjacência X e aplicar o algoritmo a seguir:

Um exemplo de matriz de mínimo caminho é ilustrado na Figura 4 .

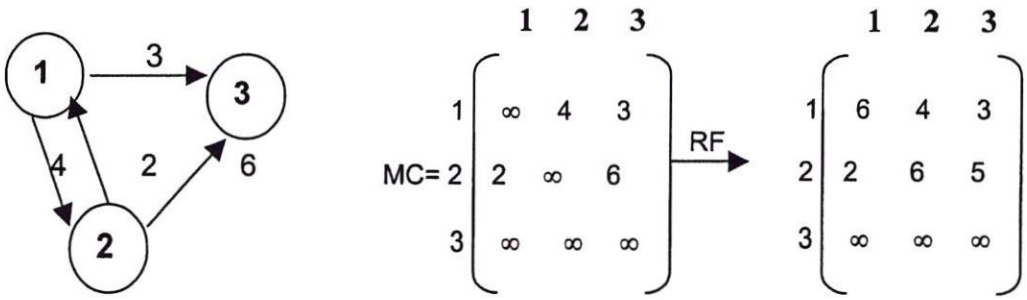


Figura 4. Um dígrafo e sua matriz de mínimo caminho

O algoritmo de Robert-Ferland (Szwarcfiter 84) que foi utilizado para obter a matriz de mínimo caminho é dado a seguir.

Algoritmo de Robert-Ferland

Dada X

Constrói MC, matriz dos caminhos de menor custo

Início

Faça MC=X

Para j=1 até n

Para i=1 até n faça

Se $MC_{ij} \neq \infty$ então

Para k=1 até n faça

$MC_{ik} = \min(MC_{ik}, MC_{ij} + MC_{jk})$

Se existe caminho de custo MC_{ij} de i a j então o custo mínimo de i a k é o mínimo entre o caminho direto de i a k e o caminho de i a j mais de j a k.

Na nossa implementação todo peso é igual a 1. Portanto, MC_{ij} será o comprimento de menor caminho de v_i a v_j.

As matrizes e algoritmos descritos anteriormente foram por nós utilizados para implementação dos programas utilizados para obtenção dos valores para as métricas de Compactação, Estratificação, Nro. de páginas que voltam à página anterior e Nro. de páginas que voltam à homepage.

Para a obtenção do valor da métrica de Impureza da Árvore, apenas os números de nós (páginas) e ligações (*links*) são utilizados, sendo o valor da métrica obtido diretamente das informações fornecidas pelo sistema DB-LiOS.

Os cálculos dos valores das métricas Nro. de páginas que voltam à página anterior (1) e Nro. de páginas que voltam à homepage (2), são realizados da seguinte forma:

(1) Para cada elemento a_{ij} da matriz de adjacência com valor igual a 1 é verificado se o elemento a_{ji} também é igual a 1. Caso seja, o valor da métrica é incrementado.

(2) Fixando na coluna 0 da matriz de adjacência, referente à homepage, verifica-se quantos elementos nessa coluna são iguais a 1. Verificamos portanto, quantas páginas possuem *links* para a homepage.

Na próxima seção são apresentadas as métricas que podem ser aplicadas a páginas.

2.2.3. Métricas aplicadas a páginas

As métricas descritas nessa seção necessitam dos dados provenientes das páginas dos sites (nós do grafo), as métricas são: Nro. de *links* que são imagens, Superfície dessas imagens, Nro. de *In Links*, Nro. de *Out Links*. Essas métricas necessitam das informações presentes apenas nos *links* das páginas, tais como: âncora, nó-destino. A ferramenta DB-LiOS, utilizada para extrair os *links* dos sites fornecidos pelo usuário, procura por *links* estáticos como mostra o exemplo a seguir:

1) Pequeno texto
2) <img src="... .gif" width="125" height="78" border="0">

Nesse exemplo as âncoras são respectivamente:

1) Pequeno texto
2) <img src="... .gif" width="125" height="78" border="0">

O uso de imagens que são âncoras pode ser conveniente porque imagens podem ser de grande ajuda para identificação imediata do propósito do *link*, mas por outro lado imagens "pesadas" podem ter um grande impacto no sistema. A seguir são apresentadas as métricas implementadas.

2.2.3.1. Nro. de links que são imagens

Essa métrica define o número de *links* que são imagens em uma página.

Conforme tem sido observado, com os recursos de multimídia mais disponíveis, a "iconização" para representar direções dos *links* ou "localização" de mais informação a ser explorada pelo usuário tem se popularizado. Isso é potencialmente bom pois facilita a leitura para o usuário, mas quando não possui uniformidade, acaba levando a inconsistência e pode desorientar o usuário.

2.2.3.2. Superfície dessas imagens

Essa métrica define a superfície total utilizada pelas imagens que são *links* em uma página.

2.2.3.3. Nro. de In *Links*

Número de *links inline* em uma página, ou seja, aqueles *links* que apontam para uma página do mesmo site do hiperdocumento. *Links inline* podem ser muito convenientes porque eles permitem manter o texto "amigável" e local, apresentando a informação de uma maneira acessível e não restringindo a informação em apenas uma página.

2.2.3.4. Nro. de Out *Links*

Número de *links outline* em uma página, isto é, aqueles *links* que apontam para páginas fora do site do hiperdocumento.

2.2.4. Implementação das métricas aplicadas a páginas

Para obter o número de *links* que são imagens são percorridas todas as âncoras fornecidas pelo sistema DB-LiOS, procurando por '<img' na primeira posição da *string* representando a âncora (veja exemplo anterior).

Quando a âncora é uma imagem, são considerados os valores de *width* e *height* e o seguinte cálculo é realizado para obter a superfície utilizada por essas imagens:

$$\sum_{i=1}^{L} width \times height$$

Onde L é o número de *links* em de uma página.

Para verificar o número de *In Links* em cada página, é considerado a *string* que representa o nó-destino e verifica-se se a *string* contém o site onde o nó fonte está localizado.

Para verificar o número de *Out Links* em cada página, é considerado a *string* que representa o nó-destino e verifica-se se a *string* não contém o site onde o nó-fonte está localizado. Portanto aponta para um página de outro site.

Todas as métricas descritas anteriormente foram implementadas, como descrito nas Seções 2.2.2 e 2.2.4, e tiveram suas métricas armazenadas em duas tabelas, uma com as métricas obtidas a partir dos sites, e a outra com as métricas obtidas com os dados das páginas.

2.3. Considerações Finais

As métricas estudadas e apresentadas nas seções anteriores foram implementadas neste trabalho. Em geral, essas métricas tiveram dados inicialmente obtidos a partir das informações fornecidas pelo sistema DB-LiOS.

Para a realização do experimento, os dados coletados por DB-LiOS tiveram que ser 'processados' de forma a representar as métricas escolhidas. Dessa forma, foram desenvolvidos módulos adicionais para processamento das estruturas de dados e computação

das métricas. Esses módulos foram implementados em Delphi para facilitar a manipulação da base de dados gerada por DB-LiOS. Para a implementação dos novos módulos foi necessário primeiramente entender a implementação do sistema DB-LiOS.

Para o entendimento da ferramenta DB-LiOS e estudo de como essas métricas seriam implementadas foi necessário um mês. A implementação dos módulos descritos nesta seção foi realizada em um mês.

A próxima seção apresenta como o experimento realizado com as métricas de qualidade de hiperdocumento foi conduzido.

3. EXPERIMENTO COM AS MÉTRICAS

3.1. Considerações Iniciais

Esta seção apresenta como o experimento com as métricas de qualidade de hiperdocumentos, apresentadas na seção anterior, foi realizado utilizando algoritmos de Aprendizado de Máquina (AM). Esse experimento teve como objetivo auxiliar o entendimento das métricas de qualidade de hiperdocumentos e possibilitar um embasamento maior sobre a "qualidade" dos *links* no domínio escolhido.

Em AM existem dois tipos de paradigmas de aprendizado – supervisionado e não supervisionado – e a escolha de qual deles usar depende dos exemplos da base, geralmente no formato atributo-valor, estarem ou não rotuladas com o atributo classe. Quando os dados estão rotulados é possível utilizar algoritmos de AM supervisionados, os quais induzem conceitos dos dados.

No caso dos dados não estarem explicitamente rotulados com uma classe, é possível utilizar algoritmos de AM não supervisionados, os quais procuram por padrões nos dados a partir de alguma caracterização de regularidade (Decker & Focardi 95). Esses padrões são denominados *clusters* (McCallum et al. 2000) e os exemplos dentro dos *clusters* são mais

similares que exemplos entre *clusters*. Basicamente, existem três razões de interesse no aprendizado não supervisionado:

(1) a coleção e o rotulamento de um grande conjunto de exemplos podem ser surpreendentemente caros em termos de custo e tempo;
(2) em muitas aplicações as características dos exemplos podem mudar vagarosamente com o passar do tempo;
(3) em alguns estágios da investigação, a descoberta de características pode ser valiosa para adquirir percepções da natureza ou da estrutura de dados.

Mas, apenas agrupar dados de acordo com alguma caracterização de similaridade, conceitualmente, pode ser pouco representativo. Muitas vezes, é de interesse do especialista do domínio tentar encontrar uma "interpretação" ou "explicação" para os dados contidos em cada *cluster*. Neste trabalho, utilizamos uma metodologia proposta em (Martins & Monard 2000) para, através de algoritmos de AM simbólicos, tentar interpretar os *clusters* obtidos utilizando algum algoritmo de AM não supervisionado. Na próxima seção é descrita a metodologia utilizada.

3.2. Metodologia utilizada no experimento

A metodologia utilizada é ilustrada na Figura 5. Basicamente, a metodologia é composta por quatro etapas. Primeiramente, uma base de dados já processada com exemplos não rotulados, no formato atributo-valor, é submetida a um processamento realizado por algum algoritmo de AM não supervisionado. Esse algoritmo é responsável por descobrir *clusters* presente na base de dados. Logo após, o resultado obtido (*clusters* encontrados) é processado por um mecanismo computável, que rotulará os exemplos da base original, ou um subconjunto desses exemplos, com o *cluster* ao qual pertencem. Assim é gerada uma base de dados com uma dimensão adicional, a qual é considerada como atributo classe desses exemplos. Essa nova base de dados possui as características necessárias para ser utilizada como entrada para algoritmos de AM supervisionados. Como o interesse principal é tentar explicar *clusters* previamente encontrados, a linguagem de descrição de conceitos (ou hipóteses) utilizada pelo algoritmo de AM supervisionado escolhido deve ser uma linguagem simbólica, tal como

regras ou árvores de decisão. Dessa forma, os *clusters* podem ser descritos, simbolicamente, através de regras ou árvores de decisão.

Figura 5. Metodologia utilizada (Martins & Monard 2000)

Após essa etapa, o conhecimento do especialista do domínio é de fundamental importância ao se tentar dar uma interpretação semântica aos *clusters*, agora descritos utilizando outro formalismo. Com a interpretação do especialista é possível, então, ter uma compreensão e uma "explicação" para os dados pertencentes a cada *cluster* encontrado. Na próxima seção é apresentado o processo utilizado para explicação dos *clusters*.

3.3. Processo utilizado para Explicação dos *Clusters*

Nesta seção é apresentado o processo, ilustrado na Figura 6, que utiliza a metodologia descrita anteriormente, usando o *Autoclass* como algoritmo de AM não supervisionado. *Autoclass* é um algoritmo de *clustering*, baseado na teoria Bayesiana em que o número de *clusters* pode ser especificado a *priori* ou encontrado automaticamente pelo próprio algoritmo. A saída gerada consiste de vários relatórios com descrições dos *clusters* encontrados e a probabilidade parcial dos exemplos nesses *clusters*. Esses relatórios são utilizados na etapa de rotulamento dos exemplos. O processo de rotulamento dos dados é realizado através de uma ferramenta denominada *InClass* descrita em (Martins & Monard 2000). Essa ferramenta utiliza como entrada um dos relatórios gerados por *Autoclass* e o conjunto de exemplos originais (não rotulados). A saída gerada por *InClass* é o conjunto de exemplos contendo agora uma dimensão adicional, pois contém o atributo classe relacionado ao *cluster* ao qual pertence cada exemplo.

Nesse processo, foi escolhido o *See5* como algoritmo de AM supervisionado. *See5* utiliza tanto regras quanto árvores de decisão como linguagem de descrição de conceitos. Assim, a

base de dados gerada por *InClass*, no formato requerido por *See5*, é processada por este para induzir regras de conhecimento. Tomando como base as regras geradas pelo *See5*, análise de erro e diversas estatísticas, o especialista pode realizar uma análise apurada para tentar explicar o agrupamento dos exemplos nos *clusters* encontrados.

Como descrito pela metodologia, o processo é finalizado com a análise do especialista sobre as regras de conhecimento geradas. É através dessa análise que realmente é possível verificar se o conhecimento gerado é importante, desconhecido ou útil.

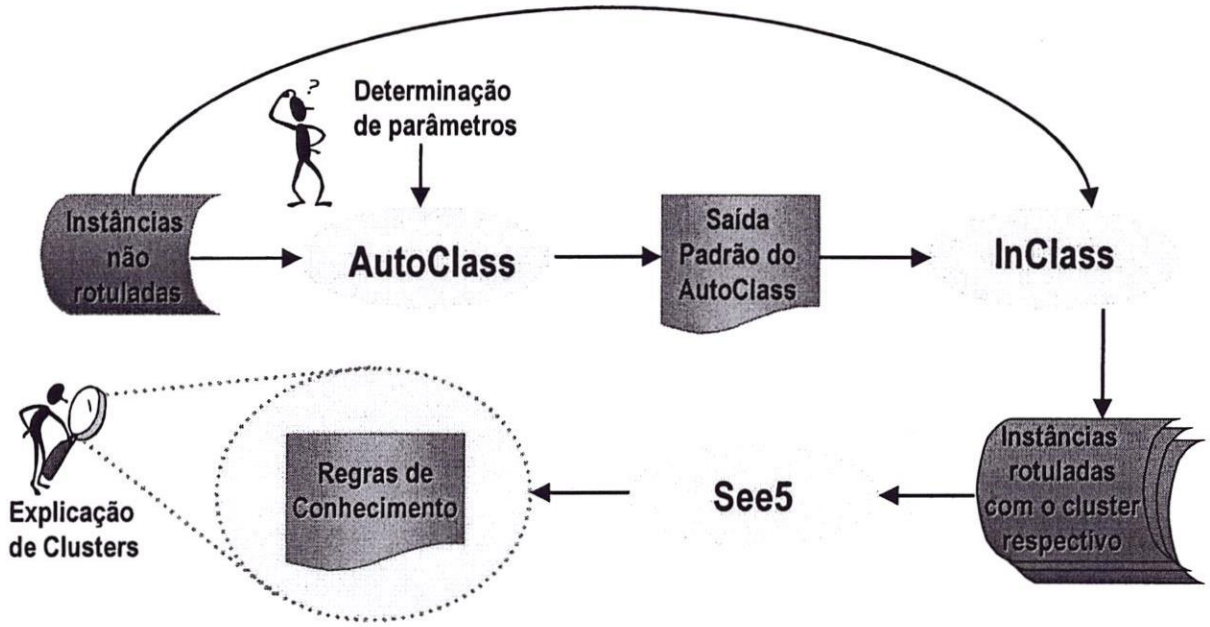


Figura 6. Processo utilizado para explicação dos *clusters* (Martins & Monard 2000)

A seguir, são descritas resumidamente as características principais dos algoritmos *AutoClass* e *See5*, bem como da ferramenta *InClass*, utilizados nesse processo.

3.3.1. Autoclass

AutoClass é um algoritmo de aprendizado não supervisionado baseado na teoria Bayesiana, desenvolvido pelo grupo de Bayes no Ames Research Center (Cheeseman & Stutz 90).

Basicamente, *AutoClass* descreve *clusters* a partir da distribuição probabilística sobre os atributos dos exemplos, considerando que existe independência condicional nos dados.

AutoClass tem sido usado e testado em muitos conjuntos de dados, pela NASA e pela indústria, meio acadêmico e outras agências. Foram encontradas e mostradas algumas classificações surpreendentes, que mostram padrões nos dados muitas vezes desconhecidos para o especialista da área. *AutoClass* é um algoritmo robusto e de domínio público, que apresenta, basicamente, as seguintes características:

(1) determina o número de *clusters* automaticamente ou permite que seja definido pelo usuário;
(2) os valores dos atributos podem ser tanto contínuos quanto discretos;
(3) manipula valores ausentes e desconhecidos;
(4) tempo de processamento é robustamente linear;
(5) gera relatórios descrevendo os *clusters* encontrados e prediz o *cluster* de novos exemplos.

AutoClass procura a melhor classificação que possa encontrar nos dados. Uma classificação poderá ser a descoberta de um conjunto de *clusters*, descrevendo qual a porcentagem provável dos exemplos estarem em cada *cluster*, e uma denominação probabilística dos exemplos para esses *clusters*. Isto é, para cada exemplo, a probabilidade relativa de ser membro de cada *cluster*. A entrada para o algoritmo consiste de um conjunto de dados na forma atributo-valor, definição de modelos e de parâmetros de busca. O *AutoClass* procura um conjunto de *clusters* que seja altamente provável com os dados e modelos especificados. A saída desse algoritmo é a descrição probabilística dos *clusters* identificados e dos exemplos pertencentes a esses *clusters*. O próprio algoritmo não impõe nenhum limite específico no número de dados, mas bases de dados com mais de 100.000 valores (número de exemplos x números de atributos) podem necessitar de tempo de execução excessivo.

3.3.2. See5

See 5 é um produto comercial para plataforma Windows™ que inclui melhorias dos algoritmos *C*4.5 e *C*4.5rules (Quinlan 93), que têm sido usados, freqüentemente, para

comparar seu desempenho com outros algoritmos de AM. O *See5* foi projetado para trabalhar com bases de dados relativamente grandes. Como seus precursores, manipula atributos com valores discretos e contínuos, induzindo conceitos expressos como árvores de decisão ou conjunto de regras não ordenadas **if-then** (Baranauska & Monard 2000). Seu desempenho tem se demonstrado muito bom na maioria dos casos.

3.3.3. Inclass

Para que o processo de rotulamento dos exemplos fosse feito de forma automática, utilizamos a ferramenta computacional *InClass*, descrita em (Martins & Monard 2000), implementada na linguagem de programação PERL[2] (Wall et al. 96). Os dados de entrada para a ferramenta são a base de dados original e um dos relatórios gerados pelo *AutoClass*. Com base nesse relatório do *AutoClass* e na base de dados original (exemplos não rotulados), o *InClass* cria uma nova base de dados com os exemplos originais rotulados com o *cluster* respectivo. Pode-se considerar, na construção da nova base, utilizando *InClass*, todos os exemplos ou apenas exemplos que pertencem, com uma dada probabilidade, aos *clusters* encontrados por *AutoClass*, dependendo dos parâmetros especificados pelo especialista.

A seguir é descrito como o experimento foi realizado utilizando o processo descrito anteriormente.

3.4. O experimento com as métricas de sites e páginas

Este experimento utiliza o processo apresentado na seção anterior e foi dividido em seis etapas, apresentadas na Tabela 4. Os dados utilizados no experimento são tabelas no formato atributo-valor, uma com os dados dos sites, utilizando como atributos: Métrica de Compactação, Métrica de Estratificação, Métrica de Impureza da árvore, Nro. de páginas que voltam à página anterior e Nro. de páginas que voltam à homepage e as outras com os dados das páginas, utilizando como atributos: Nro. de *links* que são imagens, Nro. de *In Links*, Nro. de *Out Links* e Superfície dessas imagens.

---

[2] Pratical Extraction and Report Language

Tabela 4. Etapas do experimento

Etapa	Descrição
1	Utilização do sistema DB-LiOS
2	Aplicação dos módulos responsáveis pela obtenção dos valores das métricas de sites e páginas
3	Utilização do *Autoclass*
4	Utilização do *Inclass*
5	Utilização do *See5*
6	Análise dos resultados

Nas seções seguintes são descritas cada uma das etapas do experimento.

3.4.1. Utilização do sistema DB-LiOS

O sistema DB-LiOS, apresentado na Figura 7, foi utilizado para extrair as informações necessárias para aplicação dos módulos responsáveis pela coleta das métricas de sites e páginas. Os sites utilizados no experimento são apresentados na Tabela 5, juntamente com algumas informações adicionais extraídas pelo sistema DB-LiOS.

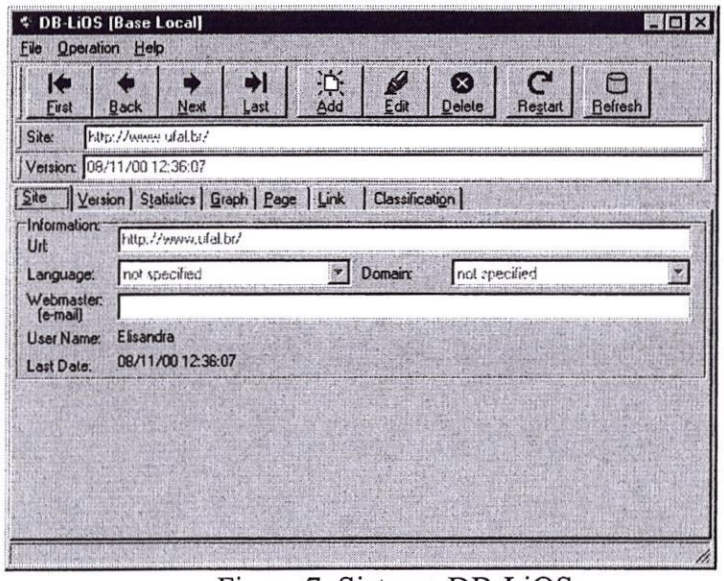


Figura 7. Sistema DB-LiOS

Tabela 5. Os sites utilizados no experimento

Número	Site	Número de páginas	Número de *links* estáticos
1	www.ufal.br	149	634
2	www.uesb.br	190	601
3	www.ufba.br	1201	7889
4	www.uece.br	244	771
5	www.ufes.br	858	6573
6	www.pucrs.br	56	43
7	www.pucsp.br	40	164
8	www.unimontes.br	160	366
9	www.unisul.rct-sc.br	94	1069
10	www.ime.eb.br	119	341
11	www.eep.br	40	49
12	www.fafica.br	48	171
13	www.ufv.br	956	5471
14	www.ufpb.br	56	432
15	www.ufpe.br	125	708
16	www.ita.cta.br	105	3216
17	www.ufop.br	312	2828
18	www.ufpa.br	801	4235
19	www.puc-rio.br	336	3724
20	www.puc-campinas.br	359	1210
21	www.unaerp.br	152	3544
22	www.ufma.br	171	1149
23	www.ufmt.br	77	218
24	www.ufmg.br	387	3744
25	www.ufrj.br	190	2014
26	www.icmc.sc.usp.br	186	2654

3.4.2. Aplicação dos módulos responsáveis pela obtenção dos valores das métricas de sites e páginas

Os módulos responsáveis por obter os valores das métricas, aplicados aos vinte e seis sites de universidades brasileiras, são:

(1) Módulo responsável pela obtenção dos dados das páginas de cada site

(2) Módulo responsável pela obtenção dos dados dos sites

Aplicando-se o módulo (1) obtemos uma tabela para cada site, com os seguintes dados: Nro. de *In Links*, Nro. de *Out Links*, Nro. de *links* que são imagens, Superfície dessas imagens. A Tabela 6 apresenta alguns dos valores obtidos com a aplicação deste módulo para algumas das 149 páginas do site número 1.

Aplicando-se o módulo (2) obtemos uma tabela, com os seguintes dados: Compactação, Estratificação, Métrica de Impureza da árvore, Nro. de páginas que voltam à página anterior, Nro. de páginas que voltam à homepage. A Tabela 7 apresenta os valores obtidos com a aplicação desse módulo aos 26 sites.

Tabela 6. Algumas páginas e valores para as métricas de páginas do site número 1.

Site 1			
In Links	*Out Links*	Imagens	Superfície
0	0	0	0
0	1	0	0
9	0	0	0
1	1	1	5396
3	1	0	0
4	5	0	0
10	2	1	4080
4	1	1	5396
1	0	1	1250
0	0	0	0
1	0	0	0
4	3	3	12028,67
3	0	0	0
11	2	3	225
34	0	0	0
1	5	0	0
10	2	1	4080
0	1	1	88080
10	1	5	17183,4

Tabela 7. Os valores obtidos para as métricas de sites

Site	Compactação	Estratificação	Impureza da árvore	Nro-páginas que voltam à pág. anterior	Nro-páginas que voltam à homepage
1	0.2213	0.3244	0.0446	113	0
2	0.0092	0.0672	0.0231	1	10
3	0.0862	0.2919	0.0092	773	21
4	0.0128	0.0926	0.0179	0	0
5	0.2141	0.9367	0.0155	736	13
6	0.0094	0.0701	-0.0080	0	0
7	0.0654	0.3864	0.1686	6	0
8	0.2807	0.2265	0.0164	100	9
9	0.9317	0.3167	0.2281	121	20
10	0.0318	0.2064	0.0323	4	14
11	0.0285	0.1928	0.0134	0	0
12	0.2646	0.4111	0.1147	33	0

13	0.1062	0.1527	0.0099	693	0
14	0.0550	0.2957	0.2538	6	11
15	0.1575	0.5886	0.0765	179	0
16	0.4572	1.3914	0.5810	219	15
17	0.5083	1.1736	0.0522	314	47
18	0.1429	0.5880	0.0107	612	31
19	0.5516	0.5769	0.0605	240	12
20	0.0372	0.1314	0.0133	122	2
21	0.7908	0.2323	0.2996	205	11
22	0.2869	0.9397	0.0681	215	0
23	0.0609	0.2497	0.0498	44	8
24	0.0617	0.2048	0.0451	552	19
25	0.8627	0.6604	0.1027	187	29
26	0.2300	1.0956	0.1450	143	13

A próxima etapa do experimento é a utilização do *Autoclass*, descrita na próxima seção.

3.4.3. Utilização do Autoclass

Após o uso da ferramenta para a coleta dos dados, os mesmos foram preparados para a utilização do *Autoclass*. Como já mencionado, o *Autoclass* é responsável por descrever *clusters* a partir da distribuição probabilística sobre os atributos dos exemplos. A Tabela 8 apresenta um dos relatórios gerados pelo *Autoclass*, o qual é utilizado por *Inclass*, na próxima etapa, para rotular os exemplos.

Tabela 8. Um dos resultados do *Autoclass*

Site	Classe	Probabilidade
1	classe_0	0.997
2	classe_2	1.000
3	classe_1	1.000
4	classe_2	1.000
5	classe_0	1.000
6	classe_2	1.000
7	classe_0	1.000
8	classe_1	1.000
9	classe_0	1.000
10	classe_1	1.000
11	classe_1	1.000
12	classe_0	1.000
13	classe_1	1.000
14	classe_0	1.000

15	classe_0	1.000
16	classe_0	1.000
17	classe_0	1.000
18	classe_0	1.000
19	classe_0	1.000
20	classe_1	1.000
21	classe_0	1.000
22	classe_0	1.000
23	classe_1	1.000
24	classe_1	1.000
25	classe_0	1.000
26	classe_0	1.000

3.4.4. Utilização do Inclass

Utilizando *Inclass* para o conjunto de dados apresentado na Tabela 8, obtemos o resultado apresentado na Tabela 9, no formato requerido por *See5*.

Tabela 9. Resultado do *Inclass*

Site	Compactação	Estratificação	Impureza da árvore	Nro-páginas que voltam à pág. anterior	Nro-páginas que voltam à homepage	Classe
1	0.2213	0.3244	0.0446	113	0	classe_0
2	0.0092	0.0672	0.0231	1	10	classe_2
3	0.0862	0.2919	0.0092	773	21	classe_1
4	0.0128	0.0926	0.0179	0	0	classe_2
5	0.2141	0.9367	0.0155	736	13	classe_0
6	0.0094	0.0701	-0.0080	0	0	classe_2
7	0.0654	0.3864	0.1686	6	0	classe_0
8	0.2807	0.2265	0.0164	100	9	classe_1
9	0.9317	0.3167	0.2281	121	20	classe_0
10	0.0318	0.2064	0.0323	4	14	classe_1
11	0.0285	0.1928	0.0134	0	0	classe_1
12	0.2646	0.4111	0.1147	33	0	classe_0
13	0.1062	0.1527	0.0099	693	0	classe_1
14	0.0550	0.2957	0.2538	6	11	classe_0
15	0.1575	0.5886	0.0765	179	0	classe_0
16	0.4572	1.3914	0.5810	219	15	classe_0
17	0.5083	1.1736	0.0522	314	47	classe_0
18	0.1429	0.5880	0.0107	612	31	classe_0
19	0.5516	0.5769	0.0605	240	12	classe_0
20	0.0372	0.1314	0.0133	122	2	classe_1
21	0.7908	0.2323	0.2996	205	11	classe_0

22	0.2869	0.9397	0.0681	215	0	classe_0
23	0.0609	0.2497	0.0498	44	8	classe_1
24	0.0617	0.2048	0.0451	552	19	classe_1
25	0.8627	0.6604	0.1027	187	29	classe_0
26	0.2300	1.0956	0.1450	143	13	classe_0

Deve-se observar que apenas *links* estáticos foram extraídos, portanto alguns valores como os obtidos para o site número 4 (Nro. de páginas que voltam à página anterior = 0 e Nro. de páginas que voltam à homepage = 0) devem-se ao fato de trabalharmos apenas com *links* estáticos.

3.4.5. Utilização do See5

Esse conjunto de exemplos é utilizado por *See5* para induzir um conjunto de regras não ordenadas **if-then**. O conjunto de regras induzidas por *See5* foram então analisadas para tentar encontrar um significado semântico para o nome dos *clusters*. Na próxima seção são descritos os resultados obtidos.

3.4.6. Análise dos resultados

Nesta seção são apresentados os resultados do experimento. Esse experimento foi realizado em duas etapas, uma com os dados das páginas e outra com os dados de sites. As próximas seções apresentam os resultados obtidos em ambas etapas do experimento.

3.4.6.1. Experimento com os dados das páginas

O experimento realizado com os dados coletados das páginas não ofereceram resultados muito significativos. O atributo presente na maioria das regras e com maior importância foi o atributo Superfície das imagens, que indica a superfície utilizada pelas imagens em uma dada página. Futuramente, pretendemos realizar outras experiências com esses dados, utilizando indução construtiva que permite criar novos atributos como uma combinação dos atributos existentes (Russel & Norvig 95).

3.4.6.2. Experimento com os dados de sites

A base de dados de todos os sites foi submetida ao *Autoclass* sem fixar um número de *clusters*, tendo sido encontrados 3 *clusters*. A Tabela 10 apresenta um resumo dos resultados obtidos, onde:

- Clusters – representa os *clusters*;
- # Exemplos – número de exemplos na base de dados;
- % Classe- porcentagem de exemplos pertencentes ao *cluster*;
- Erro aparente – erro aparente de *See5*, isto é, utilizando toda a base de dados de treinamento e teste;
- Erro verdadeiro (10CV) – erro verdadeiro obtido através de 10k-fold *cross-validation*;
- Erro CM – erro da classe majoritária;
- # Regras – número de regras encontradas por *See5*;
- # Médio Regras – número médio de regras obtido nas 10 execuções de *cross-validation.*

Tabela 10. Experimento com todos os sites – Resumo dos resultados

# Exemplos	Clusters	% Classe	Erro Aparente	Erro (10 CV) Verdadeiro	Erro CM	# Regras	# Médio Regras
26	C(0)	53,84	3,8%	15% ± 6,3%	46,14%	3	3 ± 0
	C(1)	34,61					
	C(2)	11,53					

```
Rule 1: (cover 14)                Rule 2: (cover 9)                 Rule 3: (cover 3)
        Estratificacao > 0.2919           Estratificacao > 0.0926           Estratificacao <= 0.0926
        -> class classe_0 [0.938]         Estratificacao <= 0.2919          -> class classe_2 [0.800]
                                          -> class classe_1 [0.818]
```

Figura 8. Regras extraídas para todos os sites

As regras obtidas são apresentadas na Figura 8. Nessas regras aparece apenas o atributo Estratificação, o que indica que somente esse atributo consegue diferenciar os exemplos (sites) considerados.

Diversos outros experimentos foram realizados. Um deles, consistiu em rodar *See5* agrupando as classe_2 e classe_1 em um único *cluster* denominado nova_classe_1. A Tabela 11 apresenta o resumo dos resultados obtidos e a Figura 9 apresenta as regras extraídas por *See5*.

Tabela 11. Experimento com todos os sites com classe_0 e nova_classe_1 – Resumo dos resultados

# Exemplos	Clusters	% Classe	Erro Aparente	Erro (10 CV) Verdadeiro	Erro CM	# Regras	# Médio Regras
26	C(0)	53,84	3,8%	6,7% ± 4,4%	46,15%	2	2 ± 0
	C(1)	46,15					

```
Rule 1: (cover 14)                    Rule 2: (cover 12)
        Estratificacao > 0.2919               Estratificacao <= 0.2919
        -> class classe_0 [0.938]             -> class nova_classe_1 [0.857]
```

Figura 9. Regras extraídas para todos os sites com classe_0 e nova_classe_1

Como esperado, o atributo estratificação consegue, também neste caso, diferenciar os sites considerados.

A fim de verificar a importância de outros atributos para explicar os *clusters* foi realizado outro experimento ignorando o atributo Estratificação. Os resultados obtidos são apresentados na Tabela 12. As regras são apresentadas na Figura 10. Pode ser observado que tais são os atributos utilizados pelas regras: Compactação e Impureza da árvore. Assim, pode-se considerar que os atributos Nro. de páginas que voltam à página anterior e Nro. de páginas que voltam à homepage não são fortemente relevantes para explicar esses exemplos.

Tabela 12. Experimento com todos os sites sem o atributo Estratificação – Resumo dos resultados

# Exemplos	Clusters	% Classe	Erro Aparente	Erro (10 CV) Verdadeiro	Erro CM	# Regras	# Médio Regras
26	C(0)	61,53	0,0%	10% ± 5,1%	38,45	3	3 ± 0
	C(1)	26,92					
	C(2)	11,53					

```
Rule 1: (cover 16)                Rule 2: (cover 7)                Rule 3: (cover 3)
        Impureza_arvore > 0.0231          Compactacao > 0.0128             Compactacao <= 0.0128
        -> class classe_0 [0.944]         Impureza_arvore <= 0.0231        -> class classe_2 [0.800]
                                          -> class classe_1 [0.889]
```

Figura 10. Regras extraídas para todos os sites sem o atributo Estratificação

3.5. Considerações Finais

Esta seção descreveu como foi realizado o experimento com as métricas de qualidade de hiperdocumentos utilizando sistemas de aprendizado de máquina. As métricas analisadas são freqüentemente citadas na literatura atual mas pouca avaliação empírica dessas métricas é relatada.

O experimento, como descrito nesta seção, foi dividido em seis etapas: (1) Utilização do sistema DB-LiOS, (2) Aplicação dos módulos responsáveis pela obtenção dos valores das métricas de sites e páginas, (3) Utilização do *Autoclass*, (4) Utilização do *Inclass*, (5) Utilização do *See5*, (6) Análise dos resultados.

A primeira etapa foi a que requereu mais tempo, pois as dificuldades na procura por sites "apropriados" para o experimento e na coleta de seus dados da WWW foram freqüentes. Entendendo por sites "apropriados" aqueles que possuíam um número mínimo de páginas (40) e que não esgotassem um tempo de espera de duas horas na coleta. Enfrentamos também

problemas na rede do Instituto e dessa forma, a coleta dos dados tomou um tempo maior do que o esperado.

O número inicial de sites que pensamos avaliar era de 50 sites, mas não conseguimos coletar esse número de sites e decidimos trabalhar com os 26 sites apresentados neste relatório.

Utilizando DB-LiOS, obtivemos os dados necessários para a segunda etapa. Com os dados necessários para a coleta dos valores das métricas, a segunda etapa foi realizada conforme o esperado, utilizando os módulos responsáveis pelos cálculos e obtenção dos valores para as métricas. Esses valores calculados foram então armazenados em 27 tabelas, sendo que uma com os valores calculados de todos os sites e as outras com os dados das páginas de cada site. Os dados contidos nessas tabelas foram então preparados para a utilização do *Autoclass*.

Autoclass requer alguns arquivos para sua execução, tais como arquivos de dados, arquivos de definições de tipos, etc.... Esses arquivos criados para execução do algoritmo. Primeiramente, executamos *Autoclass* sem fixar um número de *clusters* e por um tempo indeterminado. Após observar os dados, executamos novamente *Autoclass* fixando o número de *clusters* em 3.

Com os resultados do *Autoclass*, executamos o *Inclass* para que os dados pudessem ser fornecidos ao *See5*. Para executar o *Inclass*, os arquivos de entrada foram também preparados, ou seja, todas as tabelas tiveram que sofrer uma reformatação. A execução não tomou muito tempo.

See5 também requer alguns arquivos adicionais de definições de tipos de dados além do arquivo resultante de *Inclass*. Executando *See5* obtemos as regras para a análise dos resultados finais. Nessas regras apareceu apenas o atributo Estratificação, o que indica que somente esse atributo conseguiu diferenciar os exemplos (sites) considerados.

Os dados utilizados no experimento são os dados extraídos por DB-LiOS (apenas *links* estáticos). Com os valores para as métricas percebemos que alguns sites possuem poucos *links*

estáticos e possivelmente devem utilizar outro tipo de *links*. Portanto, nossa análise ficou restrita a esse tipo de *links*.

A etapa que consumiu mais tempo foi a primeira, tendo consumido dois meses. Para as outras etapas mais um mês foi necessário.

Referências Bibliográficas

(Andrews et al. 95a) Andrews, Keith, Nedoumov, Andrew, and Scherbakov, Nick: "Embedding Courseware into the Internet: Problems and Solutions"; Proceedings of ED-MEDIA 95, Graz (1995), 69-74.

(Andrews et al. 95b) Andrews, Keith, Kappe, Frank, Maurer, Hermann, and Schmaranz, Klaus: "On Second Generation Network Hypermedia Systems"; Proceedings of ED-MEDIA 95, Graz (1995), 75-80.

(Balasubramanian et al. 94) Balasubramanian, P., Isakowitz, Tomás, and Stohr, Edward A.: "Designing Hypermedia Applications"; Proceedings of the Twenty-Seventh Annual Hawaii International Conference on System Sciences, Hawaii (1994), 354-365.

(Baranauska & Monard 2000) Baranauskas, J. A. & Monard, M. C.: Reviewing some machine learning concepts and methods. Technical Report 102, ICMC-USP. ftp://ftp.icmc.sc.usp.br/pub/BIBLIOTECA/rel-tec/Rt_102.ps.zip.

(Basili et al. 94) Basili, V., Caldiera G., and Rombach, D.: "The Goal Question Metric Approach", Encyclopedia of Software Engineering, Wiley (1994).

(Bernstein et al. 91) Bernstein, Mark, Brown, Peter J., Fisse, Mark, Glushko, Robert, Landow, George, Zellweger, Polle: "Structure, Navigation, and Hypertext: The Status of the Navigation Problem"; Proceedings of Hypertext 91, ACM Press, San Antonio (1991), 363-367.

(Botafogo et al. 92) Botafogo, Rodrigo A., Rivlin, Ehud, and Shneiderman, Ben: "Structural Analysis of Hypertexts: Identifying Hierarchies and Useful Metrics", ACM TOIS, 10, 2 (1992), 143-179.

(Botafogo & Shneiderman 92) Botafogo, R. A.; Shneiderman, B.: "Structural Analysis of Hypertexts: Identifying Hierarchies and Useful Metrics" *ACM Transactions on Information Systems,* v.10, n.2, p.142-80, April, 1992.

(Briand et al. 97) Briand, L., Devandu, P. and Melo, M.: "An Investigation into Coupling Measures for C++ "; Proceedings of ICSE 97, Boston (1997), 412-421.

(Catlin & Garrett 91) Catlin, Karen Smith, and Garrett, L. Nancy: "Hypermedia Templates: An Author s Tool"; Proceedings of Hypertext 91, ACM Press, San Antonio (1991), 147-160.

(Cheeseman & Stutz 90) Cheeseman, P. & Stutz, J.: Bayesian classification (autoclass): Theory and results advances in knowledge discovery and data mining. http://ic.arc.nasa.gov/ic/projects/bayes-group/autoclass-c-program.html.

(Davis et al. 92) Davis, Hugh, Hall, Wendy, Heath, Ian, Hill, Gary, and Wilkings, Rob: "Towards an Integrated Information Environment With Open Hypermedia Systems"; Proceedings of the ACM Conference on Hypertext, ACM Press, Milan (1992), 181-190.

(DeBra 99) DeBra P. M. E.: Hypermedia Structure and Systems. Eidhoven University of Technology, http://wwwis.win.tue.nl/2L670/static/ (1999).

(Decker & Focardi 95) Decker, K. M., Focardi, S.: Technology overview: A report on data mining. Technical report, CSCS-ETH, Swiss Scientific Computing Center (1995).

(Duval & Olivié 95) Duval, Erik, and Olivié, Henk: "A Home for Networked Hypermedia"; Proceedings of ED-MEDIA 95, Graz (1995), 193-198.

(Fenton 91) FENTON, N. *Software Metrics*. New York, Chapman & Hall, 1991.

(Fortes 96) Fortes, R. P. M.: "Análise e Avaliação de Hiperdocumentos: uma abordagem baseada na Representação Estrutural". Tese de Doutorado, IFSC-USP, São Carlos - SP. 30 de agosto de 1996. 179p.

(Fortes & Nicoletti 97) Fortes, R. P. M. & Nicoletti, M. C.: "A Family of Link Based Metrics for the Evaluation of Web Documents" SIGLINK Bulletin, v.6, n.3, p.21-23, October 1997.

(Fortes & Nicoletti 99) Fortes, R.P.M.; Nicoletti, M.C. "Automatic Diagnosis of Hyperdocuments Using a Family of Quantifiable Metrics" In: SoST'99 (Symposium on Software Technology) *Proceedings* Buenos Aires - Argentina, September 1999, p.62-71.

(Garzotto et al. 91) Garzotto, Franca, Paolini, Paolo, and Schwabe, Daniel: "HDM - A Model for the Design of Hypertext Applications"; Proceedings of Hypertext 91, ACM Press, San Antonio (1991), 313-328.

(Garzotto et al. 94) Garzotto, Franca, Mainetti, Luca, and Paolini, Paolo: "Analysing the Quality of Hypermedia Applications: A Design-Oriented Framework", Workshop on hypermedia design and development, Edinburgh (1994).

(Garzotto et al. 95) Garzotto, Franca, Mainetti, Luca, and Paolini, Paolo: "Hypermedia Design, Analysis, and Evaluation Issues", Communications of the ACM, Special Issue on Hypermedia Design, August (1995).

(Goldberg et al. 96) Goldberg, M. W., Salari, S., and Swoboda, P.: "World Wide Web - Course Tool: An environment for building WWW-based courses"; Proceedings of the Fifth International World Wide Web Conference, Paris (1996), 1219-1232.

(Hatzimanikatis et al. 95) Hatzimanikatis, A. E., Tsalidis, C. T., and Christodoulakis, D.: "Measuring the Readability and Maintainability of Hyperdocuments"; J. of Software Maintenance, Research and Practice, 7 (1995), 77-90.

(Jalote 97) JALOTE, P. *An Integrated Approach to Software Engineering*. 2nd Edition, Springer, USA, 1997.

(Jordan et al. 89) Jordan, Daniel S., Russell, Daniel M., Jensen, Anne-Marie S., and Rogers, Russell A.: "Facilitating the Development of representations in Hypertext with IDE"; Proceedings of Hypertext 89, ACM Press, Pittsburgh (1989), 93-104.

(Kan 95) KAN, S.H. *Metrics and Models in Software Quality Engineering*. USA, Addison-Wesley, 1995.

(McCallum et al. 2000) McCallum, A., Nigam, K., Ungar, L.: Efficient clustering of high-dimensional data sets with application to reference matching. *KDD* (2000).

(Marmann et al. 92) Marmann, Michael, and Schlageter, Gunter: "Towards a Better Support for the Hypermedia Structuring: The HYDESIGN Model"; Proceedings of the ACM European Conference on Hypertext, ACM Press, Milano (1992), 11-22.

(Marshall et al. 91) Marshall, Catherine C., Halasz, Frank G., Rogers, Russell A., and Jr., William C. Janssen: "Aquanet: a hypertext tool to hlod your knowledge in place"; Proceedings of Hypertext 91, ACM Press, San Antonio (1991), 261-275.

(Marshall et al. 95) Marshall, Catherine C., and Shipman III, Frank M.: "Spatial Hypertext: designing for Change"; Communications of the ACM, Special Issue on Hypermedia Design, August (1995).

(Martins & Monard 2000) Martins, C. A., Monard, M. C.: "Interpretação de Clusters Utilizando Aprendizado de Máquina Simbólico"; *21 Iberian Latin American Congress on Computational Methods in Engineering – CILAMCE2000,* Rio de Janeiro (RJ), 2000.

(Mendes et al. 98) Applying Metrics to the Evaluation of Educational Hypermedia Applications in Journal Universal Computer Science Vol 4 / No 4 / p382-403

(Meyrowitz 86) Meyrowitz, N.: "Intermedia: The Architecture and Construction of an Object-oriented Hypermedia System and Applications framework"; Proceedings of the OOPSLA 86, (1986), 186-201.

(Nanard & Nanard 95) Nanard, Jocelyne, and Nanard, Marc: "Hypertext Design Environments and the Hypertext Design Process"; Communications of the ACM, Special Issue on Hypermedia Design, August (1995), 49-56.

(Papast 97) Papast, P.: Hypermedia Navigation Metrics, http://ise.eng.uts.edu.au/ise/hyptech/miniprj/mnprj97a/papast/metrics.html (1997).

(Russel & Norvig 95) Russel, S. J.; Norvig, P. *Artificial Intelligence: A Modern Approach.* Prentice Hall, 1995.

(Quinlan 93) Quinlan, J. R.: *C4.5: Programs for Machine Learning.* MK, Los Altos, California, USA.

(Rivlin et al. 94) Rivlin, Ehud, Botafogo, Rodrigo, and Schneiderman, Ben: "Navigating in Hyperspace: designing a structure-based toolbox"; Communications of the ACM, 37, 2 (1994), 87-96.

(Rossi et al. 95) Rossi, G., Schwabe, D., C. Lucena, J. P., and Cowan, D. D.: "An Object-Oriented Model for Designing Human-Computer Interface of Hypermedia Applications"; Proceedings of the IWHD 95, Montpellier (1995), 131-152.

(Seraphim & Fortes 2000) Seraphim, E.; Fortes, R. P.M. - "DB-LiOS: Suporte Automático à Avaliação da Consistência de *Links* em *WWW*" - In: XX Congresso Nacional da SBC, XXVII SEMISH, *Anais*. Curitiba-PR, julho de 2000.

(Szwarcfiter 84) Szwarcfiter, J. L. Grafos e algoritmos computacionais. Rio de Janeiro, Campus, 1984.

(Thimbleby 96) Thimbleby, Harold: "Systematic web authoring", The British HCI Symposium The Missing Link: Hypermedia Usability Research & The Web, UK (1996), http://www.cs.mdx.ac.uk/harold/webpaper/.

(Thistlewaite 95) Thistlewaite, P.: "Automatic Construction of Open Webs using Derived Link Patterns". In M. Agosti and J. Allan (Eds.), IR and Automatic Construction of Hypermedia: a Research Workshop. ACM SIGIR, 1995.

(Türing et al. 95) Türing, Manfred, Hannemann, Jörg, and Haake, Jörg: "Hypermedia and Cognition: Designing for Compreension", Communications of the ACM, Special Issue on Hypermedia Design, August (1995).

(Yamada et al. 95) Yamada, Shoji, Hong, Jung-Kook, and Sugita, Shigeharu: "Development and Evaluation of Hypermedia for Museum Education: Validation of Metrics"; ACM Transactions on Computer-Human Interaction, 2, 4 (December, 1995), 284-307.

(Wall et al. 96) Wall, L., Christiansen, T. Schwartz, R. L. *Programming in PERL.* O'Reilly, Inc. (1996).