Uma Abordagem Efetiva e Eficiente para Deduplicação de Metadados Bibliográficos de Objetos Digitais

Eduardo N. Borges[1], **Renata M. Galante**[1], **Marcos A. Gonçalves**[2]

[1]Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brasil

[2]Departamento de Ciência da Computação – Universidade Federal de Minas Gerais (UFMG) – 31.290-901 – Belo Horizonte – MG – Brasil

{enborges,galante}@inf.ufrgs.br, mgoncalv@dcc.ufmg.br

Abstract. *Digital libraries contain collections of digital objects, acquired from different sources, which can be represented through several metadata standards. These metadata are heterogeneous both in content and in structure. This paper presents an approach that identifies duplicated metadata records referring to objects from digital libraries. We propose similarity functions designed for the digital library domain that compare the content of metadata. The results of experiments show that the proposed functions, compared to three different baselines, improve the quality of metadata deduplication from 0.64 to 31.5% using an algorithm with linear complexity to compare authors' names.*

Resumo. *Bibliotecas digitais são compostas por coleções de objetos digitais, adquiridos de fontes distintas, os quais podem estar representados através de vários padrões de metadados. Estes metadados são heterogêneos em conteúdo e estrutura. Este artigo apresenta uma abordagem para identificar metadados de objetos duplicados em bibliotecas digitais. São propostas funções de similaridade específicas para o domínio das bibliotecas digitais que comparam o conteúdo dos metadados. Os resultados dos experimentos realizados mostram que as funções propostas, quando comparadas a três abordagens distintas, melhoram a qualidade da deduplicação de metadados de 0,64 a 31,5% utilizando um algoritmo de complexidade linear para a comparação de nomes de autores.*

1. Introdução

Bibliotecas digitais são compostas por coleções de objetos digitais, como, por exemplo, documentos, imagens, mapas, etc. e oferecem serviços aos seus usuários como pesquisa e publicação desses objetos [Fox et al. 1995]. Além dos objetos digitais, as bibliotecas digitais possuem um catálogo de metadados cuja função é descrever, organizar e especificar a forma como esses objetos podem ser manipulados e recuperados. Por exemplo, o *Dublin Core* [DCMI 2008] define um padrão para a representação, armazenamento e consulta de informações a respeito de artigos científicos, periódicos e páginas *Web*. A *International DOI Foundation* define um *Digital Object Identifier* (DOI) como um identificador permanente de qualquer objeto de propriedade intelectual que, ao contrário de um *Uniform Resource Locator* (URL), é independente da localização do objeto [DOI 2008]. Entretanto, não há um consenso na utilização dos padrões por todas as bibliotecas digitais existentes.

Considere o exemplo da Figura 1 onde são apresentados três conjuntos de metadados oriundos de diferentes bibliotecas digitais, mas que referenciam o mesmo objeto digital. O elemento *source* presente nos metadados da BDBComp[1] (linha 3) corresponde ao elemento *booktitle* na DBLP[2] (linha 6). As estruturas dos metadados, apesar de diferentes, fazem referência à mesma informação, ou seja, ao veículo de publicação do objeto digital. Ainda são identificados problemas na variação do conteúdo. A autoria do objeto digital, representada pelos metadados *creator* e *author*, possui conteúdo "Carlos H. Morimoto" na BDBComp (linha 2), "Carlos Hitoshi Morimoto" na DBLP (linha 5) e "Morimoto, C.H." na IEEE Xplore[3] (linha 8). Os metadados *title* também diferem na palavra *Remote* (linhas 1;4;7).

BDBComp

```
1  <title>A Computer Vision Framework for Remote Eye Gaze Tracking</title>
2  <creator>Carlos H. Morimoto</creator>
3  <source>sibgrapi2003</source>
```

DBLP

```
4  <title>A Computer Vision Framework for Eye Gaze Tracking</title>
5  <author>Carlos Hitoshi Morimoto</author>
6  <booktitle>SIBGRAPI</booktitle>
```

IEEE Xplore

```
7  <title>A computer vision framework for eye gaze tracking</title>
8  <author>Morimoto, C.H.</author>
9  <pages>406</pages>
```

Figura 1. Heterogeneidade de metadados.

A tarefa de identificar em um repositório de dados registros duplicados que referem-se a mesma entidade do mundo real, incluindo variações de grafia e omissão de palavras, é denominada deduplicação [Carvalho et al. 2006]. Conhecida também como casamento de registros (*record linkage*), de objetos (*object matching*) ou de instâncias (*instance matching*), a deduplicação é a descoberta de registros correspondentes em uma ou mais fontes de dados.

Nos últimos anos, várias abordagens para deduplicação de registros foram propostas [Dorneles et al. 2007, Carvalho et al. 2006, Bilenko and Mooney 2003, Tejada et al. 2001]. A maioria dos trabalhos é aplicada à integração de dados relacionais e à realização de consultas por similaridade. Poucas abordagens foram desenvolvidas no contexto das bibliotecas digitais. Objetos digitais têm como principais atributos os metadados que descrevem a autoria do objeto. Podem existir mais de um objeto de mesmo título com autoria diferente. Por exemplo, tanto Brioniaccyr Feverstein quanto Rick Greenwald e David Kreines publicaram livros intitulados "Oracle in a Nutshell". Outro problema comum é a variação na representação dos nomes de autores em citações bibliográficas. Alguns exemplos destas variações são abreviações, inversões de nomes, grafias diferentes, e omissão de sufixos como Júnior [Oliveira et al. 2005]. Técnicas de deduplicação aplicadas ao contexto das bibliotecas digitais devem valorizar os atributos que se referem aos nomes dos autores para identificar corretamente metadados duplicados. A grande maioria das abordagens estudadas não trata especificamente da similaridade de nomes próprios.

[1] http://www.lbd.dcc.ufmg.br/bdbcomp
[2] http://www.informatik.uni-trier.de/~ley/db
[3] http://ieeexplore.ieee.org

Este artigo apresenta uma abordagem para identificar metadados duplicados em bibliotecas digitais distintas. A deduplicação é realizada através do mapeamento entre os rótulos dos metadados em diferentes formatos e de funções de similaridade aplicadas sobre o conteúdo dos metadados. As principais contribuições do trabalho são a melhora na qualidade da identificação de metadados de objetos digitais duplicados e a especificação das funções de similaridade para o domínio das bibliotecas digitais[4]. A qualidade da deduplicação é avaliada através de uma série de experimentos, os quais comparam a técnica proposta com outras três abordagens estudadas. Os resultados dos experimentos mostram que as funções propostas melhoram a qualidade da deduplicação de metadados de 0,64 a 31,5%. Como contribuição, este artigo especifica as seguintes funções:

- *YearSim* - compara os anos de publicação de dois objetos digitais;
- *IniSim* - identifica variações na representação do nome de um autor;
- *NameMatch* - identifica diferentes representações da autoria de um objeto digital;
- *MetadataMatch* - realiza o casamento de dois conjuntos de metadados que referenciam objetos digitais, combinando os escores gerados pelas funções *YearSim*, *IniSim* e *NameMatch*.

O restante do texto está organizado da seguinte forma. A Seção 2 apresenta trabalhos relacionados ao tema deduplicação de registros. A Seção 3 especifica detalhadamente a abordagem proposta para deduplicar metadados de objetos digitais. É definido um conjunto de funções e algoritmos que especificam a abordagem de identificação de duplicatas. A Seção 4 apresenta uma série de experimentos onde a qualidade da deduplicação de objetos digitais realizada pela abordagem proposta é avaliada em relação a trabalhos relacionados. Por fim, na Seção 5 são apresentadas as conclusões e possibilidades de trabalhos futuros.

2. Trabalhos Relacionados

Chaudhuri et al. [Chaudhuri et al. 2003] tratam os registros como vetores de palavras. É proposta uma função de similaridade que considera os pesos das palavras utilizando o método *Inverse Document Frequency* (IDF) [Baeza-Yates and Ribeiro-Neto 1999]. Carvalho e Silva [Carvalho and da Silva 2003] também usam o modelo vetorial para calcular a similaridade entre objetos de múltiplas fontes. A abordagem pode ser utilizada para deduplicar objetos com estruturas complexas como, por exemplo, documentos XML.

Dorneles et al. [Dorneles et al. 2004] propõem uma série de métricas de similaridade que manipulam coleções de valores que ocorrem nos documentos XML. São propostos dois tipos de métricas: métricas para valores atômicos (MAV); e métricas para valores complexos (MCV). As MAV são dependentes do domínio de aplicação. As MCV são definidas de acordo com as características dos nodos filho. Para conjuntos determinados por valores sem ordem definida é proposta a *MCV Set*. Dorneles et al. [Dorneles et al. 2007] estendem o trabalho anterior de forma que ao invés de definir um limiar de similaridade em termos dos escores retornados por uma função de similaridade, o usuário possa especificar a precisão esperada do processo de casamento de registros. A abordagem realiza o mapeamento entre os escores de similaridade e os valores de precisão através de um conjunto de treinamento.

[4]As definições iniciais das funções *IniSim* e *NameMatch* foram apresentadas previamente [Borges and Galante 2007] e utilizadas com o objetivo de identificar versões de objetos XML.

Outras propostas utilizam técnicas de aprendizado de máquina para determinar a similaridade entre registros. O sistema Active Atlas [Tejada et al. 2001] efetua o mapeamento entre objetos a fim de integrar fontes de dados. Regras de mapeamento entre os atributos são especificadas a partir de um processo de treinamento que utiliza árvores de decisão [Quinlan 1986]. A similaridade textual de diferentes registros é explorada por Cohen e Richman [Cohen and Richman 2002] propondo uma técnica escalável e adaptativa para agrupar esses objetos. O sistema MARLIN [Bilenko and Mooney 2003] descreve um *framework* para identificação de registros duplicados que utiliza métricas de similaridade textual adaptativas aplicadas a cada atributo (de acordo com o domínio de valores). São definidas duas métricas de similaridade: uma baseada na distância de edição e outra no algoritmo *Support Vector Machine* (SVM) [Boser et al. 1992].

Recentemente, foi proposta uma abordagem baseada na programação genética para deduplicar objetos digitais [Carvalho et al. 2006] que gera automaticamente funções de similaridade que identificam registros duplicados em um dado repositório. Os experimentos realizados sobre informações de autores e citações de artigos provenientes de bibliotecas digitais mostram que a abordagem proposta produz melhores resultados que os métodos propostos por Fellegi e Sunter [Fellegi and Sunter 1969]. A deduplicação de citações bibliográficas também é abordada por Lawrence et al. [Lawrence et al. 1999], onde são propostos algoritmos de casamento de citações baseados na distância de edição [Levenshtein 1966] e no casamento de palavras e frases.

Para deduplicar objetos digitais corretamente, é muito importante identificar ambigüidades na autoria de objetos digitais. A maioria dos trabalhos apresentados nesta seção, apesar de apresentarem soluções que deduplicam registros corretamente, não tratam especificamente da deduplicação de nomes próprios.

Identificar variações nos nomes de autores presentes em citações bibliográficas pode ser considerado um subproblema da deduplicação. Algumas funções foram propostas especificamente para comparar nomes próprios. *Guth* [Guth 1976] suporta pequenas variações de grafia. *Acronyms* [Lima 2002] suporta abreviações de nomes, mas possui uma limitação quanto a presença de letras maiúsculas que não sejam iniciais. *Fragments* [Oliveira et al. 2005] também suporta abreviações. Somente *Fragments* possui suporte a inversões de nomes. A solução adotada considera inversões de quaisquer nomes intermediários (nomes do meio), mas inversões do último nome só são detectadas com a presença da vírgula como caractere indicador da inversão. Além disso, a complexidade do algoritmo é quadrática tanto em função do número quanto do tamanho dos fragmentos. Todas as funções estudadas são independentes de idioma de origem dos nomes próprios. A Tabela 1 resume as características de cada função.

Tabela 1. Comparação entre as funções de nomes próprios.

Características	Guth	Acronyms	Fragments
suporte à variações de grafia	limitado	V	V
suporte à abreviações	X	limitado	V
suporte à inversões	X	X	parcial
complexidade	linear	linear	quadrática
independência de idioma	V	V	V

Legenda: V sim X não

3. Deduplicação de Metadados

Esta seção define a abordagem para deduplicação de metadados de objetos digitais. São considerados duplicatas ou réplicas dois ou mais conjuntos de metadados que possuem equivalência semântica, ou seja, que descrevem a mesma publicação (objeto digital) indexada por diferentes bibliotecas digitais. A estrutura dos metadados é comparada através do casamento entre os respectivos esquemas [Rahm and Bernstein 2001]. Entretanto, é assumido que o mapeamento entre os metadados oriundos de diferentes fontes é realizado previamente. O conteúdo dos metadados é comparado utilizando funções de similaridade propostas para o domínio de valores de cada metadado. São especificadas quatro funções de similaridade denominadas *YearSim*, *IniSim*, *NameMatch* e *MetadataMatch*. Estas funções comparam o conteúdo dos principais metadados que descrevem objetos digitais bibliográficos.

3.1. YearSim

A função de similaridade $YearSim : \{\mathbb{N}\} \rightarrow \mathbb{N}_S$, sendo $\mathbb{N}$ o conjunto dos números naturais e $\mathbb{N}_S = \{x \in \mathbb{N} \mid x = 0 \vee x = 1\}$, compara os anos em que dois objetos digitais foram publicados. A Equação 1 define a função,

$$YearSim(year_1\ year_2\ t_Y) = \begin{cases} 1 & \text{se } |year_1 - year_2| \leq t_Y \\ 0 & \text{caso contrário} \end{cases} \quad (1)$$

onde $year_i \in \mathbb{N}$ é o ano de publicação de um objeto digital e $t_Y \in \mathbb{N}$ representa a máxima diferença entre os anos de publicação.

Se o valor absoluto da diferença entre os anos de publicação dos objetos for menor ou igual ao limiar t_Y, a função retorna o escore de similaridade $s \in \mathbb{N}_S | s = 1$. Caso contrário, é retornado o escore $s = 0$. Por exemplo, ***YearSim(2005, 2004, 1) = 1*** e ***YearSim(1995, 1999, 3) = 0***.

3.2. IniSim

A função de similaridade *IniSim* identifica variações na representação do nome de um autor considerando grafias diferentes, inversões, abreviações e omissões de nomes. São realizadas comparações somente entre as iniciais dos nomes de autores, o que torna possível a implementação da função através de um algoritmo de complexidade linear.

$IniSim : \{C\} \rightarrow \mathbb{N}_S$, sendo C o conjunto composto por qualquer cadeia de caracteres, $\mathbb{N}$ o conjunto dos números naturais e $\mathbb{N}_S = \{x \in \mathbb{N} \mid x = 0 \vee x = 1\}$, é uma função que calcula a similaridade entre as iniciais dos nomes de autores. A função *IniSim* verifica se as iniciais $(a\ b) \in C$ dos nomes próprios representam o nome de um mesmo autor. São realizadas comparações entre a primeira, a segunda e a última inicial. As condições impostas pela função partem do princípio que devido às inversões, o primeiro e o último nome de um autor podem aparecer na primeira, segunda ou última inicial. Quando as iniciais correspondem ao mesmo autor, ou seja, quando as duas representações podem expressar a mesma entidade do mundo real, a função *IniSim* retorna um escore de similaridade $s \in \mathbb{N}_S | s = 1$. Caso contrário, é retornado o escore $s = 0$. *IniSim* é definida pela Equação 2,

$$IniSim(a\ b) = \begin{cases} 1 & se \quad \begin{array}{l}(a_1 = b_1 \wedge a_m = b_n) \vee \\ (a_1 = b_2 \wedge a_m = b_1) \vee \\ (a_1 = b_1 \wedge a_2 = b_2) \vee \\ (a_1 = b_n \wedge a_2 = b_1)\end{array} \\ 0 & \text{caso contrário} \end{cases} \quad (2)$$

onde: $a\ b$ são as iniciais dos nomes de autores; a_i é a i-ésima letra da palavra a, ou seja a i-ésima inicial do nome; b_i é a i-ésima letra da palavra b, ou seja a i-ésima inicial do nome; m é o tamanho da palavra a; n é o tamanho da palavra b.

Por exemplo, considere que a função *Initials* extrai as iniciais do nome de um autor. *IniSim(Initials(Eduardo Nunes Borges), Initials(Borges, Eduardo)) = IniSim (ENB, BE) = 1* ($a_1 = b_2 \wedge a_m = b_1$) e *IniSim(Initials(Eduardo Nunes), Initials(Borges, Eduardo)) = IniSim (EN, BE) = 0*.

3.3. NameMatch

NameMatch é uma função de similaridade que compara todos os autores de um objeto digital com os autores de outro objeto. O objetivo da função é identificar diferentes representações da autoria de um objeto digital.

$NameMatch : \{C\ \mathbb{R}_S\} \rightarrow \mathbb{N}_S$, sendo C o conjunto composto por qualquer cadeia de caracteres, $\mathbb{R}$ o conjunto dos números reais, $\mathbb{R}_S = \{x \in \mathbb{R} \mid 0 \leq x \leq 1\}$, $\mathbb{N}$ o conjunto dos números naturais e $\mathbb{N}_S = \{x \in \mathbb{N} \mid x = 0 \vee x = 1\}$, é definida pelo Algoritmo

```
NAMEMATCH(K L t_N)
 1  m ← LENGTH(K);
 2  n ← LENGTH(L);
 3  for i ← 1 to m
 4        do for j ← 1 to n
 5              do if INISIM(K_i Lj) = 1
 6                    then counter ← counter + 1;
 7                         L_j ← Null;
 8  if counter/MAX(m n) < t_N
 9    then return 0;
10    else return 1;
```

onde: $K\ L$ são listas de iniciais dos nomes dos autores dos objetos digitais comparados; $K_i \in C$ é o i-ésimo elemento da lista K, ou seja as inicias no i-ésimo autor do primeiro objeto; $L_j \in C$ é o j-ésimo elemento da lista L, ou seja as inicias no j-ésimo autor do segundo objeto; $t_N \in \mathbb{R}_S$ é um valor de limiar de similaridade aplicado a função; LENGTH é uma função que retorna o tamanho de uma lista; m é o tamanho da lista K; n é o tamanho da lista L; INISIM (Seção 3.2) compara as iniciais dos nomes dos autores; $counter$ acumula o número de casamentos encontrados pela função *IniSim*; MAX é uma função que retorna o tamanho da maior lista de palavras.

O algoritmo recebe como parâmetros duas listas compostas pelas iniciais dos nomes dos autores $(K\ L)$ e o limiar mínimo de casamento entre entre os autores t_N. As duas listas $K\ L$ são percorridas (linhas 3-4) a fim de encontrar o casamento entre os nomes dos autores, o qual é realizado através da função *IniSim*. Quando ocorre o casamento entre dois autores (linha 5), é atribuído um valor nulo a uma palavra da lista L para que esta não seja utilizada em futuras comparações (linha 7). A variável *counter* conta o número de casamentos verificados (linha 6). Quando o limiar de casamento mínimo é atingido (linha 8), a função *NameMatch* retorna um escore de similaridade $s \in \mathbb{N}_S | s = 1$, caso contrário retorna $s = 0$.

Quando ocorrem erros no processo automático de aquisição dos dados pelas bibliotecas digitais, ou seja, quando a lista de autores de um determinada publicação não está completa, o limiar t_N passado como parâmetro ajusta a função para que ocorra a identificação da duplicação. Por exemplo, considere a omissão do nome de um autor no segundo parâmetro da função *NameMatch ((AB, CD, EF), (FE, BA), 0.6)*. São atribuídos os valores 3 e 2 às variáveis m e n respectivamente. A lista *(AB, CD, EF)* é percorrida e cada elemento é comparado a todos os elementos da lista (FE, BA). A função *IniSim* retorna 1 para os parâmetros (AB, BA) e (EF, FE), sendo acumulado o valor 2 na variável *counter*. Como a razão entre o número de casamentos encontrados (2) e o retorno da função *Max* (3) é igual a 66,6%, o limiar mínimo de 60% passado como parâmetro é atingido e a função *NameMatch* retorna 1. O limiar de similaridade adotado, neste caso, faz com que dois conjuntos de metadados com diferentes números de autores sejam possíveis candidatos ao casamento[5].

Quando ocorrem erros nas iniciais dos nomes de autores, a função *IniSim* pode não identificar corretamente as varições dos nomes. O limiar t_N da função *NameMatch* também evita este tipo de erro, pois não é provável que existam erros nas iniciais de vários autores de um objeto digital. Os resultados empíricos demonstram a efetividade dessas heurísticas adotadas na especificação das funções *IniSim* e *NameMatch*.

3.4. MetadataMatch

A função *MetadataMatch* realiza o casamento de dois conjuntos de metadados que referenciam objetos digitais e tem como objetivo deduplicar metadados provenientes de bibliotecas digitais distintas.

A função $MetadataMatch : \{M\ \mathbb{N}_S\ \mathbb{R}_S\} \rightarrow \mathbb{N}_S$, sendo M o conjunto de metadados de um objeto digital, $\mathbb{R}$ o conjunto dos números reais, $\mathbb{R}_S = \{x \in \mathbb{R} \mid 0 \leq x \leq 1\}$, $\mathbb{N}$ o conjunto dos números naturais e $\mathbb{N}_S = \{x \in \mathbb{N} \mid x = 0 \vee x = 1\}$, é definida pelo Algoritmo

[5]Definições adequadas de valores de limiar são discutidas na literatura [Stasiu et al. 2005] e não fazem parte do escopo do trabalho apresentado neste artigo.

```
METADATAMATCH(a b t_Y t_N t_L)
1  if YEARSIM(YEAR(a) YEAR(b) t_Y) = 1
2    then for all AUTHOR(a)
3              do ADD(iniList_a INITIALS(AUTHOR(a)));
4         for all AUTHOR(b)
5              do ADD(iniList_b INITIALS(AUTHOR(b)));
6         if NAMEMATCH(iniList_a iniList_b t_N) = 1
7           then if LEVENSHTEIN(TITLE(a) TITLE(b)) > t_L
8                  then return 1;
9  return 0;
```

onde: $(a\ b)$ são metadados de dois objetos digitais; $t_Y \in \mathbb{N}$ é a máxima diferença entre os anos de publicação; $t_N \in \mathbb{R}_S$ é o limiar mínimo de casamento entre os autores; $t_L \in \mathbb{R}_S$ é o limiar mínimo de similaridade entre os títulos das publicações; YEAR é uma função que retorna o ano de publicação de um objeto digital; YEARSIM (Seção 3.1) compara os anos de publicação; AUTHOR é uma função que retorna os autores de um objeto digital; INITIALS retorna as iniciais de um autor; $iniList_i$ é uma lista das iniciais dos nomes dos autores do objeto i; ADD é uma função que adiciona as iniciais de um autor a uma lista; NAMEMATCH (Seção 3.3) realiza o casamento entre os autores; TITLE é uma função que retorna o título de um objeto digital; LEVENSHTEIN [Levenshtein 1966] calcula a similaridade entre os títulos dos objetos.

O algoritmo *MetadataMatch* inicia checando se os anos de publicação dos objetos digitais são compatíveis (linha 1). Logo após, são extraídas as iniciais dos nomes dos autores (linhas 2-5) e adicionadas às listas $iniList_i$. É realizado o casamento entre os autores utilizando a função *NameMatch* (linha 6). Somente os objetos digitais que atenderem ao limiar de casamento têm seus títulos comparados pela função *Levenshtein*. O algoritmo retorna o escore $s \in \mathbb{N}_S | s = 1$ (linha 8) se as condições impostas pelas funções *YearSim*, *NameMatch* e *Levenshtein* forem atendidas. A qualquer momento, se uma das condições não for atendida, o algoritmo retorna o escore $s = 0$ (linha 9).

A função *NameMatch* proposta tende a obter a revocação máxima, ou seja, entre os casamentos de objetos digitais recuperados estarão todos os casamentos relevantes. Embora a função *IniSim*, utilizada internamente pela função *NameMatch*, possa confundir nomes de autores apresentando as mesmas iniciais, muito dificilmente um par de metadados terá vários autores com as mesmas inicias e com nomes diferentes. Os altos índices de precisão alcançados nos experimentos realizados demonstram este comportamento.

A precisão da deduplicação atinge valores ainda maiores quando as instâncias que satisfizerem as condições impostas pela função de similaridade *NameMatch* forem avaliadas pela função *Levenshtein*. O metadado que representa o título do objeto digital só será comparado para os pares de publicações em que a função *NameMatch* não retorne zero. Isto reduz o tempo de processamento no processo de deduplicação, pois o algoritmo proposto limita as comparações do restante dos metadados. Maiores detalhes dos algoritmos podem ser encontrados em [Borges 2008].

4. Avaliação Experimental

Foram realizados diversos experimentos para validar as funções e algoritmos propostos. Os experimentos têm como objetivo geral verificar a qualidade da deduplicação de metadados de objetos digitais. Os metadados utilizados nos experimentos são provenientes das bibliotecas digitais BDBComp e DBLP. As funções propostas são comparadas aos algoritmos estudados (*Guth*, *Acronyms*, *Fragments* e *MCV Set*) apresentados na Seção 2. Os resultados dos experimentos mostram que as funções propostas melhoram a qualidade da deduplicação de metadados de 0,64 a 31,5%.

Foram utilizados as seguintes métricas para avaliação dos resultados:

- precisão (*precision*) [Manning et al. 2008] - mensura a taxa de acerto da deduplicação. A precisão define, portanto, a porcentagem de pares de objetos digitais identificados corretamente em relação ao número de pares recuperados. A Equação 3 define a métrica precisão;

$$\text{precisão} = \frac{|\{\text{pares recuperados}\} \cap \{\text{pares relevantes}\}|}{|\{\text{pares recuperados}\}|} \tag{3}$$

- revocação (*recall*) [Manning et al. 2008] - mensura fração das respostas relevantes da deduplicação. A revocação define, portanto, a porcentagem de pares de objetos digitais identificados corretamente em relação ao número de objetos digitais duplicados existentes (relevantes). A Equação 4 define a métrica revocação;

$$\text{revocação} = \frac{|\{\text{pares recuperados}\} \cap \{\text{pares relevantes}\}|}{|\{\text{pares relevantes}\}|} \tag{4}$$

- medida F balanceada (*balanced F-measure*) [Manning et al. 2008] - mensura a relação entre a precisão e a revocação. É a média harmônica ponderada da precisão e da revocação, a qual atribui pesos iguais à precisão e revocação. A Equação 5 define a métrica medida F balanceada;

$$F = 2 \cdot \frac{\text{precisão} \cdot \text{revocação}}{\text{precisão} + \text{revocação}} \tag{5}$$

- teste T (*Student's t-test*) [Hull 1993] - verifica se o desempenho de dois algoritmos é estatisticamente significativo. O teste T avalia se as médias de duas distribuições normais de valores são estatisticamente diferentes, apresentando bons resultados mesmo quando as distribuições não são perfeitamente normais . Foi utilizado o limiar de significância estatística $\alpha = 0\ 05$. Quando o valor P bi-caudal calculado é menor que α, existe uma diferença significativa entre os desempenhos dos dois algoritmos analisados. O melhor desempenho é do algoritmo com a maior média de distribuição.

4.1. Base de Dados

Os metadados utilizados nos experimentos são provenientes das bibliotecas digitais BDBComp e DBLP. A BDBComp possui cerca de 4 mil referências para artigos científicos publicados no Brasil. Os metadados que descrevem os artigos científicos são disponibilizados através do protocolo OAI-PMH, no padrão Dublin Core. Já a DBLP conta com

mais de 800 mil referências para artigos científicos publicados em diversos países. Os metadados são disponibilizados no *web site* principal da biblioteca digital no formato XML. Também é fornecida uma DTD contendo um simples esquema.

Foram selecionados os metadados que descrevem os artigos científicos publicados em algumas conferências nacionais e internacionais. A Tabela 2 indica o número de objetos em cada biblioteca digital para cada conferência, bem como o intervalo de anos de publicação selecionado. A última linha da tabela apresenta o total de objetos digitais selecionados para o experimento. A escolha destas conferências é baseada no fato de que alguns pesquisadores brasileiros que publicam trabalhos nas conferências nacionais, estendem os artigos publicados e os submetem para conferências internacionais de mesma área do conhecimento ou áreas correlatas.

Tabela 2. Objetos digitais utilizados nos experimentos.

Conferência	Intervalo	BDBComp	DBLP	Área do Conhecimento	Abrangência
ERBD	2005	18		Bancos de Dados	Nacional
SBBD	2001-2005	**122**	**143**	Bancos de Dados	Nacional
WIDM	2001-2004		76	Bancos de Dados	Internacional
ER	2001-2004		214	Sistemas de Informação	Internacional
CAiSE	2001-2004		192	Sistemas de Informação	Internacional
SIBGRAPI	2001-2004	**242**	**274**	Computação Gráfica	Nacional
CGI	2001-2004		196	Computação Gráfica	Internacional
SVR	2001-2004	107		Realidade Virtual	Nacional
INTERACT	2003		215	Interação Humano-Computador	Internacional
SBIA	2002-2004	**93**	**93**	Inteligência Artificial	Nacional
Σ	2001-2005	582	1403		

As conferências SBBD, SBIA e SIBGRAPI são indexadas pelas duas bibliotecas digitais, logo existem metadados duplicados. Foram identificados por um usuário especialista 415 casamentos reais entre os metadados. Os experimentos realizados têm como objetivo específico deduplicar os 415 pares de metadados de objetos digitais duplicados destas três conferências.

4.2. Experimentos Realizados

Os experimentos variam os seguintes parâmetros:

- *Func. Nomes* - função de similaridade de nomes próprios, que pode variar entre *Guth*, *Acronyms*, *Fragments* e a função proposta *IniSim*;
- *Alg. Casamento* - algoritmo de casamento de autores, que pode variar entre *MCV Set* e o algoritmo proposto *NameMatch*;
- t_Y - diferença no ano de publicação, variando entre 0 e 1;
- t_L - limiar aplicado à função *Levenshtein* o qual opera os títulos dos objetos digitais, variando entre 0,5 e 0,7;
- t_F - limiar aplicado à função *Fragments*, variando entre 2 e 4;
- t_N - limiar aplicado ao algoritmo proposto *NameMatch*, variando entre 0,75 e 1,0.

Os resultados dos experimentos são sumarizados na Tabela 3, onde são apresentados o número de pares de metadados deduplicados, a precisão, a revocação e a medida F para cada consulta. São apresentados a melhor e a pior medida F para cada combinação

entre a função de similaridade de nomes próprios e o algoritmo de casamento de autores. As medidas de qualidade foram determinadas a partir dos 415 pares de metadados duplicados identificados pelo usuário especialista.

Tabela 3. Qualidade da técnica proposta frente aos trabalhos estudados.

	Func. Nomes	Alg. Casamento	t_Y	t_L	t_F	t_N	Pares	Precisão	Revocação	Medida F
1	Guth	MCV Set	0	0,5			180	99,44%	43,13%	60,17%
2	Guth	MCV Set	1	0,7			178	100,00%	42,89%	60,03%
3	Acronyms	MCV Set	0	0,7			264	100,00%	63,61%	77,76%
4	Acronyms	MCV Set	1	0,5			269	98,51%	63,86%	77,49%
5	Fragments	MCV Set	0	0,5	4		383	99,48%	91,81%	95,49%
6	Fragments	MCV Set	1	0,5	2		372	99,19%	88,92%	93,77%
7	IniSim	MCV Set	0	0,5			395	99,49%	94,70%	97,04%
8	IniSim	MCV Set	1	0,5			398	98,74%	94,70%	96,68%
9	Guth	NameMatch	0	0,5		0,75	207	99,03%	49,40%	65,92%
10	Guth	NameMatch	1	0,7		1,0	178	100,00%	42,89%	60,03%
11	Acronyms	NameMatch	0	0,5		0,75	290	99,31%	69,40%	81,70%
12	Acronyms	NameMatch	1	0,5		1,0	260	98,46%	61,69%	75,85%
13	Fragments	NameMatch	0	0,5	4	0,75	393	99,49%	94,22%	96,78%
14	Fragments	NameMatch	1	0,5	2	1,0	372	99,19%	88,92%	93,77%
15	**IniSim**	**NameMatch**	**0**	**0,5**		**0,75**	**398**	**99,50%**	**95,42%**	**97,42%**
16	IniSim	NameMatch	1	0,5		1,0	393	98,73%	93,49%	96,04%
17	MetadataMatch		1	0,7		0,75	393	100,00%	94,70%	97,28%
18	MetadataMatch		1	0,5		1,0	393	98,73%	93,49%	96,04%

O teste T foi realizado comparando a função proposta *IniSim* com as funções *Guth*, *Acronyms* e *Fragments*, utilizando tanto a *MCV Set* quanto o algoritmo proposto *NameMatch* para combinar os escores gerados pelas funções. Os resultados do experimento são sumarizados na Tabela 4, onde são apresentados os valores de P bi-caudal para 415 observações em cada caso de teste.

Tabela 4. Teste T (P bi-caudal): duas amostras em par para médias.

Alg. Casamento	IniSim x Guth	IniSim x Acronyms	IniSim x Fragments
MCV Set	3,64 E-67	5,33 E-33	0,016
NameMatch	2,12 E-57	2,43 E-31	0,025

4.3. Análise dos Resultados

Analisando os resultados apresentados na Tabela 3, todos os experimentos obtiveram precisão maior que 98,46%, ou seja, todos os algoritmos testados identificam corretamente objetos duplicados. Entretanto, as consultas que envolvem os algoritmos *Guth* e *Acronyms* obtiveram revocação menor que 70% (linhas 1-4; 9-12). Os experimentos que utilizam a função *Fragments* verificaram revocação entre 88,92 e 94,22% (linhas 5-6; 13-14) enquanto utilizando a função proposta *IniSim* retornaram valores entre 93,49 e 95,42% (linhas 7-8; 15-16).

A qualidade geral dos algoritmos pode ser analisada pela medida F. A função *Guth* obteve índices de medida F entre 60,03 e 65,92% (linhas 2;9). A função *Acronyms* apresentou melhores resultados que *Guth* com índices entre 77,49 e 81,70% (linhas 4;11). *Fragments* marcou entre 93,77 e 96,78% (linhas 6;13). A função *IniSim* obteve resultados de medida F variando entre 96,04 e 97,42% em todos os casos de teste (linhas 15-16). Portanto, a função proposta *IniSim* melhora a qualidade da deduplicação quando comparada a *Guth* em 31,5%, a *Acronyms* em 15,72% e a *Fragments* em 0,64%.

A função *IniSim* obtém os mesmos resultados que *MetadataMatch* quando adotados os mesmos limiares de similaridade. A consulta 15 obteve o melhor resultado pois não foram realizados experimentos com *MetadataMatch* utilizando a variação do ano de publicação igual a zero.

O algoritmo de casamento de autores proposto *NameMatch* melhorou a qualidade geral da deduplicação. Para todas as funções de similaridade de nomes próprios, a medida F máxima alcançada com *NameMatch* é sempre maior que a medida F alcançada com *MCV Set*, variando em 5,75% para *Guth*, 3,94% para *Acronyms*, 1,29% para *Fragments* e 0,38% para *IniSim*. Os resultados dos experimentos que utilizam o limiar $t_N = 0\ 75$ são melhores do que utilizando $t_N = 1\ 0$, demonstrando que a flexiblidade da função *NameMatch* em não ter que casar o numero exato de autores é util para deduplicar metadados de objetos digitais.

Apesar dos valores de medida F máxima serem muito próximos para as funções *Fragments* (96,78%) e *IniSim* (97,42%), a função proposta *IniSim* é implementada através de um algoritmo com complexidade linear em função do tamanho dos nomes enquanto a complexidade da função *Fragments* é quadrática tanto em função do número quanto do tamanho dos fragmentos comparados. Além disso, os experimentos evolvendo o teste T, apresentados na Tabela 4, comprovam que a função proposta *IniSim* possui desempenho estatisticamente superior em relação a *Guth*, *Acronyms* e *Fragments*, pois os valores de P bi-caudal calculados são menores que o limiar de significância estatística $\alpha = 0\ 05$. Este comportamento permanece mesmo quando substituído o algoritmo de casamento que combina os escores gerados pelas funções (*MCV Set* e *NameMatch*).

Analisando o experimento envolvendo a função proposta *IniSim* com melhor resultado (Tabela 3, linha 15), 19 pares de conjuntos de metadados relevantes não foram retornados. Estes pares de metadados apresentam os seguintes problemas:

- omissão dos sufixos Júnior (ou Jr.), Filho e Neto - a função *IniSim* considera a última inicial dos nomes dos autores. Portanto, "*Roberto Marcondes Cesar Junior*" e "*R. Cesar*", por exemplo, não são identificados como o mesmo autor. Este problema ocorreu em 7 dos 19 pares de objetos relevantes não retornados e pode ser parcialmente corrigido com a remoção dos sufixos;
- omissão do primeiro ou último nome - a função *IniSim* considera a primeira e última inicial dos nomes dos autores. Portanto, "*Gabriel P. Lopes*" e "*José Gabriel Pereira Lopes*", por exemplo, não são identificados como o mesmo autor. Este problema ocorreu em 4 dos 19 pares de objetos relevantes não retornados;
- divergência no número de autores - o algoritmo *NameMatch*, quando utiliza um limiar de 100%, não admite dois objetos digitais com número de autores diferentes. Mesmo utilizando um limiar de 75%, alguns pares de metadados diferem muito (mais de 25%) no número de autores. Este problema ocorreu em 5 dos 19 pares de metadados relevantes não retornados;
- inversão do penúltimo e último nomes - a função *IniSim* suporta inversões somente se o último nome aparece no lugar do primeiro. Por exemplo, "*Schubert R. Carvalho*" e "*Schubert Carvalho Ribeiro*" possuem o penúltimo e o último nomes invertidos, portanto a função falha. Este problema ocorreu em somente 1 dos 19 pares de metadados relevantes não retornados;

- omissão de palavras no título ou títulos diferentes - a função *Levenshtein* não identifica a similaridade entre os títulos corretamente. Por exemplo, "*3D Reconstruction of Tomographic Images of Coronal Loops based on Image Metamorphosis*" e "*Image Morphing Applied to 3D Reconstruction of Coronal Loops.*" diferem em várias palavras. Este problema ocorreu em 2 dos 19 pares de metadados relevantes não retornados.

Analisando o experimento envolvendo a função *Fragments* com melhor resultado (Tabela 3, linha 13), 24 pares de conjuntos de metadados relevantes não foram retornados. Os 24 pares incluem todos os 19 pares não retornados pela função *IniSim*. Além dos problemas previamente identificados, que ocorreram em 19 dos 24 pares, os seguintes problemas foram verificados:

- abreviação de nomes - a função *Fragments* utiliza a distância de edição para comparar os fragmentos. A função falha quando os fragmentos diferem mais do que o limiar utilizado. Por exemplo, "*Florentino Fdez-Riverola*" e "*Florentino Fernández Riverola*" não são identificados como o mesmo autor, pois a distância de edição entre "*Fdez*" e "*Fernández*" é 5. Os experimentos utilizaram limiares variando entre 2 e 4. Este problema ocorreu em apenas 1 dos 24 pares de objetos relevantes não retornados;
- inversão do último nome - a função Fragments suporta apenas inversão de nomes intermediários, ou seja, nomes do meio. Portanto, "*Hee Cho Zang*" e "*Zang Hee Cho*", por exemplo, não são identificados como o mesmo autor porque existe uma inversão no último nome. Este problema ocorre principalmente com nomes de autores de origem asiática e ocorreu em 4 dos 24 pares de objetos relevantes não retornados.

5. Conclusões

O presente trabalho apresenta uma abordagem efetiva e eficiente para deduplicação de metadados de objetos digitais. É proposta uma série de funções e algoritmos, que aplicados sobre os metadados provenientes de bibliotecas digitais distintas, identificam objetos digitais duplicados com alta precisão e revocação. São especificadas as funções de similaridade *YearSim*, *IniSim*, *NameMatch* e *MetadataMatch* as quais identificam múltiplas representações de metadados de objetos digitais.

Várias das abordagens apresentadas na Seção 2 são baseadas em métodos de aprendizado de máquina, os quais requerem conjuntos de treinamento que podem ser caros de se obter. A abordagem proposta e os algoritmos comparados não necessitam de treinamento.

Os resultados dos experimentos realizados indicam que o desempenho da função *IniSim* e do algoritmo *NameMatch* propostos na deduplicação de metadados de objetos digitais é estatisticamente superior aos desempenhos dos algoritmos comparados. *IniSim* melhorou a qualidade da deduplicação de 0,64 a 31,5%. *NameMatch* melhorou a qualidade da deduplicação de 0,38 a 5,75%. *IniSim* resolve satisfatoriamente o problema da abreviação de nomes e o suporte à inversões do último nome encontrados na função *Fragments*.

A principal contribuição do trabalho é a especificação de uma abordagem que melhora a qualidade da deduplicação de metadados de objetos digitais identificando cor-

retamente variações na representação de nomes de autores através de um algoritmo de complexidade linear. Embora tenha sido proposta no contexto de bibliotecas digitais, a abordagem pode ser utilizada para deduplicação de metadados de outros domínios nos quais a deduplicação de nomes próprios é essencial.

Como trabalhos futuros destacam-se novos experimentos sobre outras bases de dados com um número significativamente maior de instâncias e a comparação com outras abordagens de deduplicação de registros apresentadas na Seção 2.

6. Agradecimentos

Este trabalho foi parcialmente financiado pelo CNPq (processos 481516/2004-2 e 141977/2008-6).

Referências

Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *Modern Information Retrieval*. ACM Press/ Addison-Wesley.

Bilenko, M. and Mooney, R. J. (2003). Adaptive duplicate detection using learnable string similarity measures. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD, 9.*, pages 39–48. New York: ACM.

Borges, E. N. (2008). *MD-PROM: um Mecanismo para Deduplicação de Metadados e Rastreio da Proveniência*. Dissertação de Mestrado (Ciência da Computação), Instituto de Informática, UFRGS, Porto Alegre.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of Annual Workshop on Computational Learning Theory, COLT, 5.*, pages 144–152. New York: ACM.

Carvalho, J. C. P. and da Silva, A. S. (2003). Finding similar identities among objects from multiple web sources. In *Proceedings of ACM International Workshop on Web Information and Data Management, WIDM, 5.*, pages 90–93. New York: ACM.

Carvalho, M. G., Gonçalves, M. A., Laender, A. H. F., and da Silva, A. S. (2006). Learning to deduplicate. In *Proceedings of Joint Conference on Digital Libraries, JCDL, 6.*, pages 41–50. New York: ACM.

Chaudhuri, S., Ganjam, K., Ganti, V., and Motwani, R. (2003). Robust and efficient fuzzy match for online data cleaning. In *Proceedings of ACM SIGMOD International Conference on Management of Data, SIGMOD*, pages 313–324. New York: ACM.

Cohen, W. W. and Richman, J. (2002). Learning to match and cluster large high-dimensional data sets for data integration. In *Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD, 8.*, pages 475–480. New York: ACM.

DCMI (2008). Dcmi metadata terms. Disponível em: http://dublincore.org/documents/dcmi-terms. Acesso em: 25 jan. 2008.

DOI (2008). The digital object identifier system. Disponível em: http://www.doi.org. Acesso em: 15 jan. 2008.

Dorneles, C. F., Heuser, C. A., Lima, A. E. N., da Silva, A. S., and de Moura, E. S. (2004). Measuring similarity between collection of values. In *Proceedings of Annual ACM International Workshop on Web Information and Data Management, WIDM, 6.*, pages 56–63. New York: ACM.

Dorneles, C. F., Heuser, C. A., Orengo, V. M., da Silva, A. S., and de Moura, E. S. (2007). A strategy for allowing meaningful and comparable scores in approximate matching. In *Proceedings of ACM Conference on Information and Knowledge Management, CIKM, 16.*, pages 303–312. New York: ACM.

Fellegi, I. P. and Sunter, A. B. (1969). A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210.

Fox, E. A., Akscyn, R. M., Furuta, R. K., and Leggett, J. J. (1995). Digital libraries. *Commun. ACM*, 38(4):22–28.

Guth, G. J. A. (1976). Surname spellings and computerized record linkage. *Historical Methods Newsletter*, 10(1):10–19.

Hull, D. (1993). Using statistical testing in the evaluation of retrieval experiments. In *Proceedings of Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR, 16.*, pages 329–338. New York: ACM.

Lawrence, S., Giles, C. L., and Bollacker, K. (1999). Digital libraries and autonomous citation indexing. *Computer*, 32(6):67–71.

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady.*, 10(8):707–710.

Lima, A. E. N. (2002). *Pesquisa de similaridade em XML*. Monografia de Graduação (Ciência da Computação), Instituto de Informática, UFRGS, Porto Alegre.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.

Oliveira, J. W. A., Laender, A. H. F., and Gonçalves, M. A. (2005). Remoção de ambigüidades na identificação de autoria de objetos bibliográficos. In *Proceedings of Simpósio Brasileiro de Bancos de Dados, SBBD, 19.*, pages 205–219.

Quinlan, J. R. (1986). Induction of decision trees. *Mach. Learn.*, 1(1):81–106.

Rahm, E. and Bernstein, P. A. (2001). A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350.

Stasiu, R. K., Heuser, C. A., and da Silva, R. (2005). Estimating recall and precision for vague queries in databases. In *Proceedings of International Conference on Advanced Information Systems Engineering, CAiSE, 17.*, pages 187–200.

Tejada, S., Knoblock, C. A., and Minton, S. (2001). Learning object identification rules for information integration. *Inf. Syst.*, 26(8):607–633.