Jean Wanderlei Alves de Oliveira

Uma Estratégia para Remoção de Ambiguidades na Identificação de Autoria de Objetos Bibliográficos

Dissertação apresentada ao Programa de Pós Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais, como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Belo Horizonte
1 de Abril de 2005

Agradecimentos

A Deus, por me inspirar em todos os momentos.

Ao professor Alberto Laender, pela orientação indefectível, pela paciência e compreensão.

Ao professor Marcos Gonçalves, pelas grandes idéias compartilhadas.

À Hui Han, pesquisadora do Department of Computer Science and Engineering da Pennsylvania State University, pela cessão da coleção de testes da DBLP usada em nossos experimentos.

Aos professores Marcelo Bax e Osvaldo Carvalho, pela participação em minha banca.

Aos amigos da UFMG, que foram uma alegria à parte nesta caminhada.

À minha esposa e companheira, Marina, que acreditou neste sonho, mesmo quando não mais parecia ser possível.

À minha família, por tudo.

À CAPES, e UFMG, sem as quais o sonho jamais poderia acontecer.

E a todos aqueles que sempre estiveram presentes em minha vida.

Dedicatória

Ao meu sogro e inesquecível amigo, Elias.

Ao meu avô Horácio, de quem sempre me lembrarei como meu pai.

Resumo

O problema de sobrecarga informacional gerado pelo sucesso da Web provocou o surgimento de serviços que reúnem informações em contextos específicos, conhecidos como bibliotecas digitais. Bibliotecas digitais reúnem informações digitais e metadados que freqüentemente são obtidos a partir de fontes diversas. A não padronização dos metadados oriundos dessas fontes traz como consequência a ambiguidade em determinados campos. Nesta dissertação apresentamos uma estratégia para o tratamento de ambiguidades encontradas em campos referentes a nomes de autores em bibliotecas digitais. Nossa estratégia utiliza técnicas de recuperação de informação associadas a um algoritmo de agrupamento que permite a criação de índices unificados. Demonstramos a eficácia de nossa estratégia através da realização de experimentos sobre duas coleções de teste derivadas da Biblioteca Digital Brasileira de Computação (BDBComp) e Digital Bibliography of Library Project (DBLP). Para a coleção da BDBComp, a média entre a qualidade dos grupos gerados e sua fragmentação foi superior à marca de 95%, e para a coleção da DBLP, essa média foi superior a 65%.

Abstract

The problem of informational overload generated by the success of the Web has led to the emergence of services that congregate information in specific contexts, known as digital libraries. Digital libraries combine digital information and metadata that frequently are collected from diverse sources. The lack of standardization of metadata deriving from these sources brings as consequence the ambiguity in determined fields. In this dissertation we present a strategy for the disambiguation in fields referring to names of authors in digital libraries. Our strategy uses Information Retrieval techniques associated to a clustering algorithm that allows the creation of unified indexes. We demonstrate the effectiveness of our strategy through a set of experiments conducted on two test collections derived from the Biblioteca Digital Brasileira de Computação (BDBComp) and the Digital Bibliography of Library Project (DBLP). For the BDBComp collection, the average between the quality of the generated groups and its fragmentation was over the mark of 95%, and for the collection of the DBLP, this average was over 65%.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

O advento da Web e seu posterior sucesso produziram um problema de "sobrecarga informacional" ou de explosão da informação disponível. Como qualquer pessoa pode publicar livremente o que quiser e, potencialmente, atingir milhões de usuários, o volume de informação acessível cresceu absurdamente. O resultado é que não conseguimos absorver toda a informação disponível e temos grandes dificuldades em encontrar a informação de interesse no momento em que precisamos.

Para localizar informação de interesse, os usuários utilizam fundamentalmente as máquinas de busca genéricas na Web. O processo de pesquisa a partir desses mecanismos é, na maioria das vezes, tedioso e frustrante, visto que, em muitos casos, o usuário deve formular diversas consultas até encontrar o resultado almejado e, em alguns outros, acaba por desistir.

Adicionalmente, o modo como as máquinas de busca indexam documentos em geral não permite que informações contidas na Web oculta[1] (informações contidas em bancos de dados disponibilizadas dinamicamente de acordo com a interação do usuário) sejam indexadas [Bergman, 2001]. Isso dificulta a tarefa do usuário, levando-o a procurar por fontes alternativas de informação às máquinas de busca. Esse processo impede, na prática, que o usuário desfrute totalmente da riqueza de dados contida na *Web*.

Uma solução desejável e possível é um tipo de serviço central que reúna informação proveniente de várias fontes de dados distintas, porém relativas a uma mesma área de interesse, e a torne disponível segundo um esquema de dados pré-acordado. Esse tipo de serviço central é o que chamamos genericamente de uma *biblioteca digital.*

[1] Tradução do termo *Hidden Web* (ou *Deep Web*).

Bibliotecas digitais podem ser vistas como bibliotecas convencionais que incorporam conteúdo interdisciplinar envolvendo informações digitais e suas formas de definição, aquisição, organização, gerenciamento e disseminação através de redes de comunicação globais [Montez, Pistori & Willrich, 2000]. Uma biblioteca digital não se caracteriza apenas por lidar com obras na forma digital, mas também por disponibilizar serviços e recursos que usualmente atuam sobre informações digitais.

As informações presentes em bibliotecas digitais normalmente são obtidas por alguns métodos típicos. O auto-arquivamento, conforme discutido por Café & Lage [2002], consiste na submissão de metadados e textos completos ao repositório pelos próprios pesquisadores. Segundo esse mecanismo, Silva [2004] propõe um serviço de auto-arquivamento de metadados para a Biblioteca Digital Brasileira de Computação, BDBComp [Laender, Gonçalves & Roberto, 2004]. Podemos também destacar esforços de diferentes comunidades para simplificar a aquisição de novas informações em bibliotecas digitais. A *Open Archives Initiative* (OAI)[2], iniciada em outubro de 1999, fornece alguma assistência em relação a este problema [Sompel & Lagoze, 2000; Lagoze & Sompel, 2001]. A abordagem OAI é baseada em, periodicamente, realizar a colheita (*harvesting*) de dados de diferentes fontes através de um protocolo simples e bem definido, denominado *Open Archives Initiative Protocol for Metadata Harvesting* [Sompel & Lagoze, 2000]. Os dados obtidos podem ser processados, integrados aos dados de outras origens e então carregados para o repositório da biblioteca digital.

Considerando o modo como tais dados são obtidos, um problema imediato vem à tona. As fontes fornecedoras nem sempre mantêm compromisso com a padronização da escrita de seus dados. Além disso, duas fontes distintas podem utilizar padronizações diferentes. Por exemplo, enquanto uma fonte pode armazenar nomes de autores na forma "Último_sobrenome, Primeiro_nome, Iniciais_dos_nomes_intermediários", outra fonte pode utilizar o padrão de escrita "Primeiro_nome, Iniciais_dos_nomes_intermediários, Último_sobrenome".

Ao realizarmos uma consulta em uma biblioteca digital, a conseqüência direta desse problema é a fragmentação das respostas obtidas. Para exemplificar, suponhamos que uma biblioteca digital faça referência a um autor *a* através das cadeias de caracteres c_1 e c_2

[2] http://www.openarchives.org

correspondentes a formas distintas de seu nome. Se no repositório R dessa biblioteca *a* estiver representado nos registros e_1 e e_2 por c_1 e nos registros e_3 e e_4 por c_2, então, ao realizarmos uma busca por *a*, receberemos com alguma sorte, as suas obras divididas em conjuntos correspondentes a c_1 e c_2. Por outro lado, autores homônimos podem ter seus conjuntos de obras associados. Se em um repositório o autor a_1, com trabalhos representados pelos registros e_1 e e_2, e o autor a_2, com trabalhos representados pelos registros e_3 e e_4 tiverem seus nomes escritos de forma idêntica através da cadeia de caracteres *c*, ao realizarmos uma busca por a_1, fatalmente receberemos como resposta o conjunto de obras formado por e_1, e_2, e_3 e e_4. Estes fatos ocorrem independentemente da padronização adotada, mas podem ser agravados caso os sobrenomes intermediários de um autor estejam abreviados ou não disponíveis. Um exemplo típico ocorre no CiteSeer[3]. Ao verificarmos nas estatísticas qual o autor mais citado em Ciência da Computação, recebemos como resposta o nome "D. Johnson", com 14928 citações[4]. Entretanto, podemos verificar com pouco esforço que esta cadeia de caracteres se refere a diversas pessoas do mundo real. Nomes de autores sofrem problemas de variação causados por abreviações (José S. A. Silva ou José Silva), apelidos (William ou Bill), permutações (Bin Liu ou Liu Bin), grafias diferentes (Osvaldo, Oswaldo), uso de letras maiúsculas e minúsculas (José Da Silva, José da Silva), hiferação (Ribeiro-Neto, Ribeiro Neto), composição de nomes (El-Masri, Elmasri), uso de prefixos e sufixos (Sr., Jr., números, etc.), além de que algumas pessoas mudam de nome ao se casarem [Ley, 2002].

Neste trabalho, apresentamos uma estratégia que auxilia a detecção de formas variantes de nomes próprios em bases de dados bibliográficas. Essa estratégia utiliza funções para casamento de padrões e técnicas de recuperação de informação, aproveitando informações adicionais, como, por exemplo, títulos de trabalhos, nomes de co-autores e veículos de publicação para remover ambigüidades existentes em campos bibliográficos que representam nomes de autores, podendo também ser aplicada a outros campos. Nossa estratégia não exige coleções de treinamento podendo, portanto, ser aplicada a repositórios sem que haja um esforço inicial para a geração dessas coleções. Por ser não interativa, dispensa também qualquer supervisão durante todo o processamento. O resultado da aplicação de nossa estratégia a um repositório é um tipo de arquivo de autoridade [Auld, 1982] a que chamamos

[3] http://citeseer.ist.psu.edu/
[4] http://citeseer.ist.psu.edu/mostcited.html, em 30/11/2004

índice unificado. Índices unificados têm a finalidade de manter a correspondência entre todas as formas permissíveis de cadeias de caracteres em um campo bibliográfico em particular, como, por exemplo, nomes de instituições, títulos de periódicos, nomes de conferências e, em nosso caso, nomes de autores. Com isso, reduzimos a fragmentação dos conjuntos de obras de cada autor e ainda criamos conjuntos mais confiáveis.

1.1. Descrição do Problema

O problema de criação de índices unificados pode ser definido como se segue. Considerando a representação XML [Abiteboul, Buneman & Suciu, 1999] de um repositório R, formado pelo conjunto $E = \{e_1, e_2, ..., e_n\}$ de elementos XML que representam objetos em um domínio específico, onde cada e_i por sua vez é formado por um conjunto de subelementos $S_i = \{s_{i1}, s_{i2}, ..., s_{im}\}$, a criação de índices unificados consiste em agrupar os elementos de E por seus subelementos s_{ij} de mesmo rótulo T, sendo que, para que esse agrupamento seja possível, deve haver pelo menos um s_{ij} de rótulo T em cada $e_i \in E$.

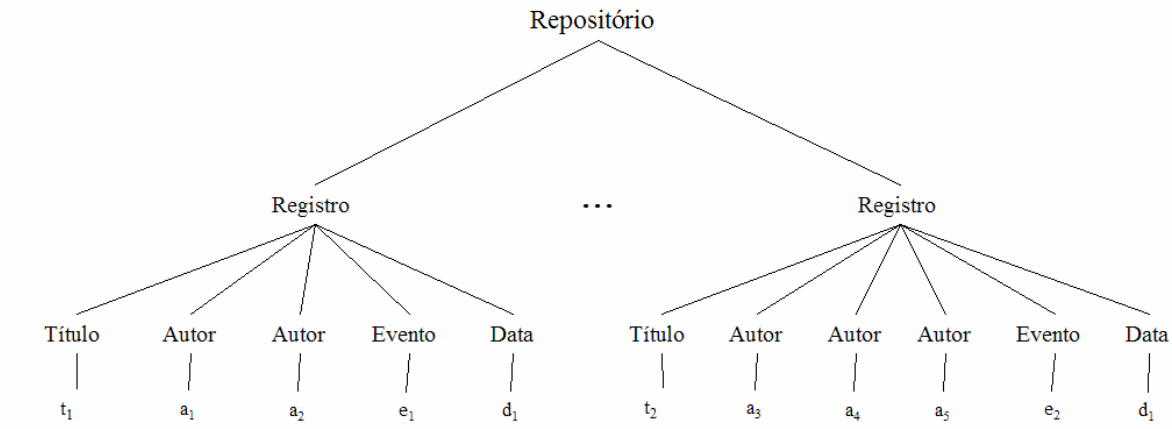

Figura 1.1. Exemplo de um Repositório de Metadados de Objetos Bibliográficos.

Para exemplificar, consideremos o repositório de metadados de objetos bibliográficos contido na Figura 1.1. No repositório, cada e_i corresponde a um elemento Registro. Para e_1, o conjunto de subelementos corresponde a $S_1 = \{t_1, a_1, a_2, e_1, d_1\}$ e os rótulos desses

subelementos são: {Título, Autor, Autor, Evento, Data}. Criar um índice unificado para esse repositório consiste em agrupar seus elementos Registro através de algum subelemento de mesmo rótulo, por exemplo, Evento, desde que em cada S_i haja pelo menos um subelemento com tal rótulo.

Desse modo, um índice unificado reúne em cada grupo todas as formas possíveis do subelemento de rótulo T escolhido para agrupamento. Em nosso caso, estamos particularmente interessados na criação de índices unificados para nomes de autores de objetos bibliográficos, o que implica diretamente na diminuição de problemas de ambigüidade como o citado anteriormente referente ao CiteSeer. Utilizamos para isso técnicas que consideram evidências adicionais como, por exemplo, títulos de trabalhos, nomes de co-autores e eventos, permitindo supor que dois nomes distintos em um repositório R se referem à mesma pessoa. A seguir apresentamos um resumo dos trabalhos mais relevantes que tratam problemas relacionados ao descrito.

1.2. Trabalhos Relacionados

Encontramos na literatura diversos trabalhos que tratam o problema da criação de índices unificados diretamente ou formas variantes desse problema. Han et al [2004] descrevem duas estratégias para criação de arquivos de autoridade para nomes de autores. Essas estratégias utilizam o modelo *Naive Bayes* [Han et al, 2005] ou *Support Vector Machines* [Cristianini & Shawe-Taylor, 2000] e têm por objetivo a remoção de ambigüidades em citações bibliográficas. Uma coleção de treinamento deve ser fornecida aos algoritmos para que estes gerem arquivos de autoridade com qualidade, o que não acontece em nossa estratégia. Como em nosso trabalho, são utilizados nomes de co-autores, títulos de trabalhos e nomes de veículos de publicação como fontes de evidência.

French, Powell & Schulman [2000] propõem um algoritmo de agrupamento para a criação de arquivos de autoridade em bibliotecas digitais. Nesse trabalho, o objetivo é criar arquivos de autoridade para os nomes das instituições às quais os autores pertencem. De forma semelhante ao nosso trabalho, são armazenadas as formas mais representativas para os nomes das instituições. Não são utilizadas fontes de evidência adicionais para a criação dos arquivos

de autoridade, o que é feito com base apenas nas cadeias de caracteres que representam os nomes das instituições. A revisão da saída do algoritmo por um especialista é considerada parte do processo de criação dos arquivos de autoridade. Esse trabalho se baseia apenas na estrutura sintática o que, de um modo geral, não resolve o problema para nomes de autores, pois, na prática, consideraria autores homônimos como sendo a mesma pessoa.

O controle de acesso é descrito por Snyman & Rensburg [2000] como uma forma de controle de autoridade em que não existe uma forma autoritária de escrita de nomes. Essas formas de escrita são identificadas através de um código chamado INSAN (*International Standard Author Number*), que é gerado a partir de dados dos autores como, por exemplo, o número de identidade. Por essa abordagem, é exigida a participação de um órgão que administre os códigos, como uma agência central. Os autores devem ser cadastrados no sistema, o que difere fundamentalmente de nossa estratégia, que trata do problema de forma automática. Essa abordagem exige um número considerável de informações e que, na maioria das vezes, não estão disponíveis em bibliotecas digitais.

O problema de detecção de objetos similares na integração de dados de diferentes fontes é tratado por Carvalho & Silva [2003], Cohen & Richman [2002] e Tejada, Knoblock & Minton [2001] e consiste em uma variante do problema de criação de arquivos de autoridade em que o objetivo não é determinar a correspondência entre entidades representadas pelos objetos, mas sim a correspondência entre os próprios objetos. Carvalho & Silva [2003] apresentam quatro estratégias para o cálculo da similaridade entre objetos de diferentes fontes. Em todas elas, a medida do cosseno [Baeza-Yates & Ribeiro-Neto, 1999; Chakrabarti, 2002] é aplicada sobre as fontes de evidência disponíveis em metadados XML. Esse trabalho, assim como o nosso, não utiliza coleções de treinamento para atingir seu objetivo de identificar objetos similares em diferentes fontes de dados. Além disso, a estratégia proposta não é particularizada a um contexto específico, podendo ser aplicada a objetos de domínios diferentes. Cohen & Richman [2002] propõem duas abordagens, agrupamento adaptativo e casamento adaptativo. No agrupamento adaptativo, a partir de um único conjunto de dados e de alguns exemplos, o sistema aprende a agrupar objetos que se referem à mesma entidade do mundo real. No casamento adaptativo, o sistema tenta identificar em um conjunto de dados que elementos de outro conjunto se referem à mesma entidade do mundo real. Tejada, Knoblock &

Minton [2001] extraem informações de páginas através de um sistema mediador de informações chamado Ariadne [Knoblock et al, 2001]. Através de um mecanismo de recuperação de informação que considera diversas transformações possíveis, entre elas, *stemming*, *soundex* e *acronym*, verificam por uma tabela *hash* a similaridade entre cada fonte de evidência, que é calculada através da medida do cosseno [Baeza-Yates & Ribeiro-Neto, 1999; Chakrabarti, 2002]. Após o cálculo de todas as similaridades, é realizada a etapa de aprendizagem de regras de mapeamento, que verifica quais são as fontes de evidência mais importantes a serem consideradas. Uma árvore de decisão [Quinlan, 1996] possibilita o processo de aprendizagem. Depois de criada, a árvore de decisão é convertida em regras de mapeamento. Nessa abordagem, o sistema realiza a solicitação de exemplos ao usuário, o que diminui a interação necessária.

Gu et al [2004] descrevem em seu trabalho as diversas etapas e possibilidades para o desenvolvimento de soluções para a remoção de ambigüidades, citando também algumas aplicações já desenvolvidas no meio acadêmico, governamental e comercial. Entre elas destacam-se o GDRIVER[5], desenvolvido pelo United States Bureau of Census e que se baseia na padronização de nomes e endereços através de uma análise sintática auxiliada por diversos arquivos de referência. Esses arquivos de referência trazem correspondências entre abreviaturas do tipo st para *street* e rd para *road*, para endereços e, assim como em nossa estratégia, outras correspondências entre nomes e apelidos ou abreviaturas. O Febrl, ou *Freely Extensible Biomedical Record Linkage* [Christen, Churches & Hegland, 2004], consiste em outro exemplo de software que realiza remoção de ambigüidades. Para isso, realiza uma padronização de dados através de técnicas de *machine learning* supervisionadas implementadas através de modelos markovianos ocultos (hidden Markov models) [Rabiner, 1989]. Por utilizar modelos markovianos ocultos, o Febrl necessita de dados de treinamento, o que não ocorre em nossa estratégia. O Febrl, como em nossa abordagem, também conta com métodos que possibilitam que comparações desnecessárias entre registros possam ser descartadas. Por último, o GRLS, ou *Generalized Record Linkage System* [Fair, 2004], criado pela Agência Nacional de Estatística do Canadá, é também um exemplo de software que realiza a remoção de ambigüidades. Para isso, se baseia no método de Fellegi-Sunter [Fellegi

[5] Documentação completa em http://nedinfo.nih.gov/docs/US%20Census%20Bureau%20Record%20Linkage%20SW%20User%20Documentation.pdf

& Sunter, 1969], agrupando registros em grupos considerados fracos ou fortes. O sistema disponibiliza uma interface gráfica ao usuário, permitindo que sejam criadas regras que contribuam para a remoção de ambigüidades.

Al-Kamha & Embley [2004], Bagga & Baldwin [1998] e Mann & Yarowsky [2003] tratam o agrupamento de documentos que fazem menção à mesma entidade do mundo real. Esse problema é uma variação do problema de criação de arquivos de autoridade em que o ponto de partida consiste de uma coleção de documentos, o que implica na necessidade de análise sintática. Al-Kamha & Embley [2004] agrupam os resultados de pesquisas por nomes de pessoas submetidas a mecanismos de busca. Nesse trabalho é utilizada uma abordagem multifacetada que considera *links* entre *sites*, similaridade entre páginas e atributos das pessoas, como número de telefone, endereço eletrônico, estado, cidade e código postal como aspectos relevantes que evidenciam que duas citações são para a mesma entidade. A estratégia proposta cria uma matriz de confiança e requer um conjunto de treinamento. Ao final, obtém-se o resultado das buscas agrupado por nomes de pessoas. Bagga & Baldwin [1998] apresentam uma abordagem para reconhecer entidades, por exemplo, pessoas, com nomes iguais em diferentes documentos. Como Al-Kamha & Embley [2004], duas entidades de nomes idênticos podem ser consideradas pessoas diferentes. Através de um extrator de sentenças, é criado um sumário a respeito da entidade de interesse. Com um módulo para remoção de ambigüidades e utilizando a medida do cosseno, são calculadas as similaridades entre os documentos. Sumários com similaridade acima de um determinado limite são considerados referentes à mesma entidade. Essa abordagem não necessita de coleções de treinamento. Mann & Yarowsky [2003] utilizam extratores para obter informações de páginas da Web. Algumas técnicas para extração de informações contribuem com dados mais precisos, como idade e data de aniversário, nacionalidade e ocupação. A cada documento é associado um vetor de características, ou palavras, extraídas automaticamente. Esse modelo utiliza um método de agrupamento aglomerativo em que, a cada etapa, os dois vetores mais similares são unidos, gerando um novo grupo. A operação é realizada amiúde até que todos os documentos estejam agrupados. Como em nossa abordagem, *stop words* também são eliminadas.

Monge & Elkan [1997] tratam o problema da detecção de tuplas duplicatas em bancos de dados, o que consiste em uma variação do problema de criação de arquivos de autoridade

em que o objetivo não é detectar entidades recorrentes, mas sim tuplas que se repetem. Para isso, é realizada uma ordenação no banco de dados através de uma chave e em seguida, assim como em nossa proposta, cada par de tuplas é comparado. Esse trabalho utiliza distância de edição para detectar registros duplicados em uma abordagem livre de contexto, calculando o membro mais representativo para cada grupo gerado. Diferente de nosso trabalho, Monge & Elkan [1997] trabalham diretamente em um banco de dados, enquanto nosso ponto de partida é um repositório XML.

Finalmente, Zhang, Gonçalves e Fox [2003] descrevem uma abordagem que tem por objetivo a extração automática de ETD's (*Electronic Theses and Dissertations*) da NDLTD (*Networked Digital Library of Theses and Dissertations*)[6] para integração à CITIDEL (*Computing and Information Technology Interactive Digital Educational Library*)[7]. Apesar de o objetivo desse trabalho ser a classificação de ETD's da NDLTD como relacionados ou não à área de computação, a técnica utilizada para tanto é semelhante à nossa e às dos demais trabalhos descritos. A estratégia apresentada é dividida em três passos principais. No primeiro passo, é utilizada a classificação baseada no conteúdo, que considera títulos, assunto e resumos como fontes de evidência. No segundo passo, a filtragem baseada nos nomes de colaboradores é utilizada para melhorar os resultados da classificação realizada no primeiro passo, sendo que um conjunto de nomes de autores extraídos da ACM DL[8] é usado como referência e, caso a maioria dos colaboradores de uma determinada ETD esteja nessa lista, o trabalho é considerado relacionado a computação. Por último, é aplicada a filtragem baseada no campo de assunto dos metadados, melhorando os resultados quando registros no segundo passo não contêm o campo de colaboradores. Esse trabalho utiliza *Support Vector Machines* e uma coleção de treinamento para especificar as classes a serem aprendidas, aplicando a medida do cosseno para calcular similaridades.

1.3. Contribuições

Apresentamos nesta dissertação uma estratégia que auxilia a detecção de formas variantes de nomes próprios em bases de dados bibliográficas. Seu mecanismo está baseado na

[6] http://www.ndltd.org
[7] http://www.citidel.org
[8] http://portal.acm.org/dl.cfm

criação de índices unificados através da utilização de técnicas de recuperação de informação e algoritmos de agrupamento. De acordo com essa estratégia, a criação de um índice unificado é dividida em duas fases: (1) a fase de pré-avaliação, que tem a finalidade de reconhecer, a partir de uma função para casamento de padrões, a presença em elementos XML que representam objetos bibliográficos, de nomes que potencialmente representam a mesma pessoa, e (2) a fase de cálculo de similaridades, que através de medidas usadas em recuperação de informação e utilizando informações adicionais, avalia se os candidatos identificados na fase de pré-avaliação efetivamente se referem à mesma pessoa.

Indicamos como principais contribuições do nosso trabalho as seguintes:

1. A proposta de uma função de casamento de padrões específica para comparação de nomes de pessoas, a qual denominamos comparação por fragmentos;
2. A proposta de um algoritmo de agrupamento para a criação de índices unificados para nomes de autores;
3. O estudo de métricas de qualidade para avaliação dos índices unificados gerados;
4. Uma experimentação extensa, avaliando duas coleções com características diferenciadas.

Nossa abordagem apresenta como vantagens o fato de não utilizar uma coleção de treinamento, não adotar uma estratégia supervisionada e ser facilmente adaptável às características do repositório utilizado e dos objetivos esperados. Por exemplo, como resultado de experimentos realizados sobre uma coleção de teste em que nomes de autores aparecem predominantemente completos, obtivemos média entre a qualidade dos grupos gerados e sua fragmentação superior à marca de 90%. Para uma coleção onde os nomes de autores aparecem em sua maioria abreviados, essa média é superior a 65%. Por outra vertente, através da utilização de fontes de evidência adicionais, é possível priorizar a geração de índices unificados que apresentem grupos com maior qualidade ou menor fragmentação.

1.4. Estrutura da Dissertação

Esta dissertação está organizada como segue. No Capítulo 2 discutimos nossa estratégia para a criação de índices unificados, fornecendo definições sobre seus principais componentes

e detalhando os algoritmos propostos. No Capítulo 3 apresentamos os resultados experimentais obtidos na criação de índices unificados para dois repositórios com características diferenciadas. Por último, no Capítulo 4, expomos nossas conclusões e definimos algumas direções para trabalhos futuros.

Capítulo 2

Estratégia Proposta

As bibliotecas digitais se destacam atualmente como ferramentas de grande utilidade para os mais diferentes usuários. Através delas se obtém com facilidade informações relativas a trabalhos publicados em um determinado evento ou por um autor qualquer. Ao buscarmos determinado evento em uma biblioteca digital, podemos receber como resposta um *link* para o *site* do evento, como na DBLP[9], ou uma página listando os artigos publicados neste evento, como na BDBComp[10].

Os dados retornados por consultas a bibliotecas digitais se encontram freqüentemente armazenadas em repositórios alimentados por fontes diversas. Não é comum qualquer tipo de padronização entre as fontes, que em muitas vezes são outras bibliotecas digitais. Essas informações são normalmente coletadas e convertidas para o formato do esquema seguido pelo repositório. No entanto, essas conversões não detectam variações e erros de grafia e não evitam certas ambigüidades. Uma busca por determinado autor em uma biblioteca digital traz à tona os problemas gerados pela falta de padronização das fontes. A Figura 2.1 ilustra uma pesquisa realizada na BDBComp por autores cujo primeiro nome é “Osvaldo”.

Observamos na figura a presença de dez *links* para páginas que supostamente listam as coleções de dez autores diferentes. Entretanto, podemos perceber com facilidade que os *links* “Osvaldo S. F. de Carvalho”, “Osvaldo Sérgio F. de Carvalho” e “Osvaldo Sérgio Fahrat de Carvalho” se referem a páginas com dados sobre os trabalhos de uma única pessoa. De forma idêntica, “Osvaldo Saavedra” e “Osvaldo Saavedra Mendez” são duas variações na grafia do

[9] http://dblp.uni-trier.de
[10] http://www.lbd.dcc.ufmg.br/bdbcomp

nome de uma única pessoa, o que nos faz concluir que existem ao todo apenas sete pessoas distintas.

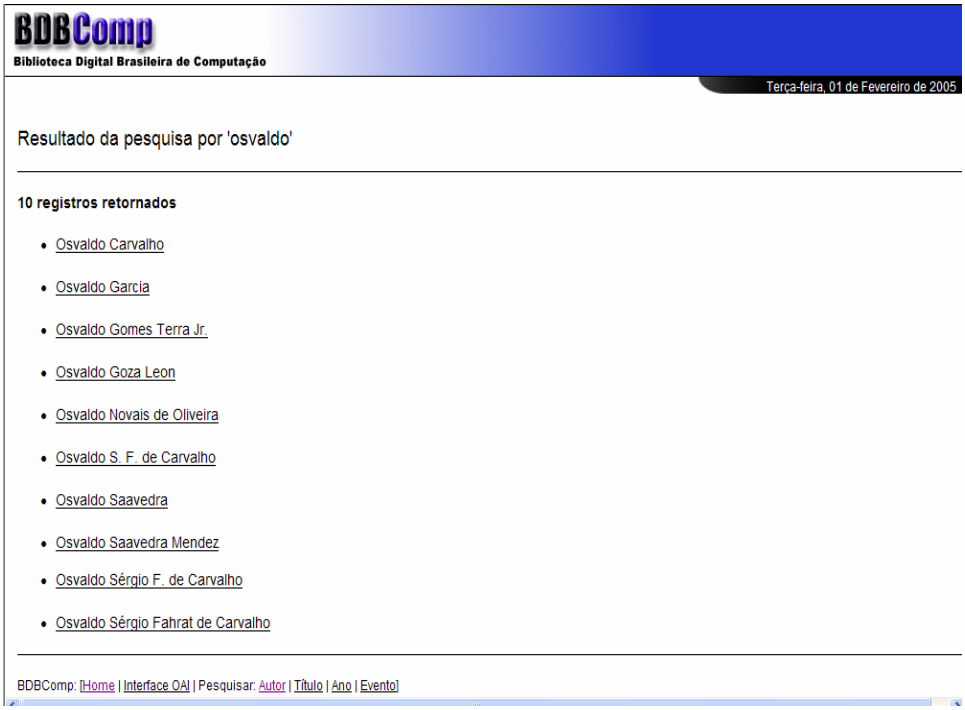

Figura 2.1. Resultado Retornado pela BDBComp para a consulta “osvaldo”.

A criação de índices unificados é uma solução para o problema acima. Índices unificados são inspirados em arquivos de autoridade que, como descreve [Auld, 1982], mantêm a correspondência entre todas as formas permissíveis de cadeias de caracteres para determinados atributos de objetos bibliográficos. Um índice unificado, além disso, pode também referenciar as entradas em um repositório que pertencem a um determinado autor. Desta forma, ao realizarmos uma busca, os problemas de variação e ambigüidade descritos anteriormente estariam solucionados ou, pelo menos, minimizados.

Apresentamos neste capítulo a estratégia desenvolvida para a criação de índices unificados para nomes de autores em bibliotecas digitais. A Figura 2.2 descreve a estrutura básica utilizada pela estratégia.

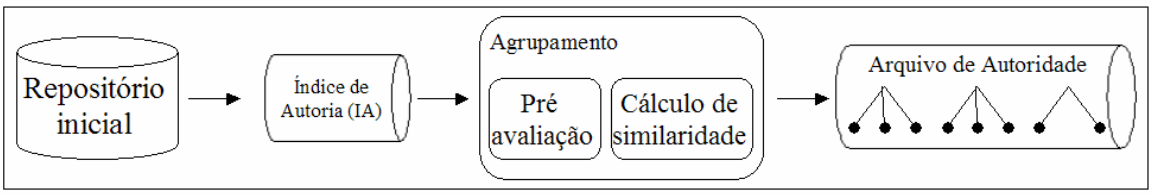

Figura 2.2. Estrutura Utilizada pela Estratégia Proposta.

Nossa estratégia parte de um repositório XML que contém metadados de objetos bibliográficos. Por objetos bibliográficos nos referimos a artigos publicados em eventos, livros, capítulos de livro, dissertações de mestrado, entre outros. Metadados deste domínio freqüentemente incluem o título do trabalho que representam, nomes de autores e do veículo de publicação. Realizamos uma análise sintática (*parsing*) e um processamento desses dados e criamos um índice de autoria, formado por uma lista de tuplas que contém uma entrada para cada nome de autor presente no repositório inicial. O índice de autoria (IA) corresponde à principal entrada do algoritmo de agrupamento, que é formado por dois módulos: o módulo de pré-avaliação, que tem a função de identificar quais são as entradas do índice de autoria que podem corresponder à mesma entidade do mundo real, e o módulo para cálculo de similaridades, que utiliza os dados disponíveis nas tuplas para, através de uma função de similaridade, estabelecer se dois nomes candidatos podem efetivamente ser considerados como referentes à mesma entidade do mundo real.

O algoritmo de agrupamento processa o índice de autoria gerando um índice unificado que embute um arquivo de autoridade. Cada entrada deste índice é composta pela forma mais completa para cada nome de autor e uma lista com as posições de seus trabalhos no repositório. Nosso algoritmo pode ser classificado como um método de agrupamento hierárquico [Berkhin, 2002; Chakrabarti, 2002; Frakes & Baeza-Yates, 1992] que, a cada passo, associa o par de entradas no índice de autoria com similaridade superior a um determinado limite mínimo e que ainda não faz parte do mesmo grupo.

2.1. Índice de Autoria

Devido à diversidade de fontes utilizadas, os dados presentes em um repositório tendem a ser comprometidos por inúmeras diferenças de grafia. Estas diferenças podem decorrer de um enumerado de fatores e variar de acordo com a semântica dos dados. Conforme já mencionado, nomes de autores sofrem problemas de variação causados por abreviações, grafias diferentes, uso de letras maiúsculas e minúsculas, hiferação, uso de prefixos e sufixos, etc. Eventos podem vir escritos por extenso ou por suas siglas e títulos de trabalhos podem aparecer em idiomas diferentes. Em função destes problemas, necessitamos de uma estrutura de dados normalizada, onde as cadeias de caracteres tenham sido pré-processadas.

O índice de autoria (IA) é uma estrutura de dados que reúne as informações contidas no repositório inicial em uma seqüência de tuplas. Para construirmos o índice de autoria de um repositório R como o da Figura 2.3, realizamos um processamento individualizado de cada elemento disponível. As definições abaixo descrevem os componentes da estrutura do repositório inicial.

Definição 1. *Um **alfabeto completo**, Σ, é um conjunto de caracteres formado por todas as letras (maiúsculas, minúsculas e acentuadas), dígitos e símbolos especiais. Um **alfabeto simplificado**, Σ', é um conjunto formado apenas por letras minúsculas e dígitos.*

Definição 2. *Um **fragmento** é uma seqüência finita de caracteres extraídos de Σ. Uma **cadeia de caracteres** é uma seqüência finita de fragmentos separados por um ou mais espaços em branco.*

Definição 3. *Uma **cadeia de caracteres normalizada**, c, é uma cadeia de caracteres formada apenas por elementos de Σ'. Referimos a cada fragmento de c pela notação $c[i]$, com $1 \leq i \leq n$, sendo n o número total de fragmentos de c.*

Como podemos observar na Figura 2.3, R é composto por três elementos <registro>. Esses elementos por sua vez são formados pelos elementos <título>, <autor>, <evento> e

<data>, que são cadeias de caracteres derivadas de Σ. Todos os elementos <registro> têm uma posição dentro de R e todos os elementos <autor> têm uma posição dentro de cada elemento <registro>. Como nosso objetivo é a criação de um arquivo de autoridade para nomes de autores, criamos uma tupla para cada elemento <autor> constante em R. A seguir apresentamos a definição de Índice de Autoria.

Definição 4. *Um* **índice de autoria** *é formado por uma lista ordenada de tuplas, T,. Cada tupla t_i em T segue a estrutura <código$_i$, autor$_i$, co-autores$_i$, informações_adicionais$_i$>, para $1 \leq i \leq |A_r|$, onde A_r é o conjunto de elementos <autor> de R. Na estrutura, código$_i$ é formado por um número inteiro que expressa a posição seqüencial do elemento <registro> dentro do repositório separado por um hífen da posição do elemento <autor> no elemento <registro>, iniciada da posição 0; autor$_i$ é a cadeia de caracteres normalizada que representa o nome de um autor em R; co-autores$_i$ corresponde às demais cadeias de caracteres normalizadas de nomes de autores contidos no elemento <registro> onde autor$_i$ aparece; finalmente, informações_adicionais$_i$ corresponde a todas as demais cadeias de caracteres normalizadas referentes a informações disponíveis em R.*

Para o repositório da Figura 2.3, temos como informações adicionais o título, o evento e a data de publicação dos trabalhos. Após a análise sintática, realizamos uma normalização das cadeias de caracteres. R. Yanilos [1997] e French, Powell & Schulman [2000] reportam normalizações semelhantes a essa, que consiste de seis passos, sendo que dois deles (passos 2 e 5) se aplicam apenas a cadeias de caracteres de nomes de autores:

1. Substituir letras acentuadas (á, ç, etc.) por suas versões não acentuadas (a, c, etc.). Por exemplo, “Jr., José G. Silva” se torna “Jr., Jose G. Silva”.
2. Em cadeias de caracteres correspondentes a nomes de pessoas, inverter a ordem dos nomes na ocorrência de vírgulas. Por exemplo, “Jr., Jose G. Silva” se torna “Jose G. Silva Jr.”
3. Substituir caracteres especiais (hífens, pontos, etc.) por espaços em branco. Por exemplo, “Jose G. Silva Jr.” se torna “Jose G Silva Jr ”.

4. Substituir seqüências de espaços em branco por um único espaço em branco: “Jose G Silva Jr ” passa a ser escrita “Jose G Silva Jr”.
5. Em cadeias de caracteres correspondentes a nomes de pessoas, expandir, através de um dicionário de abreviaturas, nomes que apareçam comumente simplificados. Por exemplo, “Jose G Silva Jr” se torna “Jose G Silva Junior”.
6. Substituir todas as letras maiúsculas por minúsculas. Por exemplo, “Jose G Silva Junior” se torna “jose g silva junior”.

```
1. <repositorio nome = "R">
2.          <Registro>
3.                  <titulo>Tratamento de herança em uma interface gráfica de consulta a bancos de dados relacionais</titulo>
4.                  <autor>Alberto Laender</autor>
5.                  <autor>Luciana Mendonça</autor>
6.                  <evento>sbbd</evento>
7.                  <data>1999</data>
8.          </Registro>
9.          <Registro>
10.                 <titulo>DEByE - uma ferramenta para extração de dados semi-estruturados</titulo>
11.                 <autor>Alberto Laender</autor>
12.                 <autor>Elaine S. Silva</autor>
13.                 <autor>Altigran S. da Silva</autor>
14.                 <evento>sbbd</evento>
15                  <data>1999</data>
16.         </Registro>
17.         <Registro>
18.                 <titulo>Uma interface gráfica para consulta a fontes de dados XML</título>
19.                 <autor>Tatiana Coelho</autor>
20.                 <autor>Alberto Henrique Frade Laender</autor>
21.                 <autor>Altigran Soares da Silva</autor>
22.                 <autor>Karine de Goes Louly</autor>
23.                 <evento>sbbd</evento>
24.                 <data>2000</data>
25.         </Registro>
26. </repositório>
```

Figura 2.3. Exemplo de Repositório Inicial.

O índice de autoria tem por finalidade armazenar as informações do repositório de forma normalizada e permitir a identificação do elemento <registro> de origem de cada elemento <autor> de R. Para isso, normalizamos também as cadeias de caracteres contidas nos elementos <título> e <evento>. Realizamos dois passos adicionais sobre os títulos: (1) tradução automática para a língua inglesa através da ferramenta de idiomas do Google[11] e (2) remoção de *stop words*.

[11] http://www.google.com - ferramentas de idiomas.

Ainda que a tradução não seja boa, o que pode ser observado através de uma rápida comparação entre as Figuras 2.3 e 2.4, destacamos a necessidade desse passo já que freqüentemente encontramos títulos escritos em diferentes idiomas. O efeito deste passo é a padronização das informações, o que, conforme Gu et. al. [2004], se faz necessário, já que a ausência desse passo permite que avaliações incorretas ocorram

Apresentamos na Figura 2.4 o índice de autoria para o repositório da Figura 2.3. Após realizarmos a análise sintática e a normalização das informações contidas no repositório inicial, criamos uma tupla para cada nome de autor em R. Por exemplo, no primeiro elemento <registro> do repositório R, Figura 2.3, temos dois nomes de autores: “Alberto Laender” e “Luciana Mendonça”. Geramos para este registro duas tuplas. Na primeira, “Alberto Laender” aparece como autor e “Luciana Mendonça” como co-autor. Na segunda ocorre o inverso. Lembrando que as posições são contadas a partir de 0, por exemplo, o código gerado para a tupla que armazena a cadeia de caracteres “Altigran S. da Silva”, do segundo registro de R, teria o código “1-2”, pois o registro ao qual a cadeia pertence aparece na posição 1 de R e a cadeia “Altigran S. da Silva” aparece na posição 2 desse registro. Note que, na Figura 2.4, as tuplas do índice de autoria estão ordenadas pelos nomes dos autores.

<2-1;alberto henrique frade laender;altigran soares da silva:karine de goes louly:tatiana coelho;graphical interface consultation sources data xml;sbbd;2000>
<0-0;alberto laender;luciana mendonca;treatment inheritance graphical interface consultation relationary data bases;sbbd;1999>
<1-0;alberto laender;elaine s silva:altigran s da silva;debye tool extration half structuralized data;sbbd;1999>
<1-2;altigran s da silva;alberto laender:elaine s silva;debye tool extration half structuralized data;sbbd;1999>
<2-2;altigran soares da silva;alberto henrique frade laender:karine de goes louly:tatiana coelho;graphical interface consultation sources data xml;sbbd;2000>
<1-1;elaine s silva;alberto laender:altigran s da silva;debye tool extration half structuralized data;sbbd;1999>
<2-3;karine de goes louly;alberto henrique frade laender:altigran soares da silva:tatiana coelho;graphical interface consultation sources data xml;sbbd;2000>
<0-1;luciana mendonca;alberto laender;treatment inheritance graphical interface consultation relationary data bases;sbbd;1999>
<2-0;tatiana coelho;alberto henrique frade laender:altigran soares da silva:karine de goes louly;graphical interface consultation sources data xml;sbbd;2000>

Figura 2.4. Índice de Autoria para o Repositório Inicial.

Observamos na Figura 2.4 a existência de oito cadeias de caracteres normalizadas de nomes de autores. Entretanto, esses nomes se referem a apenas seis autores. Por exemplo, as cadeias de caracteres nas tuplas 2-1, 0-0 e 1-0, apesar da escrita diferenciada, representam a

mesma pessoa. O mesmo ocorre com os nomes nas tuplas 1-2 e 2-2. As seções seguintes descrevem os módulos de pré-avaliação e de cálculo de similaridades que, juntos, compõem nossa estratégia para criação de arquivos de autoridade.

2.2. Pré-Avaliação

O módulo de pré-avaliação consiste em uma função para casamento de padrão que compara os nomes dos autores contidos no índice de autoria. Quando ocorre um casamento entre dois nomes de autores, estes são considerados candidatos a representarem a mesma entidade do mundo real. O módulo para cálculo de similaridades avalia as demais evidências, por exemplo, co-autores e título do trabalho. Sendo o resultado superior a um determinado limite arbitrário, o par avaliado é associado, formando um grupo. Um nome de autor ou um grupo pode também ser comparado a outro grupo. Neste caso, uma cadeia de caracteres gerada a partir das cadeias de nomes de autores contidas no grupo é utilizada para a função de casamento.

Utilizamos o casamento exato de padrão como função mais simples para o módulo de pré-avaliação. Neste caso, para que dois nomes sejam considerados candidatos, devem ter sua grafia idêntica. Na Figura 2.4, a tupla “2-1” seria comparada a todas as demais, sendo considerada candidata apenas a tupla de código “0-0”. Nenhum outro par seria considerado candidato. Entretanto, para chegarmos a esta conclusão, deveríamos realizar 36 comparações, tratando uma a uma todas as tuplas existentes. Uma simplificação pode ser realizada se considerarmos a forma com que nomes normalmente são escritos. Cadeias de caracteres que representam nomes de pessoas são formadas, no mínimo, pela primeira letra do primeiro nome mais o último sobrenome. Podemos, com isto, realizar uma poda, comparando apenas nomes iniciados pela mesma letra e com último sobrenome coincidente. Considerando apenas nomes com mesma letra inicial, realizaríamos apenas 10 avaliações e, considerando também o último sobrenome, 4 avaliações para a lista de tuplas da Figura 2.4.

Outra opção para a função de casamento é a distância de edição [Levenstein, 1965]. Neste caso, ao compararmos duas cadeias de caracteres, cada inserção, remoção ou substituição de um caracter é contabilizada como um erro. O número de erros permitidos

influencia, portanto, o resultado da comparação. Para a Figura 2.4, os autores contidos nas tuplas “1-2” e “2-2” só seriam considerados candidatos caso o número de erros permitidos fosse igual a 5. Apesar de sabermos que os nomes contidos nessas tuplas referem-se à mesma entidade do mundo real, o modo como nomes são representados sugere que uma função de casamento específica seja utilizada.

A comparação por fragmentos, inspirada nos trabalhos de Yanilos [1997] e French, Powell & Schulman [2000], é uma função de casamento de padrão que, através do algoritmo de distância de edição, avalia um a um cada fragmento de duas cadeias de caracteres que representam nomes. Os parâmetros necessários são duas cadeias de caracteres normalizadas, s_1 e s_2, e o limite L utilizado para a distância de edição. O resultado retornado é verdadeiro se s_1 e s_2 são compatíveis e eventualmente podem representar a mesma entidade do mundo real, e falso caso contrário. O limite L pode ser um número inteiro e, neste caso é utilizado diretamente pelo algoritmo, ou um número entre 0 e 1, em cujo caso o número máximo de erros permitidos será igual ao tamanho do menor fragmento comparado multiplicado por L. Por exemplo, para os fragmentos “jose” e “jonas”, se L = 3, então as cadeias são compatíveis ou, se L = 0,75, então o número de edições deve ser menor ou igual a 3, ou seja, $0{,}75 \times 4$ (que é o tamanho do menor fragmento comparado). Qualquer número menor que 0,75 faria com que as duas cadeias fossem consideradas incompatíveis.

O algoritmo na Figura 2.5 descreve a comparação por fragmentos. Considerando que nomes de pessoas são escritos minimamente com a inicial do primeiro nome e o último sobrenome, os dois passos a seguir são fundamentais para que qualquer par de cadeias de caracteres normalizadas seja candidato a representar a mesma entidade do mundo real. O primeiro passo para a comparação de duas cadeias é a verificação da compatibilidade do primeiro fragmento. Para isto, uma entre as quatro condições abaixo deve ser satisfeita:

1. Se tanto $s_1[1]$ quanto $s_2[1]$ forem formadas por mais de um caracter, então a distância de edição entre elas deve ser menor ou igual a L.
2. Se $s_1[1]$ contiver mais de um caracter e $s_2[1]$ tiver apenas um, então o primeiro caracter de $s_1[1]$ deve ser igual a $s_2[1]$.
3. Se $s_2[1]$ contiver mais que um caracter e $s_1[1]$ tiver apenas um, então o primeiro caracter de $s_2[1]$ deve ser igual a $s_1[1]$.
4. Se $s_1[1]$ e $s_2[1]$ tiverem apenas um caracter, então ambas devem ser iguais.

Essas condições correspondem aos testes realizados nas linhas 7 a 11 do algoritmo. O passo seguinte consiste em verificar o último sobrenome ou fragmento de s_1 e s_2. Para isto, considerando n_1 o número de fragmentos de s_1 e n_2 o número de fragmentos de s_2, a distância de edição entre $s_1[n_1]$ e $s_2[n_2]$ deve ser menor ou igual a L (linhas 13 a 15 do algoritmo). Na seqüência, passamos a avaliar os fragmentos restantes, que podem aparecer em qualquer ordem e abreviados através de iniciais. Avaliamos primeiramente fragmentos por extenso em s_1 e s_2. Caso encontremos quaisquer $s_1[i]$ e $s_2[j]$ cuja distância de edição seja menor que L, realizamos uma marcação dos dois fragmentos para que futuras comparações não sejam realizadas (linhas 17 a 20).

```
1 Comparação por Fragmentos (c1, c2: cadeia de caracteres; Lim: inteiro): lógica
2 variáveis
3     int n1, n2, i, j;
4     n1 ← número de fragmentos de c1;
5     n2 ← número de fragmentos de c2;
6 início
7     Se tamanho(c1[1]) > 1 e tamanho(c2[1]) > 1 então
8         Se Distância de Edição(c1[1], c2[1]) > Lim então retorne falso;
9     Senão Se tamanho(c1[1]) > 1 então
10        Se Primeira letra(c1[1]) != c2[1] então retorne falso;
11    Senão Se c1[1] != Primeira letra(c2[1]) então retorne falso;
12
13    Se tamanho(c1[n1]) > 1 e tamanho(c2[n2]) > 1 então
14        Se Distância de Edição(c1[n1], c2[n2]) > Lim então retorne falso;
15    Senão retorne falso;
16
17    Para i de 2 até n1-1 faça
18        Para j de 2 até n2-1 faça
19            Se tamanho(c1[i]) > 1 e tamanho(c2[j]) > 1 e Distância de Edição(c1 [i], c2[j]) < Lim então
20                Marcar c1 [i] e c2[j];
21    Para i de 2 até n1-1 faça
22        Para j de 2 até n2-1 faça
23            Se não marcado(c1[i]) e tamanho(c1 [i]) > 1 e tamanho(c2[j]) = 1 e Primeira letra(c1[i]) = c2[j] então
24                Marcar c1 [i] e c2[j];
25    Para i de 2 até n1-1 faça
26        Para j de 2 até n2-1 faça
27            Se tamanho(c1[i]) = 1 e não marcado(c2[j]) e tamanho(c2[j]) > 1 e c1[i] = Primeira letra(c2[j]) então
28                Marcar c1[i] e c2[j];
29    Para i de 2 até n1-1 faça
30        Para j de 2 até n2-1 faça
31            Se não marcado(c1[i]) e não marcado(c2[j]) e tamanho(c1[i]) = 1 e tamanho(c2[j]) = 1 e c1[i] = c2[j] então
32                Marcar c1[i] e c2[j];
33    Para i de 2 até n1-1 faça
34        Se não marcado(c1[i]) então
35            Para j de 2 até n2-1 faça
36                Se não marcado(c2[j]) então retorne falso;
37            retorne verdadeiro;
38 fim
```

Figura 2.5. Algoritmo para Comparação por Fragmentos.

Passamos então a comparar fragmentos por extenso de s1 com iniciais em s2 (linhas 21 a 24). A seguir comparamos iniciais em s_1 com nomes por extenso em s_2 (linhas 25 a 28) e, por

último, iniciais em s1 com iniciais em s2, linhas (29 a 32). Ao final, caso haja pelo menos um componente em s_1 e em s_2 que não esteja marcado, as cadeias de caracteres são consideradas incompatíveis, caso contrário, são consideradas compatíveis (linhas 33 a 37)

Existem outras variações possíveis para o algoritmo. Uma variação seria considerarmos que, qualquer cadeia de caracteres para ser considerada compatível com outra, além de satisfazer as condições anteriores, deve ainda ser formada por pelo menos dois fragmentos com mais que um caracter cada. Outra variação seria o modo como o limite é utilizado. No modo convencional, passamos o limite como parâmetro e este é utilizado independentemente do tamanho das cadeias de caracteres comparadas. Por exemplo, com um limite fixo L = 1, ao compararmos os fragmentos “sebastiao” e “sebastian”, teremos um casamento, o que também ocorre quando comparamos “lee” e “ley”. Poderíamos, alternativamente, passar um número entre 0 e 1 e realizarmos o cálculo do número máximo de erros permitido para a distância de edição. Para isto, multiplicaríamos esta constante pelo tamanho da menor cadeia comparada. Por exemplo, se tivéssemos uma constante $\alpha = 0,3$, o número de erros permitido para a primeira comparação seria $0,3 \times 9 = 2,7$, que arredondamos para 2, já para a segunda teríamos $0,3 \times 3 = 0,9$, portanto 0 erros. Mostramos no Capítulo 4 os resultados experimentais para estas variações.

2.3. Cálculo de Similaridades

Quando o módulo de pré-avaliação conclui que duas cadeias de caracteres são compatíveis, devemos ainda nos certificar se estas realmente representam a mesma entidade do mundo real. Para isto, necessitamos de fontes adicionais de evidência que sejam capazes de reforçar a conclusão do módulo de pré-avaliação. Metadados de objetos bibliográficos, em sua maioria, disponibilizam informações sobre títulos, co-autores e veículos de publicação. Podemos então, utilizar essas informações para concluirmos se dois nomes efetivamente se referem à mesma pessoa. Em nosso índice de autoria, essas informações, às quais nos referiremos daqui em diante por fontes de evidência, ficam disponíveis no arquivo de tuplas e nos arquivos invertidos associados.

A utilização dos títulos é válida se considerarmos que, freqüentemente, um pesquisador publica trabalhos sobre o mesmo assunto em veículos diferentes. Isto faz com que alguns de seus trabalhos tenham títulos compostos por palavras semelhantes. Utilizamos co-autores tendo em vista que pesquisadores costumam ter um grupo relativamente estável de colaboradores, dando origem a trabalhos em conjunto. O veículo de publicação, por sua vez, demonstra a linha de pesquisa na qual determinado autor trabalha. Outras informações poderiam também ser utilizadas como, por exemplo, a data em que determinado trabalho foi publicado. Por exemplo, podemos afirmar de forma mais confiável que duas cadeias de caracteres compatíveis representam a mesma entidade do mundo real se estiverem associadas a artigos publicados no mesmo evento em anos consecutivos. Uma informação útil na criação de arquivos de autoridade é a instituição de origem. Entretanto, esta informação na maioria das vezes não está disponível nos metadados de objetos bibliográficos.

Para avaliarmos a similaridade entre títulos, veículos de publicação e co-autores de trabalhos, utilizamos alguns métodos empregados em recuperação de informação. Particularmente, quando veículos de publicação estão disponíveis através de suas siglas, adotamos o casamento exato de padrão, assim como o fizemos para a informação adicional sobre a data. Quando temos à disposição informações escritas por extenso, utilizamos para avaliação a medida do cosseno [Baeza-Yates & Ribeiro-Neto, 1999; Chakrabarti, 2002], *bag of words* [Chakrabarti, 2002] e o coeficiente de Jaccard [Salton, 1988]. Para co-autores, utilizamos também a comparação por fragmentos, descrita pelo algoritmo da Figura 2.5.

A seguir, apresentamos uma breve descrição das principais medidas de similaridade usadas neste trabalho.

2.3.1. Medida do Cosseno

Considerando-se um repositório representado por um documento XML, como o mostrado na Figura 2.3, um conjunto de elementos de mesmo rótulo, por exemplo, <título>, é sempre formado por um vocabulário finito de *n* palavras. Em cada elemento do conjunto podem aparecer quaisquer palavras do vocabulário. Portanto, é possível criar uma representação vetorial em um espaço Euclidiano multidimensional para cada elemento do

conjunto. Cada eixo desse espaço corresponde a uma palavra. A coordenada de um elemento *e* na direção correspondente a uma palavra *p* é determinada por duas medidas:

- TF (*term frequency)*, que corresponde ao número de vezes que uma palavra *p* aparece em um elemento *e*;
- IDF (*inverse document frequency*), que corresponde ao peso das palavras de acordo com o inverso de sua freqüência nos elementos.

A medida IDF deve ser utilizada, pois os eixos no espaço vetorial não são igualmente importantes. Assim, palavras de maior freqüência no conjunto de elementos devem ser penalizadas. Sendo *E* o conjunto de todos os elementos de mesmo rótulo pertencentes a um repositório *R* e E_p o conjunto dos elementos de *E* que contêm determinada palavra *p*, uma forma comum para o cálculo do IDF de *p* é:

$$w_p = \log\left(1 + \frac{|E|}{|E_p|}\right)$$

Elementos longos tendem a ser favorecidos por conterem um maior número de palavras diferentes. Com isso, é necessário realizar uma normalização em função do tamanho do elemento, que é determinada pela fórmula:

$$w_e = \sqrt{\sum {w_{e,p}}^2}$$

onde $w_{e,p}$ corresponde ao peso das palavras em relação ao elemento e é calculado através da regra TF×IDF

$$w_{e,p} = (1 + \log(f_{e,p})) \times w_p$$

sendo $f_{e,p}$ o número de ocorrências da palavra *p* no elemento *e*.

A similaridade entre dois elementos é calculada através da medida do cosseno entre suas representações vetoriais. Quanto maior o cosseno, maior a similaridade. Para isto, utilizamos a fórmula:

$$\cos(e_1, e_2) = \frac{1}{w_{e_1} w_{e_2}} \sum_{p \in (e_1 \cap e_2)} (w_{e_1,p} \times w_{e_2,p})$$

Estabelecemos experimentalmente um limite inferior, α, para a medida do cosseno entre dois elementos. Qualquer par de tuplas do índice de autoria que contenha dois elementos de mesmo rótulo com medida do cosseno superior a α é considerado similar por indicação desta fonte de evidência.

2.3.2. Coeficiente de Jaccard

Uma forma mais simples para se computar a similaridade entre elementos é o coeficiente de Jaccard. Neste método, consideramos que um elemento *e* pode ser representado através do conjunto de palavras que o compõe, denotado por *P(e)*. Para esta medida, o número de ocorrências das palavras em cada elemento é indiferente. A similaridade entre dois elementos, e_1 e e_2, é calculada através da fórmula:

$$Jac(e_1, e_2) = \frac{|P(e_1) \cap P(e_2)|}{|P(e_1) \cup P(e_2)|}$$

que representa a razão entre o número de palavras em comum entre e_1 e e_2 e o número de palavras diferentes que compõem e_1 e e_2.

Estabelecemos experimentalmente um limite inferior, α, para o coeficiente de Jaccard entre dois elementos. Qualquer par de tuplas que contenha dois elementos de mesmo rótulo com coeficiente de Jaccard superior a α é considerado similar por indicação desta fonte de evidência.

2.3.3. *Bag of Words*

O *bag of words* é a mais simples entre as medidas de similaridade adotadas neste trabalho. Considerando novamente que um conjunto de elementos de mesmo rótulo é composto por um vocabulário finito de *n* palavras, podemos representar cada elemento *e* como um vetor *n*-dimensional. Cada eixo desse vetor representa uma palavra *p* e recebe o valor 1 caso *p* esteja presente em *e*, e 0 caso contrário. Assim como no Coeficiente de Jaccard, não nos importamos com o número de ocorrências de cada palavra em um documento. Para computarmos a similaridade entre dois elementos, e_1 e e_2, realizamos o produto interno entre os vetores $\vec{e_1}$ e $\vec{e_2}$ que os representam:

$$bow(e_1, e_2) = \vec{e_1} \cdot \vec{e_2}$$

Para isso, definimos experimentalmente um número mínimo de ocorrências, α, que deve ser satisfeito para que dois elementos sejam considerados similares.

2.4. Algoritmo para Criação de Arquivos de Autoridade

Nesta seção descrevemos o algoritmo proposto neste trabalho para criação de arquivos de autoridade. Antes, porém, apresentamos a definição do conceito de cadeia de caracteres mais representativa, que será útil para nossa discussão a seguir.

Definição 5. *Dado um conjunto de cadeias de caracteres normalizadas, C, a* ***cadeia de caracteres mais representativa,*** c_r*, é formada a partir da composição das cadeias de caracteres* $c_1, c_2, ..., c_m \in C$*, de modo que, para qualquer* $c_i \in C$*, se* $c_i[q]$ *não faz parte de* c_r *então, se existir* $c_r[r]$*, formado por apenas um caracter e, se o primeiro caracter de* $c_i[q]$ *for igual a* $c_r[r]$*, então* $c_r[r]$ *é substituído por* $c_i[q]$*. Caso contrário,* $c_i[q]$ *é inserido em* $c_r[r]$*.*

Por exemplo, a cadeia de caracteres mais representativa para C = {“j g antonio silva”, “jose a silva”} é c_r = “jose g antonio silva”. O cálculo da cadeia mais representativa facilita o

trabalho do módulo de pré-avaliação, que passa a analisar cadeias de caracteres com maior número de fragmentos.

Índices unificados, assim como arquivos de autoridade, mantêm a correspondência entre todas as formas permissíveis de cadeias de caracteres em um campo bibliográfico em particular, por exemplo, nomes de autores ou títulos de periódicos [Auld, 1982]. Deste modo, um índice unificado pode ser formado, por exemplo, pela cadeia de caracteres mais representativa para uma determinada entidade e uma lista com referências para os registros nos quais figuram as cadeias de caracteres em sua forma original. A Figura 2.6 traz o índice unificado exemplo para o repositório da Figura 2.3.

<alberto henrique frade laender;0-0:1-0:2-1>
<altigran soares da silva;1-2:2-2>
<elaine s silva;1-1>
<karine de goes louly;2-3>
<luciana mendonca;0-1>
<tatiana coelho;2-0>

Figura 2.6. Arquivo de Autoridade Exemplo.

A primeira entrada do arquivo mostra a cadeia de caracteres mais representativa para a entidade participante do elemento <registro> 0, posição 0, elemento <registro> 1, posição 0, e elemento <registro> 2, posição 1. Isto significa que essas cadeias são formas variantes da escrita do nome de um único autor no mundo real.

Apresentamos na Figura 2.7 o algoritmo para criação de índices unificados, conforme descrito acima. O algoritmo recebe sete parâmetros de entrada: o índice de autoria; o limite de similaridade, que deve ser superado para que tuplas ou grupos sejam associados, gerando novos grupos; um valor α para nomes de autores, que é utilizado pela função de casamento de padrão; um valor α para títulos, utilizado como valor mínimo para que dois títulos sejam considerados similares; um valor α para nomes de co-autores, também utilizado como mínimo para similaridade entre esses nomes; o tamanho do índice de autoria, que corresponde ao número de tuplas que o compõem; e um valor α para datas. Vale dizer que esses parâmetros podem mudar de acordo com as fontes de evidência disponíveis. Por exemplo, se não houver datas disponíveis, não precisamos do último parâmetro.

```
1 Agrupa (Ia: índice de autoria; limite_sim, alfa_autores, alfa_títulos, alfa_coautores: real; tam_ia, alfa_data: inteiro): arquivo de autoridade
2 variáveis
3       i, j: inteiro;
4       grupo[tam_ia], avaliado[tam_ia]: vetor de inteiro;
5       representante[tam_ia], componentes[tam_ia]: vetor de cadeia de caracteres;
6       resultado: arquivo de autoridade;
7 início
8       Inicializações:
9       para i de 0 até tam_ia faça
10          grupo[i] ← componentes[i] ← i;
11          avaliado[i] ← 0;
12          representante[i] ← tupla[i].autor;
13      fim-para;
14
15      Pesquisa por tuplas candidatas e cálculo de similaridades:
16      para i de 0 até tam_ia faça
17          para j de i+1 até tam_ia faça
18              se (Primeira_letra(representante[grupo[i]]) = Primeira_letra(representante[grupo[j]])) então
19                  se (!avaliado[grupo[j]]) e (grupo[i] != grupo[j]) então
20                      avaliado[grupo[j]] = 1;
21                      se match(representante[grupo[i]], representante[grupo[j]] ) > alfa_autores então
22                          se Similaridade(grupo[i], grupo[j], alfa_títulos, alfa_coautores, alfa_data) > limite_sim então
23                              representante[grupo[i]] ← novo_representante(representante[grupo[i]], representante[grupo[j]]);
24                              componentes[grupo[i]] ← "componentes[grupo[i]]:componentes[grupo[j]]";
25                              para cada c ∈ componentes[grupo[j]] faça
26                                  grupo[c] ← grupo[i];
27                              fim-para
28                              grupo[j] ← grupo[i];
29                              componentes[grupo[j]] ← "";
30                          fim-se;
31                      fim-se;
32                  senão
33                      j ← tam_ia;
34                  fim-se
35              fim-se
36          fim-para;
37          para j de i+1 até tam_ia faça
38                  avaliado[grupo[j]] = 0;
39      fim-para;
40
41      Criação do arquivo de autoridade:
42      para i de 0 até tam_ia faça
43          se componentes[i] != "" então
44              escreva em resultado "representante[i];";
45              para cada c ∈ componentes[i] faça
46                  escreva em resultado "tupla[componentes[c]].código:";
47              fim-para;
48          fim-se;
49      fim-para;
50      retorne resultado;
51 fim
```

Figura 2.7. Algoritmo para Criação de Arquivos de Autoridade.

Podemos também destacar algumas estruturas de dados importantes em nosso algoritmo. A primeira, denominada *grupo*, consiste em um vetor de inteiros que é preenchido com a posição inicial de cada entrada da lista de tuplas. A primeira tupla tem, portanto, *grupo*

= 0. A segunda estrutura, denominada *componentes*, armazena, para cada grupo, a posição de seus componentes dentro do índice de autoria. Inicialmente, cada tupla faz parte de seu próprio grupo, que contém apenas um elemento. A próxima estrutura, de nome *avaliado*, armazena, na fase de processamento, quais grupos já foram avaliados, evitando processamentos repetidos. A última estrutura, denominada *representante*, consiste em um vetor de cadeia de caracteres que, inicialmente, armazena as cadeias de caracteres correspondentes a nomes da forma como elas aparecem na lista de tuplas e, a cada nova inclusão de um membro ao grupo, a cadeia de caracteres mais representativa é recalculada.

Nas linhas 15 e 16 observamos os dois laços principais do algoritmo. Para cada nova tupla percorrida pelo laço externo, representada por *grupo*[i], todas as tuplas, da posição seguinte até a última posição do índice de autoria, são percorridas pelo laço interno, representadas por *grupo*[j], tendo em vista que todas as tuplas representam grupos inicialmente formados por um único *componente*. Considerando que o índice de autoria é ordenado pelas cadeias de caracteres de nomes, caso o *representante* do *grupo*[i] seja iniciado por letra diferente do *representante* do *grupo*[j] (linha 17) o processamento para *grupo*[i] é interrompido imediatamente (linha 32) e um novo grupo passa a ser analisado. Para que dois grupos sejam avaliados pela função de casamento de padrão do módulo de pré-avaliação, o *grupo*[i] deve ser diferente do *grupo*[j] (linha 18). Além disso, nenhum outro membro do *grupo*[j] deve ter sido comparado ao *grupo*[i] (linha 18) pois, desta avaliação, só existem dois resultados possíveis:

1. Os avaliados são considerados compatíveis e, neste caso, todos os *componentes* dos dois grupos são unidos, gerando um novo grupo (linhas 21 a 27) ou
2. Os avaliados são considerados incompatíveis e, neste caso, nenhum dos *componentes* do *grupo*[i] pode ser compatível com qualquer componente do *grupo*[j].

Na linha 21, o módulo de pré-avaliação compara os representantes do *grupo*[i] e *grupo*[j]. As funções possíveis para esta comparação são o casamento exato de padrão, distância de edição e comparação por fragmentos. Caso o resultado da comparação seja superior ao limite α para nomes de autores, o módulo para cálculo de similaridades recebe o *grupo*[i] e o *grupo*[j] como parâmetros, além dos valores de α para cada fonte de evidência (linha 22). Caso a similaridade calculada entre os grupos seja maior que o limite α, os grupos são unidos, gerando um novo grupo (linhas 21 a 28). O cálculo da similaridade é realizado avaliando-se cada membro do *grupo*[i] com cada membro do *grupo*[j] individualmente.

Para criarmos o arquivo de autoridade (linhas 38 a 45), utilizamos um laço que analisa as estruturas de dados descritas anteriormente. Para cada grupo do índice de autoria, se *componentes* for diferente de vazio, o *representante* é escrito no arquivo de autoridade (linha 42) bem como a lista com os códigos identificadores das cadeias de caracteres com nomes de autores no repositório (linha 45).

No próximo capítulo, mostramos os resultados experimentais obtidos através da combinação das fontes de evidência disponíveis com as medidas de similaridade descritas anteriormente utilizando duas coleções de teste: uma derivada do repositório da BDBComp e outra da DBLP.

Capítulo 3

Resultados Experimentais

Para avaliarmos nossa estratégia para criação de índices unificados, realizamos experimentos utilizando duas coleções de teste. A primeira coleção foi criada a partir do repositório da BDBComp [Laender, Gonçalves & Roberto, 2004] e, através de consultas a diferentes fontes, por exemplo, LATTES[12] e Google[13], geramos manualmente o seu índice unificado. Obtivemos nossa segunda coleção junto ao grupo de pesquisa coordenado pelo professor Lee Giles[14] da *Pennsylvania State University*. Essa coleção foi criada a partir da DBLP[15] e contava com um índice unificado também gerado manualmente.

De posse dessas coleções, geramos automaticamente diversos índices unificados e fizemos comparações aos arquivos criados manualmente. O ambiente experimental predominantemente utilizado foi um computador com processador Celeron 2.4 com 512MB de memória principal. Toda a implementação foi realizada utilizando a linguagem de programação PERL. Mostramos na Seção 3.1 mais detalhes sobre as coleções de teste utilizadas, descrevemos as medidas de qualidade adotadas na Seção 3.2 e analisamos os resultados obtidos na Seção 3.3.

3.1. Dados Sobre as Coleções Teste

A primeira coleção de teste utilizada se baseia no repositório da BDBComp do dia 14/07/2004. Os dados contidos neste repositório se referem à produção nacional na área de

[12] http://buscatextual.cnpq.br/buscatextual/index.jsp
[13] http://www.google.com.br
[14] http://clgiles.ist.psu.edu
[15] http://dblp.uni-trier.de

Ciência da Computação. Autores brasileiros desta área, na maioria das vezes, escrevem seus nomes por extenso em trabalhos ou, alternativamente, abreviam os nomes intermediários. Como conseqüência, o repositório da BDBComp é relativamente bem comportado, contendo um baixo percentual de nomes ambíguos e de cadeias de caracteres com algum erro de grafia. Deste modo, ao realizarmos experimentos preliminares, percebemos que os casos onde o problema de ambigüidade em nomes se manifesta representam uma pequena parcela do repositório. Optamos então por trabalhar apenas com os nomes cujo último sobrenome é um dos cem mais freqüentes do repositório. Criamos índices de autoria individuais para cada um desses 100 sobrenomes. Em seguida, realizamos uma análise descartando 14 índices de autoria formados apenas por entradas com nomes correspondentes a um único autor. Isso se deve ao fato de que esses grupos poderiam distorcer os resultados já que, para eles, a solução ótima poderia ser alcançada trivialmente unindo todas as entradas em um único grupo. Os índices de autoria restantes somados totalizaram 2767 entradas e 1781 grafias diferentes, sendo que constam 66 nomes formados pela primeira inicial e o último sobrenome, a que chamamos nomes curtos. As 2767 entradas correspondem a 1365 autores diferentes, média de 2 entradas por autor. Os 86 sobrenomes mais freqüentes aparecem, portanto, em 37% das entradas do índice de autoria inicial. Realizamos experimentos gerando índices unificados para cada um dos 86 índices de autoria. Avaliamos a qualidade de cada índice unificado gerado e então obtivemos a média para os índices de autoria. Os resultados obtidos são mostrados na Seção 3.3.

A segunda coleção foi montada com base no repositório da DBLP, com 300.000 registros bibliográficos XML. Os atributos em cada registro foram organizados de modo similar às tuplas de nosso índice de autoria. Os nomes de autores com mesma letra inicial e último sobrenome foram agrupados gerando conjuntos. Cada conjunto contém um determinado número de registros bibliográficos e autores. Foram escolhidos conjuntos com mais de dez autores diferentes. A Tabela 3.1 mostra as composições de primeira letra e último sobrenome e o número de conjuntos por composição.

Podemos observar pela tabela que o segundo repositório de testes conta com 4297 entradas para 222 autores diferentes, totalizando uma média de aproximadamente 19 entradas por autor. Observamos ainda a presença de 208 grafias diferentes e 2270 nomes curtos, o que totaliza 52% do total de nomes na coleção. Inicialmente, a coleção era formada por 100% de

nomes curtos, o que foi provocado artificialmente para que pudesse ser usada nos experimentos descritos por Han et al [2004, 2005]. Nesses trabalhos, o tema é a remoção de ambigüidades em citações bibliográficas onde, por padrão, nomes de autores aparecem resumidos na forma de nomes curtos. Em nosso caso, como o enfoque é a remoção de ambigüidades em bibliotecas digitais, necessitávamos que a coleção estivesse o mais semelhante possível à original. Considerando que no repositório original da DBLP essas informações estavam presentes, conseguimos realizar uma extração dos nomes do repositório XML[16] da DBLP com base nos títulos dos artigos, atingindo 52% de nomes curtos e chegando a uma situação mais próxima da realidade.

Composições	Nº de Entradas	Nº de autores	Grafias Diferentes	Nomes Curtos
jmartin	112	16	16	32
mbrown	153	13	13	74
jrobinson	171	12	13	58
akumar	242	14	13	94
mjones	260	13	14	121
ktanaka	280	10	11	136
djohnson	368	15	13	206
mmiller	412	12	11	341
agupta	576	26	23	196
cchen	799	61	57	435
jsmith	924	30	24	577

Tabela 3.1. Dados Sobre a Coleção de Teste Extraída da DBLP.

Para que pudéssemos avaliar o resultado de nossos experimentos, fizemos um levantamento para determinar as principais medidas para avaliação da qualidade de índices unificados e na subseção a seguir destacamos as que consideramos mais relevantes para o nosso trabalho.

[16] Disponível em http://dblp.uni-trier.de/xml/

3.2. Medidas de Qualidade

Encontramos na literatura uma grande variedade de medidas aplicáveis à avaliação de grupos gerados por algoritmos de agrupamento (*clustering*). Medidas como *entropia* e *pureza* recebem descrições similares por Dayanik & Nevill-Manning [2004], French, Powell & Schulman [2000], Rossel, Kann & Litton [2004] e Strehl, Ghosh & Mooney [2000], sendo utilizadas com grande freqüência em conjunto ou separadamente. As medidas *revocação* e *precisão* [Baeza-Yates & Ribeiro-Neto, 1999; Chakrabarti, 2002], tradicionais na área de Recuperação de Informação, são também utilizadas por Carvalho & Silva [2003], Cohen & Richman [2002], Rossel, Kann & Litton [2004]. Cohen & Richman [2002] descrevem uma forma adaptada de *revocação* e *precisão* para algoritmos de agrupamento. Tejada, Knoblock & Minton [2001] e Han et al [2004] avaliam os resultados experimentais através da medida acurácia (*accuracy*). Entretanto, optamos neste trabalho pela utilização das medidas PMG (pureza média por grupo) [Lapidot, 2002; Solomonoff et al, 1998], PMA (pureza média por autor), adaptada da medida ASP (*average speaker purity*) [Lapidot, 2002, Ajmera, Bourlard & Lapidot, 2002], e K [Lapidot, 2002], que corresponde à média geométrica entre PMG e PMA. Essa decisão se deve ao fato de as medidas PMG e PMA capturarem de forma mais completa a essência dos grupos gerados.

Para discutirmos as medidas acima, consideremos o índice de autoria representado na Figura 3.1. Na Figura 3.1(a), cada nodo corresponde a uma tupla do índice de autoria. Na Figura 3.1(b), grupos formados por nodos compõem o índice unificado gabarito, criado manualmente, para o repositório representado pelo índice de autoria da Figura 3.1a.

Figura 3.1. Representações para um Índice de Autoria e para um Índice Unificado

A medida *entropia* considera como os vários autores do índice de autoria foram distribuídos dentro de cada grupo. Encontramos fórmulas diferenciadas nos trabalhos que utilizam essa medida, entretanto, todas elas se baseiam no mesmo conceito e tendem a gerar resultados aproximados para índices unificados com a mesma característica. O valor ótimo para esta medida é obtido trivialmente quando nenhum agrupamento é realizado, por exemplo, para um índice unificado exatamente igual ao índice de autoria da Figura 3.1a, o valor da *entropia* é igual a 0. A medida *pureza*, que constantemente é utilizada em associação à *entropia*, considera que um determinado grupo está associado ao autor que detém as tuplas predominantes neste grupo. Por exemplo, na Figura 3.2, o grupo formado pelas tuplas 1, 2 e 3 foi ligado ao grupo formado pelas tuplas 4 e 5. O grupo gerado pertence então ao autor dos trabalhos 1, 2 e 3 que, pelo índice unificado da Figura 3.1b, sabemos ser o mais numeroso. No entanto, o grupo formado pelos trabalhos 4 e 5 não foi avaliado pela medida. A medida *pureza*, assim como a *entropia*, atinge seu valor ótimo quando nenhum agrupamento é realizado. Por permitirem que uma parcela dos grupos gerados por nossa estratégia fosse não avaliada, concluímos que as medidas pureza e entropia não seriam ideais para serem adotadas como métricas de avaliação. Além disso, essas medidas não avaliam o efeito a que chamamos "fragmentação". Este efeito se torna perceptível quando observamos que um conjunto de trabalhos pertencentes a um único autor do índice unificado usado como gabarito aparece subdividido em um índice unificado gerado automaticamente.

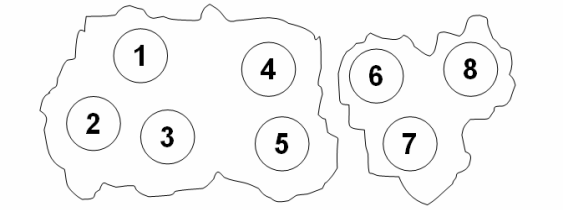

Figura 3.2. Arquivo de Autoridade Exemplo para Pureza

Analisamos também as medidas *revocação* e *precisão*. Para utilizá-las, poderíamos considerar que cada grupo no índice unificado gerado automaticamente corresponde à resposta a uma consulta. Neste caso, *revocação* corresponderia à razão entre o número de elementos corretamente inseridos em cada grupo do índice unificado gerado automaticamente pelo número de elementos constantes em cada grupo correspondente do índice unificado gabarito e *precisão* corresponderia à razão entre o número de elementos corretamente inseridos em cada grupo do índice unificado gerado automaticamente pelo número total de elementos de cada grupo do índice unificado gerado. Esta seria uma forma adaptada das medidas de *revocação* e *precisão*, pois não seria possível traçar um gráfico tendo as duas medidas como eixos. Para obtermos a *revocação* e *precisão* de um índice unificado gerado, calcularíamos a média de todos os grupos pertencentes a este arquivo. Esta medida é mais adequada que *entropia* e *pureza* para a avaliação dos índices unificados gerados por nossa estratégia, no entanto, estaríamos novamente sujeitos à não avaliação de certos grupos como o exemplo da Figura 3.2, já que, novamente, autores com um número pequeno de trabalhos publicados poderiam não ter seus grupos avaliados. Estas medidas capturam melhor a essência dos grupos gerados, penalizando, através da *revocação,* a fragmentação. A medida F1, ou *F-measure*, consiste na média harmônica entre *revocação* and *precisão* [Baeza-Yates & Ribeiro-Neto, 1999; Chakrabarti, 2002].

A medida *acurácia* pode ser utilizada de forma semelhante à *precisão* e *revocação*, avaliando cada grupo do índice unificado gerado automaticamente de forma individual e calculando em seguida a média para todos os grupos. Essa medida consiste no percentual de elementos corretamente inseridos em um determinado grupo. Sua utilização não é adequada para avaliação dos arquivos gerados por nossa estratégia, pois não considera a fragmentação e, além disso, deveríamos ainda decidir a que autor cada grupo gerado se refere.

Adotamos neste trabalho as medidas PMA, PMG e K para avaliação da qualidade dos índices unificados gerados. A medida PMG mede o quanto elementos de grupos diferentes do índice unificado gabarito foram misturados no índice unificado gerado. Deste modo, quanto maior for o número de elementos inseridos erroneamente em grupos do índice unificado gerado automaticamente, mais baixo será o valor de PMG, que varia de 0 a 1. A fórmula para o cálculo de PMG é

$$PMG = \frac{1}{N} \sum_{i=0}^{q} n_i \sum_{j=0}^{R} \frac{n_{ij}^2}{n_i^2}$$

onde: R é o número de grupos no gabarito;

N é o número total de tuplas do índice de autoria;

q é o número de grupos gerados automaticamente;

n_{ij} é o número total de elementos do grupo i gerado automaticamente pertencente ao grupo j do gabarito;

n_i é o número total de elementos do grupo i gerado automaticamente.

A medida PMA tem como finalidade medir o grau de fragmentação dos índices unificados gerados. Seus valores podem variar entre 0 e 1 e, quanto menos fragmentado estiver um índice unificado, mais próximo de 1 será o valor de PMA. A fórmula para PMA é

$$PMA = \frac{1}{N} \sum_{j=0}^{R} n_j \sum_{i=0}^{q} \frac{n_{ij}^2}{n_j^2}$$

onde: n_j é o número total de elementos do grupo *j* do gabarito.

A medida K consiste na média geométrica entre PMG e PMA, sendo expressa pela fórmula: $K = \sqrt{PMG \times PMA}$, ela é utilizada para verificarmos o equilíbrio entre as medidas. O melhor caso possível é atingido quando tanto PMG quanto PMA são iguais a 1. Essas medidas apresentam ainda a vantagem de realizarem uma ponderação considerando o tamanho dos grupos gerados.

Para exemplificar, consideremos novamente o índice de autoria expresso na Figura 3.1a e o índice unificado proposto na Figura 3.2. Neste caso temos:

$PMG = \frac{1}{8} \times \left(\frac{13}{25} \times 5 + 1 \times 3 \right) = 0.7$, $PMA = \frac{1}{8} \times (1 \times 3 + 1 \times 2 + 1 \times 3) = 1$ e $K = \sqrt{0,7 \times 1} = 0,8366$.

Como podemos observar, tivemos uma união de dois grupos, o que gerou uma penalização pela medida PMG. Por outro lado, consideremos que o resultado do agrupamento fosse o índice de autoria da Figura 3.1a, ou seja, que nenhum agrupamento houvesse sido realizado. Neste caso, temos $PMG = \frac{1}{8}(8) = 1$, $PMA = \frac{1}{8} \times 3 = 0,375$ e $K = \sqrt{1 \times 0,375} = 0,6123$. Observamos que as medidas capturam de forma correta a essência dos agrupamentos gerados, obtendo um equilíbrio entre a organização dos grupos e não permitindo que haja fragmentação.

Mostramos na subseção a seguir os resultados obtidos para as coleções da BDBComp e DBLP.

3.3. Resultados Obtidos

3.3.1. Experimentos com a Coleção da BDBComp

O primeiro conjunto de experimentos foi executado com a coleção de teste da BDBComp, considerando as entradas de seu índice de autoria que continham os 86 sobrenomes mais freqüentes. Realizamos primeiramente experimentos para observarmos a efetividade das funções de casamento de padrão aplicadas aos nomes dos autores. Para isso, não utilizamos o módulo para cálculo de similaridades, agrupando entradas consideradas compatíveis pelo módulo de pré-avaliação. Aplicamos essas funções de dois modos diferentes a cada um dos 86 índices de autoria gerados. No primeiro, realizamos uma poda sobre os índices de autoria, descartando comparações entre nomes que se iniciavam com letras diferentes e, no segundo, não realizamos nenhuma poda. Os resultados para esta avaliação se encontram na Tabela 3.2.

Método	Número de Grupos Gerados	PMG	PMA	K
Comparação por Fragmentos (L = 0,25)	15,6977	0,9671	0,9548	0,9596
Comparação por Fragmentos (L = 1)	15,6744	0,9671	0,9576	0,9612
Distância de Edição (L = 7)	14,1628	0,9021	0,9184	0,9076
Comparação por Fragmentos (L = 0,25)	15,6744	0,9654	0,9548	0,9588
Comparação por Fragmentos (L = 1)	15,6628	0,9660	0,9576	0,9606
Distância de Edição (L = 2)	19,0814	0,9721	0,7571	0,8526

Tabela 3.2. Resultados para Agrupamentos Realizados sobre a Coleção de Teste da BDBComp.

Na tabela, a primeira coluna informa o método utilizado, a segunda mostra o número médio de grupos gerados, a terceira trás o PMG médio, a quarta trás o PMA médio e a última trás o K médio. Os três primeiros resultados foram gerados com a poda descrita anteriormente e os três últimos sem nenhuma poda. Geramos esta tabela variando L de 0 a 0,75 em passos de 0,05 e de 1 a 6 em passos de 1, para a comparação por fragmentos, e de 0 a 12 em passos de 1, para a distância de edição. Inserimos na tabela apenas os resultados que apresentaram o maior K. Podemos, portanto, observar que, para a comparação por fragmentos, os resultados se mantiveram inalterados quando utilizamos poda ou não, sendo que os resultados foram semelhantes para L = 0,25 e L = 1. Já para distância de edição, a utilização da poda fez com que o melhor resultado fosse obtido para L = 7 e, sem poda, para L = 2, o que nos faz concluir que a poda influenciou diretamente esse resultado. Verificamos com isso que a comparação por fragmentos é uma função mais estável e pode ser considerada a mais apropriada para o módulo de pré-avaliação. Além disso, devemos destacar que os resultados obtidos com esse método atingiram PMG e PMA médios superiores a 0,95, e que o número médio de grupos gerados ficou entre 15,66 e 15,69 o que é um resultado satisfatório, visto que nos 86 índices unificados gabarito esse número médio foi de 15,8721. Concluímos com isso que, para uma coleção de teste com nomes de autores completos disponíveis na maioria dos registros, como a da BDBComp, uma boa função de casamento de padrão é suficiente para gerar índices unificados de boa qualidade. Nos certificamos desse fato realizando avaliações pelo cálculo de similaridade de todas as fontes de evidência disponíveis, a saber: títulos, co-autores, eventos e datas de publicação. No entanto, nenhuma dessas fontes melhorou o resultado obtido através do módulo de pré-avaliação.

3.3.2. Experimentos com a Coleção da DBLP

Nosso segundo conjunto de experimentos foi realizado sobre a coleção de teste da DBLP. Essa coleção tem características diferentes da primeira, sendo que aproximadamente 52% dos nomes são curtos, ou seja, formados apenas pela primeira letra do primeiro nome e pelo último sobrenome. Deste modo, pudemos verificar se o módulo de pré-avaliação seria novamente o suficiente para a criação de índices unificados ou se informações adicionais poderiam melhorar os resultados. Vale a pena ressaltar que o número total de grupos no índice unificado gabarito é 222. A primeira tentativa que realizamos consistiu em criar índices unificados utilizando apenas o módulo de pré-avaliação. Caso este retornasse uma resposta positiva, cada par analisado era associado formando um novo grupo. Visto que esse fragmento de repositório é formado em grande parte por nomes curtos, adotamos duas estratégias diferentes para tratá-los. No primeiro caso, não associamos nomes curtos ainda que fossem considerados compatíveis. Essa estratégia privilegia a PMG em detrimento à PMA, visto que tende a gerar um grande número de grupos. No segundo caso, associamos todos os nomes curtos que fossem considerados compatíveis pelo módulo de pré-avaliação, o que tende a gerar grupos com baixa PMG e melhor PMA. Os resultados obtidos para as funções de casamento de padrão combinadas às estratégias descritas são mostrados na Tabela 3.3.

Método	Número de Grupos Gerados	PMG	PMA	K
Distância de Edição (L = 0)	2473	0,9634	0,3164	0,5521
Comparação por Fragmentos (L = 0,2)	2440	0,9274	0,3212	0,5458
Distância de Edição (L = 0)	214	0,5522	0,6683	0,6075
Comparação por Fragmentos (L = 0,2)	167	0,4806	0,7119	0,5849

Tabela 3.3. Resultados para Agrupamentos Realizados sobre a Coleção da DBLP.

Na tabela, os dois primeiros resultados foram gerados segundo a estratégia de não agrupar entradas do índice de autoria que contivessem nomes curtos. Desse modo, um grande número de grupos foi criado, fazendo com que a medida PMA, que avalia a fragmentação, ficasse abaixo de 0,35 tanto para distância de edição quanto para comparação por fragmentos. Por outro lado, a medida PMG, que avalia a qualidade dos grupos gerados, ficou acima de 0,9 para ambos os métodos, o que sugere que não agrupar nomes curtos provoca a criação de um

número muito acentuado de grupos, mas com boa qualidade. Os dois últimos resultados foram gerados segundo a estratégia de agrupar todas as entradas do índice de autoria que contivessem nomes curtos. Desse modo, geramos um número reduzido de grupos, melhorando a medida PMA, mas piorando muito a medida PMG.

Observamos que apenas a utilização do módulo de pré-avaliação não é suficiente para gerar índices unificados satisfatórios, visto que, se geramos grupos com qualidade elevada, a fragmentação aumenta demasiadamente e, por outro lado, se geramos um número reduzido de grupos, melhoramos a fragmentação, mas pioramos consideravelmente a medida PMG. Optamos com isso por avaliar também as fontes de evidência, títulos, eventos e co-autores, combinadas com o módulo de pré-avaliação utilizando comparação por fragmentos. Com isso, cada par de tuplas do índice de autoria avaliado foi associado gerando um novo grupo apenas quando o módulo de pré-avaliação confirmou a compatibilidade dos nomes e o módulo para cálculo de similaridades confirmou sua similaridade.

A Tabela 3.4 mostra os melhores resultados obtidos usando os títulos dos trabalhos como fonte de evidência e os métodos *bag of words*, coeficiente de Jaccard e medida do cosseno para cálculo da medida de similaridade. Os resultados foram selecionados como se segue. Para cada método, escolhemos o valor de α que obteve PMG imediatamente superior a 0,9 para que pudéssemos avaliar qual seria a PMA se só tivéssemos títulos de trabalhos como fonte de evidência e a prioridade fosse a geração de índices unificados com alta precisão. Em outras palavras, gostaríamos de saber qual seria a fragmentação gerada caso fôssemos criar um índices unificados tendo apenas essa fonte de evidência para o cálculo de similaridade. Para cada método, também escolhemos o valor de α que produziu PMG imediatamente superior a 0,7. Dessa vez, fizemos um relaxamento da PMG mínima para que obtivéssemos uma elevação da PMA, ou seja, estamos interessados em índices unificados em que a fragmentação seja menor, mas ainda mantendo o compromisso com a qualidade dos grupos gerados.

Método	Número de Grupos Gerados	PMG	PMA	K
Bag of Words (α = 2)	1726	0,8228	0,2886	0,4873
Bag of Words (α = 3)	2764	0,9814	0,1182	0,3406
Coeficiente de Jaccard (α = 0,15)	1403	0,7260	0,3590	0,5105
Coeficiente de Jaccard (α = 0,25)	2204	0,9063	0,1728	0,3958
Medida do cosseno (α = 0,2)	1558	0,7844	0,3251	0,5050
Medida do cosseno (α = 0,3)	2293	0,9486	0,1544	0,3827

Tabela 3.4. Resultados da Utilização da Fonte de Evidência Título na Geração de índices unificados para a Coleção da DBLP.

Podemos observar na tabela que o método que resultou na melhor média geométrica, K, entre PMG e PMA foi o coeficiente de Jaccard. Para PMG imediatamente superior a 0,7, este método obteve K = 0,5105, superando o método *bag of words* com K = 0,4873 e a medida do cosseno, com K = 0,5050. Isso significa que se houvesse a necessidade da criação de um índice unificado para nomes de autores com base unicamente nos títulos dos trabalhos, e se o objetivo fosse minimizar a fragmentação nos índices unificados, o método mais indicado seria o coeficiente de Jacard. Para PMG imediatamente superior a 0,9, o coeficiente de Jaccard obteve K = 0,3958, sendo novamente superior ao método *bag of words* com K = 0,3406 e à medida do cosseno com K = 0,3827. Isso significa que esse método também é o mais indicado para criar índices unificados baseado na fonte de evidência fornecida pelos títulos dos trabalhos quando a prioridade é a precisão dos grupos gerados. Estes resultados são inicialmente inferiores aos resultados obtidos para os índices unificados produzidos apenas com o módulo de pré-avaliação. Entretanto, a utilização concomitante das fontes de evidência aumenta a confiabilidade dos agrupamentos gerados e melhora os resultados iniciais. A partir da Tabela 3.4 podemos ainda perceber que apenas a utilização dos títulos dos trabalhos não é suficiente para a criação de bons índices unificados, sendo que a fragmentação avaliada através da medida PMA é alta (quanto menor PMA, maior a fragmentação). Isso se explica ao considerarmos que um pesquisador pode publicar trabalhos em diversas áreas, com títulos contendo palavras coincidentes entre trabalhos da mesma área e não coincidindo palavras em títulos de trabalhos de áreas diferentes.

A segunda fonte de evidência que avaliamos foi o veículo de publicação dos trabalhos. A Tabela 3.5 mostra os melhores resultados com essa fonte quando utilizamos coeficiente de Jaccard, medida do cosseno e *bag of words* para o cálculo da medida de similaridade.

Salientamos que na coleção de teste da BDBComp esta informação é disponibilizada através da sigla do evento, enquanto que na coleção da DBLP, a informação é escrita por extenso.

Método	Número de Grupos Gerados	PMG	PMA	K
Bag of Words (α = 3)	1853	0,7605	0,2587	0,4436
Bag of Words (α = 5)	2960	0,9178	0,1118	0,3204
Coeficiente de Jaccard (α = 0,25)	2204	0,9063	0,1728	0,3958
Medida do cosseno (α = 0,25)	1423	0,7435	0,2946	0,4680
Medida do cosseno (α = 0,4)	2072	0,9090	0,1453	0,3634

Tabela 3.5. Resultados da Utilização da Fonte de Evidência Veículo de Publicação na Geração de Índices Unificados para a coleção da DBLP.

Podemos perceber que, para essa fonte de evidência, o melhor resultado foi obtido com a medida do cosseno, quando se conseguiu K = 0,4680, para PMG imediatamente superior a 0,7. Novamente utilizamos esta faixa para avaliarmos qual seria a fragmentação dos conjuntos de trabalhos de cada autor presente no índice unificado caso fizéssemos um relaxamento da restrição imposta pela PMG em 0,9. O método *bag of words* obteve K = 0,4436 e o coeficiente de Jaccard obteve K = 0,3958, sendo que o valor imediatamente superior a 0,7 para PMG desse método foi 0,9063. Para PMG acima de 0,9, o coeficiente de Jaccard obteve o melhor K, com o valor 0,9063. O método *bag of words* obteve K = 0,3204 e a medida do cosseno, K = 0,3634. Repetimos que os resultados para essa fonte de evidência são, isoladamente, inferiores aos obtidos para os índices unificados criados apenas com o módulo de pré-avaliação. Isso se deve ao fato de estarmos impondo restrições adicionais, o que tende a gerar grupos com maior PMG e PMA reduzido. A fonte de evidência fornecida pelos veículos de publicação não deve ser usada isoladamente, pois gera índices unificados mais fragmentados que o desejado. A principal justificativa para esse fato é que um determinado autor pode publicar trabalhos em diversos eventos pertencentes a uma mesma área, ou a áreas diferentes, não obrigatoriamente havendo uma coincidência de palavras entre os nomes dos eventos de áreas diferentes.

A última fonte de evidência que avaliamos foram os nomes dos co-autores dos trabalhos. Utilizamos para o cálculo da similaridade dessa fonte de evidência a medida do cosseno, coeficiente de Jaccard, *bag of words* e comparação por fragmentos. Apresentamos na Tabela 3.6 os melhores resultados obtidos para essa fonte de evidência.

Método	Número de Grupos Gerados	PMG	PMA	K
Bag of Words (α = 2)	1536	0,7736	0,3516	0,5215
Bag of Words (α = 4)	3045	0,9306	0,1403	0,3613
Coeficiente de Jaccard (α = 0,15)	1285	0,7004	0,4020	0,5306
Coeficiente de Jaccard (α = 0,55)	2566	0,9750	0,1290	0,3546
Medida do cosseno (α = 0,1)	1302	0,7018	0,4004	0,5301
Medida do cosseno (α = 0,35)	1772	0,9107	0,2754	0,5008
Comparação por fragmentos (α = 4)	1328	0,7293	0,3650	0,5159
Comparação por fragmentos (α = 0,35)	1698	0,9139	0,2935	0,5179

Tabela 3.6. Resultados da Utilização da Fonte de Evidência Co-autores na Geração de Índices Unificados para a Coleção da DBLP.

Para PMG imediatamente acima de 0,7, o coeficiente de Jaccard atingiu o melhor K, 0,5306, sendo seguido pela medida do cosseno, com K = 0,5301, *bag of words*, com K = 0,5215 e por último a comparação por fragmentos, com K = 0,5159. Para PMG imediatamente acima de 0,9, a comparação por fragmentos atingiu o melhor K, 0,5179, ficando em segundo lugar a medida do cosseno, com K = 0,5, em terceiro lugar o método *bag of words*, com K = 0,36 e atingindo o pior K, 0,35, o coeficiente de Jaccard. Podemos observar que a utilização de nomes de co-autores como fonte de evidência para a criação de índices unificados gerou a menor fragmentação entre as fontes de evidência analisadas. A explicação é que um pesquisador pode publicar trabalhos em eventos diferentes com títulos diferentes, mas de um modo geral, alguns membros de seu grupo de pesquisa o acompanham na co-autoria desses trabalhos e, com isso, há um maior número de coincidências para essa fonte de evidência. Obviamente, trabalhos publicados individualmente por um autor não geram bons resultados para essa fonte de evidência.

Após determinarmos os melhores métodos e parâmetros isoladamente para cada uma das fontes de evidência, realizamos novos experimentos combinando-as. A similaridade entre os pares analisados passou a ser calculada considerando as combinações descritas na Tabela 3.7.

Como pode ser observado na Tabela 3.7, combinamos as fontes de evidência através dos conectivos lógicos “e” e “ou”. Para cada uma das fontes de evidência, ao utilizamos o conectivo lógico “e”, estávamos impondo uma nova restrição para a criação de novos grupos, pois, nesse caso, havia a necessidade de que a similaridade entre pelo menos duas fontes de

evidência fosse verificada simultaneamente, o que diminui o número de grupos gerados, ocasionando maior fragmentação. Para que um número maior de grupos pudesse ser gerado, utilizamos o valor para α que obteve melhor K para PMG imediatamente superior a 0,7. Por outro lado, quando utilizamos o conectivo lógico "ou", houve um relaxamento de restrições, sendo que a similaridade de apenas uma das fontes de evidência unidas pelo conectivo "ou" deveria ser verificada. Desse modo, utilizamos o método que obteve melhor K para PMG imediatamente superior a 0,9 para cada uma das fontes de evidência. A Tabela 3.8 mostra um resumo dos melhores resultados para cada fonte de evidência. Para exemplificar, suponhamos o cálculo da similaridade de um par de tuplas do índice de autoria pela combinação disposta na primeira entrada da Tabela 3.7, "co-autores e título". Nesse caso, de acordo com a Tabela 3.8, devemos utilizar como método para cálculo da similaridade o coeficiente de Jaccard, com α = 0,15, para co-autores e também o coeficiente de Jaccard, com α = 0,15, para títulos. Desse modo, para que um par seja associado gerando um grupo, deve alcançar o valor mínimo de α para as duas medidas de similaridade nos dois métodos.

Suponhamos agora o cálculo da similaridade de um par de tuplas do índice de autoria pela combinação disposta na quinta entrada da Tabela 3.7, "co-autores ou título". Nesse caso, utilizamos a combinação de métodos e parâmetros que obtiveram melhor K para PMG imediatamente superior a 0,9. A justificativa para esta escolha é que, quando utilizamos o conectivo "e", todas as fontes de evidência utilizadas devem alcançar o α mínimo de acordo com o método adotado, o que é mais difícil de ser conseguido do que quando utilizamos o conectivo "ou", que exige que apenas uma das fontes de evidência alcance o α mínimo de acordo com o método adotado.

Combinações de Fontes de Evidência
co-autores e título
co-autores e evento
título e evento
co-autores e título e evento
co-autores ou título
co-autores ou evento
título ou evento
co-autores ou título ou evento
co-autores e (título ou evento)
co-autores ou (título e evento)
título e (co-autores ou evento)
título ou (co-autores e evento)
evento e (título ou co-autores)
evento ou (título e co-autores)

Tabela 3.7. Combinações Possíveis para Fontes de Evidência.

	PMG > 0,7	PMG > 0,9
co-autores	Coeficiente de Jaccard – α = 0,15	Comparação por fragmentos – α = 0,35
título	Coeficiente de Jaccard – α = 0,15	Coeficiente de Jaccard – α = 0,25
evento	Medida do cosseno – α = 0,25	Coeficiente de Jaccard – α = 0,25

Tabela 3.8. Resumo dos Métodos que Obtiveram Melhor K para Cada Fonte de Evidência.

Realizamos experimentos para todas as combinações constantes na Tabela 3.7 utilizando a comparação por fragmentos como função de casamento de padrão do módulo de pré-avaliação e adotando duas estratégias diferentes de acordo com a característica dos nomes dos autores analisados. No primeiro caso, quando o módulo de pré-avaliação afirmou a compatibilidade de um determinado par de tuplas do índice de autoria, o cálculo de similaridade foi realizado independentemente de o nome do autor ser um nome curto ou não. No segundo caso, quando o módulo de pré-avaliação afirmou a compatibilidade entre um determinado par de tuplas do índice de autoria, verificamos os nomes dos autores. Quando pelo menos um dos nomes avaliados era curto, calculamos a similaridade entre o par e, se esta estava acima de um limite arbitrário, o par era associado gerando um novo grupo.

Método	Número de Grupos Gerados	PMG	PMA	K
co-autores e título	1995	0,8513	0,2601	0,4705
co-autores e evento	2329	0,8532	0,1821	0,3941
título e evento	2707	0,9474	0,1367	0,3599
co-autores e título e evento	3351	0,9872	0,0839	0,2878
co-autores ou título	828	0,7541	0,4978	0,6127
co-autores ou evento	419	0,5440	0,6030	0,5727
título ou evento	543	0,5529	0,5647	0,5588
co-autores ou título ou evento	332	0,5251	0,6401	0,5797
co-autores e (título ou evento)	1631	0,7413	0,3537	0,5120
co-autores ou (título e evento)	868	0,6963	0,5494	0,6185
título e (co-autores ou evento)	1496	0,8193	0,3598	0,5429
título ou (co-autores e evento)	1075	0,6923	0,4466	0,5561
evento e (título ou co-autores)	2324	0,9709	0,2052	0,4464
evento ou (título e co-autores)	519	0,5454	0,5900	0,5673

Tabela 3.9. Resultados Experimentais para o Uso Combinado de Fontes de Evidência Adotando a Primeira Estratégia para Tratamento de Pares com Nomes Curtos.

Método	Número de Grupos Gerados	PMG	PMA	K
co-autores e título	941	0,7277	0,5567	0,6365
co-autores e evento	1141	0,7295	0,4661	0,5831
título e evento	1417	0,8571	0,4688	0,6338
co-autores e título e evento	1836	0,9039	0,3990	0,6006
co-autores ou título	434	0,6271	0,6461	0,6365
co-autores ou evento	281	0,5038	0,6519	0,5731
título ou evento	325	0,5196	0,6264	0,5705
co-autores ou título ou evento	224	0,4908	0,6748	0,5755
co-autores e (título ou evento)	826	0,6564	0,5559	0,6041
co-autores ou (título e evento)	489	0,6392	0,6660	0,6525
título e (co-autores ou evento)	738	0,6843	0,5986	0,6400
título ou (co-autores e evento)	503	0,6085	0,6099	0,6092
evento e (título ou co-autores)	1274	0,8803	0,5059	0,6673
evento ou (título e co-autores)	308	0,5120	0,6478	0,5759

Tabela 3.10. Resultados Experimentais para o Uso Combinado de Fontes de Evidências Adotando a Segunda Estratégia para Tratamento de Pares com Nomes Curtos.

Se, de outro modo, nenhum dos nomes avaliados era curto, o par era associado, gerando um grupo sem o cálculo de sua similaridade. Apresentamos na Tabela 3.9 os resultados obtidos para os experimentos realizados de acordo com a primeira estratégia descrita acima e, na

Tabela 3.10, os resultados obtidos para os experimentos realizados de acordo com a segunda estratégia. Lembramos que o número total de grupos no índice unificado gabarito é 222. Entre os resultados das tabelas acima, para garantir a qualidade dos índices unificados gerados, estamos interessados apenas nas combinações que resultaram em PMG > 0,9. Desta forma, garantimos que o número de associações erradas de tuplas a grupos estará em um patamar aceitável. Podemos verificar nas tabelas que, independentemente da estratégia utilizada para o tratamento de nomes curtos, a combinação das fontes de evidência através do conectivo "e" gera índices unificados com PMG mais elevado e combinações com o conectivo "ou" geram índices unificados com PMG's mais baixos. Entre as combinações com PMG acima de 0,9, estamos interessados naquela em que a média geométrica K foi a mais elevada pois, com isso, garantimos que a fragmentação foi a mais baixa para a geração de índices unificados de qualidade. Para a primeira estratégia avaliada, a combinação com PMG acima de 0,9 que gerou o maior K foi "evento e (título ou co-autores)", com PMG = 0,9709 e K = 0,4464. Para a segunda estratégia avaliada, a combinação com PMG acima de 0,9 que atingiu o maior K foi "co-autores e título e evento", com PMG = 0,9039 e K = 0,6006. Entretanto, a combinação de "evento e (título ou co-autores)" obteve PMG = 0,8803 e K = 0,6673, a qual atinge um bom patamar para PMG e tem média K 11% melhor que a primeira combinação. Concluímos com isso que a segunda estratégia foi superior à primeira, atingindo PMG próximo de 0,9 e gerando ao todo 1050 grupos a menos que a primeira estratégia. Este resultado comprova que, quando temos à disposição nomes escritos por extenso, as demais fontes de evidência não precisam ser avaliadas.

3.3.3. Avaliação dos Resultados

Como última análise sobre a experimentação produzida, apontamos que a estratégia apresentada para a criação de índices unificados manteve resultados semelhantes nas duas coleções de teste. Na primeira coleção, em que os nomes em sua maioria aparecem escritos por extenso, o módulo de pré-avaliação utilizando a função de casamento de padrão por comparação por fragmentos obteve, sem nenhum cálculo de similaridade adicional, excelentes resultados, atingindo PMG = 0,9671, PMA = 0,9576 e K = 0,9612. Realizamos experimentos adicionais conferindo o impacto da utilização de fontes de evidência sobre esses resultados.

Observamos que para qualquer fonte de evidência utilizada, a PMG é incrementada em prejuízo da PMA e, conseqüentemente, da K. Isso se deve ao fato de que impomos uma nova restrição a ser satisfeita a cada vez que exigimos que uma nova fonte de evidência tenha sua similaridade verificada. No entanto, diante da qualidade dos resultados obtidos, novas restrições não são necessárias.

Para a segunda coleção de testes, devido ao percentual de nomes curtos, cerca de 52%, testamos duas estratégias diferentes para gerarmos índices unificados com base apenas na pré-avaliação. No primeiro caso, não associamos entradas do índice de autoria que contivessem nomes curtos, ainda que fossem considerados compatíveis pela pré-avaliação, o que favorece a criação de índices unificados com PMG elevada e também maior fragmentação. No segundo caso, associamos automaticamente todas as entradas do índice de autoria que contivessem nomes curtos e que fossem considerados compatíveis pelo módulo de pré-avaliação, reduzindo a PMG e, conseqüentemente, a fragmentação. Como esperado, a estratégia que gerou melhor PMG foi a primeira, com o módulo de pré-avaliação utilizando a função de casamento de padrão por comparação por fragmentos, obtendo PMG = 0,9274, PMA = 0,3212 e K = 0,5458. Apesar de os índices unificados gerados apenas com o módulo de pré-avaliação utilizando a função de casamento de padrão por distância de edição ter alcançado PMG um pouco superior, PMG = 0,9634, PMA = 0,3164 e K = 0,5521, preferimos a comparação por fragmentos, pela sua flexibilidade em reconhecer a similaridade entre nomes curtos e nomes escritos por extenso.

Com o objetivo de diminuirmos a fragmentação nos índices unificados gerados, realizamos outros experimentos sobre a segunda coleção de testes, mas dessa vez, utilizando também o módulo para cálculo de similaridades. Novamente desenvolvemos duas estratégias diferentes para tratarmos nomes curtos e nomes por extenso. No primeiro caso, o módulo para cálculo de similaridades verificou a similaridade entre todos os pares considerados compatíveis pelo módulo de pré-avaliação, independentemente de haver nomes curtos ou nomes por extenso nos pares, associando apenas aqueles considerados similares. No segundo caso, quando o módulo de pré-avaliação detectou a compatibilidade entre um par e este não continha nomes curtos, o par era associado gerando um novo grupo sem que a sua similaridade fosse calculada e, quando havia nomes curtos no par, o módulo para cálculo de similaridades determinava a sua similaridade, permitindo apenas a associação de pares que tinham sua

similaridade verificada. As duas estratégias se fizeram necessárias devido ao número elevado de nomes curtos contidos nessa coleção. Observamos através dos experimentos que a segunda estratégia gerou o melhor resultado obtido através da combinação das fontes de evidência "evento e (título ou co-autores)". Reduzindo o PMG de 0,9274 para 0,8803, melhoramos o PMA de 0,3212 para 0,5059 e K de 0,5458 para 0,6673. O resultado se explica ao observarmos que, em nossa primeira estratégia, tínhamos uma restrição que forçava a verificação da similaridade entre pares que tinham apenas nomes escritos por extenso e que já haviam sido considerados do mesmo grupo pelo módulo de pré-avaliação. Essa restrição fazia com que pares que realmente pertenciam ao mesmo grupo fossem considerados incompatíveis pelo módulo de cálculo de similaridades. Como tentativa para melhorar os resultados, nossa segunda estratégia verificava a similaridade entre as entradas do índice de autoria apenas quando uma delas continha nomes curtos. Essa segunda estratégia fez com que os resultados melhorassem, pois neste caso, o cálculo de similaridades era realizado apenas quando a informação presente nas cadeias de caracteres de nomes não era suficiente para se concluir se os pares comparados se referiam a obras de uma mesma pessoa. Obviamente o resultado obtido ainda contém alta fragmentação, visto que o número de grupos no índice unificado gerado manualmente era 222 e o número de grupos gerados por essa estratégia foi de 1274. No entanto, consideramos que esse é um resultado de boa qualidade, pois a coleção inicial continha 2270 nomes curtos, o que comprova que apesar de não conseguir remover totalmente a ambigüidade existente entre nomes de autores, nossa estratégia atinge resultados que podem ser utilizados na prática.

Algumas situações verificadas com pequena freqüência durante a experimentação merecem destaque. Observamos que algumas heurísticas poderiam ser criadas a partir de um estudo aprofundado dessas situações, fazendo com que os resultados mostrados anteriormente melhorassem. Como concluímos que nomes escritos por extenso eram evidência suficiente para que conjuntos de obras fossem considerados da mesma pessoa, uma exceção ocorria sempre que autores homônimos tinham seus nomes escritos por extenso. Nesse caso, seus conjuntos de trabalhos eram unidos. Para solucionarmos este problema poderíamos, intuitivamente, propor que a freqüência com que os nomes apareciam no repositório inicial fosse utilizada como evidência adicional. No entanto, observamos que, na prática, diversos

nomes e sobrenomes raros apareciam com freqüência maior que aqueles sobrenomes comuns, o que é justificado, principalmente, pela alta produtividade de tais autores.

Um segundo caso ocorria quando as iniciais de um autor estavam incorretas, ainda que seu sobrenome estivesse certo. Por exemplo, uma autora chamada Karina Nunes Silva poderia ter seu nome abreviado como K. Nunes Silva ou, erroneamente, como C. Nunes Silva. Neste caso, o conjunto de obras da autora ficaria dividido como se pertencesse a dois autores distintos. A solução intuitiva poderia ser comparar cada entrada do índice de autoria com todas as demais. No entanto, percebemos que, na prática, o número de ocorrências como esta era pequeno em relação ao custo para realizar todas as comparações.

Como terceiro e último caso, vale ressaltar aquele ocorrido quando a ordem dos nomes aparecia trocada, e sem indicadores, por exemplo, “Geraldo Silva” e “Silva Geraldo” ao invés de “Silva, Geraldo”. Este caso é ainda mais complicado, já que utilizamos em nossa estratégia a primeira inicial e o último sobrenome como base para definir se as comparações deveriam ou não ser feitas. Com isso, novamente o conjunto de trabalhos do autor seria considerado pertencente a duas pessoas diferentes. Como solução, poderíamos utilizar a combinação entre os nomes e sobrenomes para identificar quais pares seriam compatíveis, no entanto, o número de pares a serem comparados cresceria exageradamente, elevando novamente o custo da operação..

Capítulo 4

Conclusões e Trabalhos Futuros

A sobrecarga informacional provocada pelo crescimento desgovernado da Web tornou necessários sistemas que centralizam informações segundo domínios restritos. Bibliotecas digitais são sistemas projetados para reunir informações em domínios pré-determinados, conferindo fácil acesso a objetos de interesse. Uma biblioteca digital adquire seus dados através de fontes diversas, não contendo, portanto, padronização de seus dados. Dados não padronizados tornam-se ambíguos à medida que utilizamos a forma como cadeias de caracteres são escritas para associá-las a entidades do mundo real. Dessa forma, um mecanismo para a remoção de ambigüidades em bibliotecas digitais é a criação de um arquivo que reúna as formas correspondentes de cadeias de caracteres, de modo que se possa traçar uma correspondência entre essas cadeias e uma determinada entidade real.

Neste trabalho, estudamos a criação de índices unificados como forma de remoção de ambigüidades em cadeias de caracteres que representam nomes de pessoas dentro de bibliotecas digitais. Apresentamos para isso uma estratégia que utiliza técnicas de recuperação de informação e algoritmos de agrupamento, sendo dividida nas fases de pré-avaliação e cálculo de similaridades. A primeira tem a finalidade de reconhecer, a partir de uma função para casamento de padrão, a presença em elementos XML que representam objetos bibliográficos, de nomes que potencialmente representam a mesma pessoa. A segunda avalia, através de medidas utilizadas em recuperação de informação e utilizando informações adicionais, se os candidatos identificados pela fase de pré-avaliação efetivamente se referem à mesma pessoa.

Avaliamos a efetividade da estratégia proposta através de duas coleções de teste com características distintas. Para a primeira coleção, montada a partir do repositório da BDBComp

[Laender, Gonçalves & Roberto, 2004] e formada predominantemente por nomes de autores brasileiros, obtivemos PMG e PMA superiores a 90%. Para a segunda coleção, montada a partir da DBLP[17], onde havia a predominância de nomes incompletos, obtivemos PMG superior a 90% e PMA próximo a 50%. A partir desses resultados, podemos concluir que nossa estratégia é capaz de criar arquivos de autoridade para remover ambigüidades entre nomes de autores de forma segura, gerando conjuntos de trabalhos fragmentados em casos extremos.

Como trabalhos futuros, destacamos a possibilidade de utilização desta estratégia na BDBComp de forma que, a cada vez que um novo registro seja inserido, haja uma verificação automática dos autores aos quais o novo objeto bibliográfico possa pertencer. Para que outras bibliotecas digitais possam integrar o repositório da BDBComp a seus repositórios, torna-se necessária a disponibilização do índice unificado atualizado da BDBComp através de uma interface OAI disponibilizada pela biblioteca. No projeto OPUS da UFMG, poderia ser desenvolvida uma interface que, a partir de uma cadeia de caracteres curta fornecida pelo usuário, propusesse referências completas para co-autores já armazenados na base. Neste caso, o algoritmo poderia procurar um máximo de casamentos para oferecer a escolha mais ampla para os usuários, o que resultaria em uma minimização da fragmentação existente na base.

Novas experimentações devem ser realizadas para avaliar o efeito da utilização de outros métodos no cálculo da similaridade entre fontes de evidência. Por exemplo, a medida OKAPI, descrita por Fhang, Tao & Zhai [2004], pode ser utilizada para avaliar a similaridade entre as fontes de evidência. Em nossa experimentação, o grau de similaridade entre as fontes de evidência foi combinado de forma binária. Cada fonte de evidência foi considerada similar ou não de acordo com as medidas utilizadas. Uma alternativa é considerar as similaridades de forma não binária, verificando qual o grau obtido individualmente para cada fonte e associando esses resultados. Christen, Churches & Hegland [2004] desenvolveram um módulo para geração de bases artificiais que poderiam ser úteis em novas experimentações. Além disso, seria também interessante uma avaliação sobre o uso de *stemming* na fase de tradução.

Nesse trabalho, a estratégia desenvolvida foi aplicada apenas a nomes de autores. Entretanto, o arcabouço desenvolvido nesta dissertação permite que arquivos de autoridade sejam também construídos para objetos em diferentes domínios. Dentro do domínio

[17] http://www.informatik.uni-trier.de/~ley/db/

bibliográfico, poderíamos construir arquivos de autoridade para outros campos, como veículos de publicação. Um problema típico que pode ser solucionado por essa adaptação é o ocorrido no portal Qualis[18], da CAPES. Nesse portal, nomes de conferências e títulos de periódicos freqüentemente aparecem escritos de formas diferenciadas. É possível criar um arquivo de autoridade que reúna essas diferentes formas, solucionando as ambigüidades atuais.

Uma interface gráfica para usuários também poderia ser criada, permitindo ao usuário a aplicação de nossa estratégia a qualquer coleção desejada, além da modificação de alguns parâmetros do algoritmo. Outra possibilidade interessante seria avaliar o quanto o algoritmo melhoraria em PMA, PMG e K caso uma coleção de treinamento fosse utilizada. Por último, as exceções descritas ao final do Capítulo 3 poderiam ser estudadas para que novas heurísticas fossem criadas, solucionando esses problemas.

[18] http://qualis.capes.gov.br/

Referências Bibliográficas

Abiteboul, S.; Buneman, P. & Suciu, D. *Data on the Web: From Relations to Semistructured Data and XML*. Morgan Kaufmann , San Francisco, CA, 1999.

Ajmera, J.; Bourlard, H. & Lapidot, I. Unknown-multiple speaker clustering using HMM, In *Proceedings of the 7th International Conference on Spoken Language*, Denver, Colorado, USA, 2002, 573-576.

Al-Kamha, R. & Embley, D. W. Grouping search-engine returned citations for person-name queries. In *Proceedings of the Sixth ACM CIKM International Workshop on Web Information and Data Management*, Washington, DC, USA, November, 2004, 96-103.

Auld, L. Authority Control: An Eighty-Year Review. *Library Resources and Technical Services*, 1982, 26: 319-330.

Baeza-Yates, R. & Ribeiro-Neto, B. *Modern Information Retrieval*. Addison-Wesley, New York, 1999.

Bagga, A., Baldwin, B. Entity-Based Cross-Document Coreferencing Using the Vector Space Model. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics*. Montreal, Quebec, Canada, 1998, 79-85.

Bergman, M. K. The Deep Web: Surfacing Hidden Value. *Journal of Electronic Publishing*, 7(1), Agosto de 2001. Disponível em www.press.umich.edu/jep/07-01/bergman.html. Data de acesso: 15/04/03

Berkhin, P. *Survey of clustering data mining techniques*. *Technical report*, Accrue Software, San Jose, California, 2002. Disponível em http://citeseer.nj.nec.com/berkhin02survey.html. Data de acesso: 15/11/03.

Bowman, C. M.; Danzig, P. B.; Harky, D. R.; Manber, U. & Schwartz, M. F. The Harvest Information Discovery and Access System, *Computer Networks and ISDN Systems*, 28: 119-126, 1995.

Café, L. & Lage, M. B. Auto-arquivamento: uma opção inovadora para a produção científica. *Datagramazero*, 3 (3), 2002. Disponível em http://dici.ibict.br/archive/00000036, data de acesso: 11/04/2004.

Carvalho, J. C. P. & Silva, A. S. Finding similar identities among objects from multiple web sources. In *Proceedings of the Fifth ACM CIKM International Workshop on Web Information and Data Management*, New Orleans, Louisiana, USA, November, 2003, 90-93

Chakrabarti, S. *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan Kaufmann Publishers, San Francisco, CA, USA, 2002.

Christen, P., Churches, T. & Hegland, M. A Parallel Open Source Data Linkage System. In *Proceedings of the 8th Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Sydney, Autralia, 2004, 638-647.

Cohen, W. & Richman, J. Learning to match and cluster large high-dimensional data sets for data integration. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Alberta, Canadá, 2002, 23-26.

Cristianini, N. & Shawe-Taylor, J. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK, 2000.

Dayanik, A. & Nevill-Manning, C. G. Clustering in Relational Biological Data. In *Proceedings of the ICML-2004 Workshop on Statistical Relational Learning and its Connections to Other Fields*. Banff, Alberta, Canada, 2004, 49-54. Disponível em http://www.cs.umd.edu/projects/srl2004/Papers/dayanik.pdf. Data de acesso: 02/11/2004

Fair, M. Generalized Record Linkage System - Statistics Canada's Record Linkage Software. *Austrian Journal of Statistics*. Vienna, Áustria, 33(1,2): 37-55, 2004. Disponível em http://www.stat.tugraz.at/AJS/ausg041+2/041+2Fair.pdf. Data de acesso: 08/04/2005.

Fang, H.; Tao, T. & Zhai, C. A formal study of information retrieval heuristics. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Sheffield, UK, 2004, 49-56.

Fellegi, I. P. & Sunter, A. B. A Theory of Record Linkage. *Journal of the American Statistical Association*, 64: 1183–1210, 1969.

Frakes, W. & Baeza-Yates, R. *Information Retrieval: Data Structures & Algorithms*. Prentice Hall, Englewood Cliffs, New Jersey, 1992

French, J. C.; Powell, A. L. & Schulman, E. Using clustering strategies for Creating Authority Files. *Journal of the American Society for Information Science*, 51 (8): 774-786, 2000.

Gu, L., Baxter, R., Vickers, D. & Rainsford, C. *Record Linkage: Current Practice and Future Directions. Technical Report*, Commonwealth Scientific and Industrial Research Organisation, Mathematical and Information Sciences, Canberra, Australia, 2003. Disponível em http://citeseer.ist.psu.edu/585659.html. Data de acesso: 15/05/2005

Han, H.; Giles, C.L.; Zha, H.; Li, C. & Tsioutsiouliklis, K. Two supervised learning approaches for name disambiguation in author citations. In *Proceedings of the 4th ACM/IEEE-CS Joint Conference on Digital libraries*, Tucson, Arizona, USA, 2004, 296-305.

Han, H.; Xu, W.; Zha, H. & Giles, L. A Hierarchical Naive Bayes Model for Name Disambiguation in Author Citations. *In Proceedings of the 20th Annual ACM Symposium on Applied Computing Special Track on Information Access and Retrieval*, Santa Fé, New Mexico, USA, 2005, 1074-1078. Disponível em http://www.cse.psu.edu/~hhan/homepage/pub/hhan_SACIAR0021.pdf. Data de acesso: 10/03/2005

Knoblock, C. A.; Minton, S.; Ambite, J. L.; Ashish, N.; Muslea, I.; Philpot, A. & Tejada, S. The Ariadne Approach to Web-Based Information Integration. *International Journal of Cooperative Information Systems*, 10(1,2): 145-169, 2001.

Laender, A. H. F.; Gonçalves, M. A. & Roberto, P. A. BDBComp: Building a Digital Library for the Brazilian Computer Science Community. In *Proceedings of the Fourth ACM/IEEE Joint Conference on Digital Libraries*, Tucson, Arizona, USA, 2004, 23-24.

Lagoze, C.; Sompel, H. V. The Open Archives Initiative: building a low-barrier interoperability framework. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries*, Virginia, USA, 2001, 54-62. Disponível em www.openarchives.org. Data de acesso: 20/10/2003

Lapidot, I. *Self-Organizing-Maps with BIC for Speaker Clustering*. *IDIAP-Research Report-02-60*, Martigny, Switzerland, 2002.

Levenshtein, V. Binary codes capable of correcting spurious insertions and deletions of ones. *Problems of Information Transmission*, 1:8-17, 1965.

Ley, M. The DBLP Computer Science Bibliography: Evolution, Research Issues, Perspectives. In *Proceedings of String Processing and Information Retrieval, 9th International Symposium*, Lisbon, Portugal, 2002, 1-10. Disponível em http://link.springer.de/link/service/series/0558/bibs/2476/24760001.htm. Data de acesso: 14/02/2004.

Mann, G. S. & Yarowsky, D. Unsupervised Personal Name Disambiguation. In: *Proceedings of The Seventh Conference on Natural Language Learning*, Edmonton, Canada, 2003, 33-40. Disponível em http://cnts.uia.ac.be/conll2003/pdf/03340man.pdf. Data de acesso: 20/01/2005

Monge, A. E. & Elkan, C. An Efficient Domain-Independent Algorithm for Detecting Approximately Duplicate Database Records. In *Proceedings of the SIGMOD Workshop on Data Mining and Knowledge Discovery*, Tucson, Arizona, USA, 1997, 23-29.

Montez, C.; Pistori, J. & Willrich, R. Experiências na Implementação da Biblioteca Digital Multimídia RMAV/Florianópolis. In *Anais do II Workshop RNP*. Belo Horizonte, Brasil, 2000.

Disponível em www.rnp.br/wrnp2/2000/posters/bibliotecadigital.pdf. Data de acesso: 23/04/03.

Quinlan, J. R. Improved use of continuous attributes in c4.5. *Journal of Artificial Intelligence Research*, 1996, 4: 77-90. Disponível em http://citebase.eprints.org/cgi-bin/citations?id=oai:arXiv.org:cs/9603103. Data de acesso: 15/04/04.

Rabiner, L. R., A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77 (2): 257–286, 1989.

Rosell, M.; Kann, V. & Litton, J. E. Comparing Comparisons: Document Clustering Evaluation Using Two Manual Classifications. In *Proceedings of the International Conference on Natural Language Processing*, Hyderabad, India, 2004, 207–216 Disponível em http://www.nada.kth.se/~rosell/publications/papers/rosellkannlitton04.pdf. Data de acesso: 14/11/2004

Ristad, E. S. & Yianilos, P. N. Learning string edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20 (5): 522–532, 1998.

Salton, G. *Automatic text processing*. Addison-Wesley, Reading, Massachussetts, 1988

Silva, L.V. *Um Serviço de Auto-arquivamento de Publicações Cientificas Compatível com o Padrão OAI. Dissertação de Mestrado*, Departamento de Ciência da Computação, UFMG, Belo Horizonte, 2004.

Snyman, M. M. M. & Rensburg, M. J. Revolutionizing name authority control. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, San Antonio, TX, USA, 2000, 185-194.

Solomonoff, A.; Mielke, A.; Schmidt, M., & Gish, H. Clustering speakers by their voices. In *Proceedings of IEEE International Conference on Accoustics, Speech and Signal Processing*, Munich, Germany, 2: 757-760, 1998.

Sompel, H. V.; Lagoze, C. The Santa Fe Convention for the Open Archives Initiative, *D-Lib Magazine*, 2000, 6. Disponível em http://www.dlib.org/dlib/february00/vandesompel-oai/02vandesompel-oai.html. Data de acesso: 23/10/2003

Strehl, A.; Ghosh, J. & Mooney, R. J. Impact of similarity measures on web-page clustering. In *Proceedings of AAAI Workshop on AI for Web Search*, Austin, TX, USA, 2000, AAAI/MIT Press, 58-64.

Tejada, S.; Knoblock, C. A. & Minton, S. Learning object identification rules for information integration. *Information Systems*, 26 (8): 607-633, 2001.

Yianilos, P. *The LikeIt Intelligent String Comparison Facility*, *Technical Report* 97-093, NEC Research Institute, Princeton, NJ, USA, 1997. Disponível em: http://www.neci.nec.com/homep

ages/pny/papers/likeit/main.html. Data de acesso: 05/04/2004

Zhang, B.; Gonçalves, M. A. & Fox, E. A. An OAI-Based Filtering Service for CITIDEL from NDLTD. In *Proceedings of The 6th International Conference on Asian Digital Libraries*, Kuala Lumpur, Malaysia, 2003, 590-601.