Similaridade entre Objetos Localizados em Fontes de Dados Heterogêneas

Rubens Guimarães[1], Gustavo Zanini Kantorski[1]

[1]Curso de Sistemas de Informação – Universidade Luterana do Brasil (ULBRA)
Campus Santa Maria – Santa Maria – RS – Brasil

rubens.poa@gmail.com, gustavoz@cpd.ufsm.br

Abstract. *The structured and semi-strucutured data sources integration is a major challenge for the database area. The objective of this paper is to present a tool capable to achieve integration and enable the identification of duplicates in structured and semi-strucutured data sources.*

Resumo. *A integração de fontes de dados estruturadas e semi-estruturadas é um dos grandes desafios para a área de banco de dados. O objetivo deste trabalho é apresentar uma proposta de ferramenta para realizar a integração e permitir a identificação de duplicatas em fontes de dados estruturadas e semi-estruturadas.*

1. Introdução

Atualmente com a expansão da internet, crescimento da disponibilidade e da demanda por informação, vem surgindo cada vez mais a necessidade de integrar dados de organizações distintas e permitir o acesso integrado a múltiplas fonte de dados. Estas fontes geralmente são heterogêneas, autônomas e distribuídas e que necessitam ser integradas para que a informação de diferentes setores de uma mesma organização, utilizando diferentes sistemas com grande redundância de dados e operações, torne-se algo limpo e transparente para o usuário.

Muitos problemas surgem quando são necessárias integrações de informações de várias fontes na web [Wiederhold 1993]. Um desses problemas é a existência de objetos em vários formatos, entre eles o XML. Dados XML são semi-estruturados e são organizados hierarquicamente. O formato XML torna complexa a tarefa de identificação de objetos, comparada com técnicas que tratam com fontes estruturadas tais como bancos de dados relacionais. Dados XML possuem estruturas diferentes e hierarquias que complicam a identificação dos objetos.

Este trabalho apresenta o desenvolvimento de uma ferramenta web, de código fonte aberto, cujo principal objetivo é realizar a identificação de similaridades de dados providos de documentos XML. A ferramenta proposta é parte do projeto denominado CORIDORA, desenvolvido em âmbito acadêmico na Universidade Luterana do Brasil, campus Santa Maria. O projeto CORIDORA tem como objetivo realizar o tratamento de inconsistências, e possíveis limpezas de dados, em bancos de dados, derivadas da representação de equivalências de um mesmo objeto do mundo real. O tratamento de inconsistência é realizado através do mapeamento de esquemas conceituais, identificando, consistindo e comparando divergências entre os objetos equivalentes, sem prejudicar a autonomia local das fontes de dados conforme proposta de [Ribeiro 1995].

A ferramenta que realiza o mapeamento de esquemas entre as fontes de dados heterogêneas, por meio da metodologia proposta por [Ribeiro 1995], está descrita nos trabalhos de [Meneghetti, Paes e Kantorski 2007a]. O acesso às fontes de dados e o resultado da consulta podem ser visualizados no trabalho de [Paes 2008]. Uma limitação na ferramenta desenvolvida por [Paes 2008] é a identificação no resultado da consulta de dados similares que existem em diferentes fontes. O objetivo deste artigo é apresentar uma solução para tratar o resultado da consulta realizada por [Paes 2008] através da identificação de similaridades entre documentos XML.

A próxima seção apresenta a ferramenta que realiza a consulta integrada nas fontes de dados heterogêneas. Na seção 3 é apresentada a proposta para resolver o problema resultante da consulta. Trabalhos relacionados são mostrados na seção 4. A seção 5 apresenta as considerações finais e trabalhos futuros.

2. Ferramenta de Consulta Integrada

Esta ferramenta baseia-se nos resultados obtidos durante os processos mapeamento de esquemas conceituais e identificação de equivalências semânticas, identificados nos trabalhos de [Meneghetti, Paes, Kantorski 2007a], [Meneghetti, Paes, Kantorski 2007b], [Meneghetti, Paes, Kantorski 2008], efetuados pelo ambiente Coridora para proporcionar a integração dos dados sem a necessidade da interação do usuário para realizar este processo.

O usuário deve escolher a equivalência que deseja consultar e então a ferramenta provê uma interface uniforme de acesso aos dados, de tal forma que abstraia a localização, conflitos semânticos ou até mesmo linguagem de consulta [Paes 2008]. Nesta interface o usuário deve informar os filtros que deseja fazer para sua consulta e a ferramenta analisa as informações adquiridas, onde novas consultas são geradas para, posteriormente, serem executadas nas diversas fontes de dados. A figura 1 ilustra o resultado da consulta.

Uma das dificuldades encontradas nessa etapa, é que a ferramenta tem a capacidade de determinar os objetos que são equivalentes, porém não é capaz de determinar quais objetos representam uma mesma entidade, retornando assim dados redundantes contidos nas diferentes fontes de dados selecionadas. Isto pode ser observado na figura 1 para a coluna “nome”. A consulta realizada para o nome “Rubens” pode retornar a mesma pessoa em fontes de dados diferentes.

3. Similaridade entre Objetos

Este trabalho tem por objetivo identificar objetos equivalentes providos do resultado da ferramenta de consulta integrada proposta por [Paes 2008] através da similaridade dos valores, calculada através da definição de pesos para os atributos e da utilização de algoritmos de similaridade. Os algoritmos são definidos por [Suder e Dornelles 2006] como funções pré-definidas que procuram identificar equivalências entre tipos de dados atômicos.

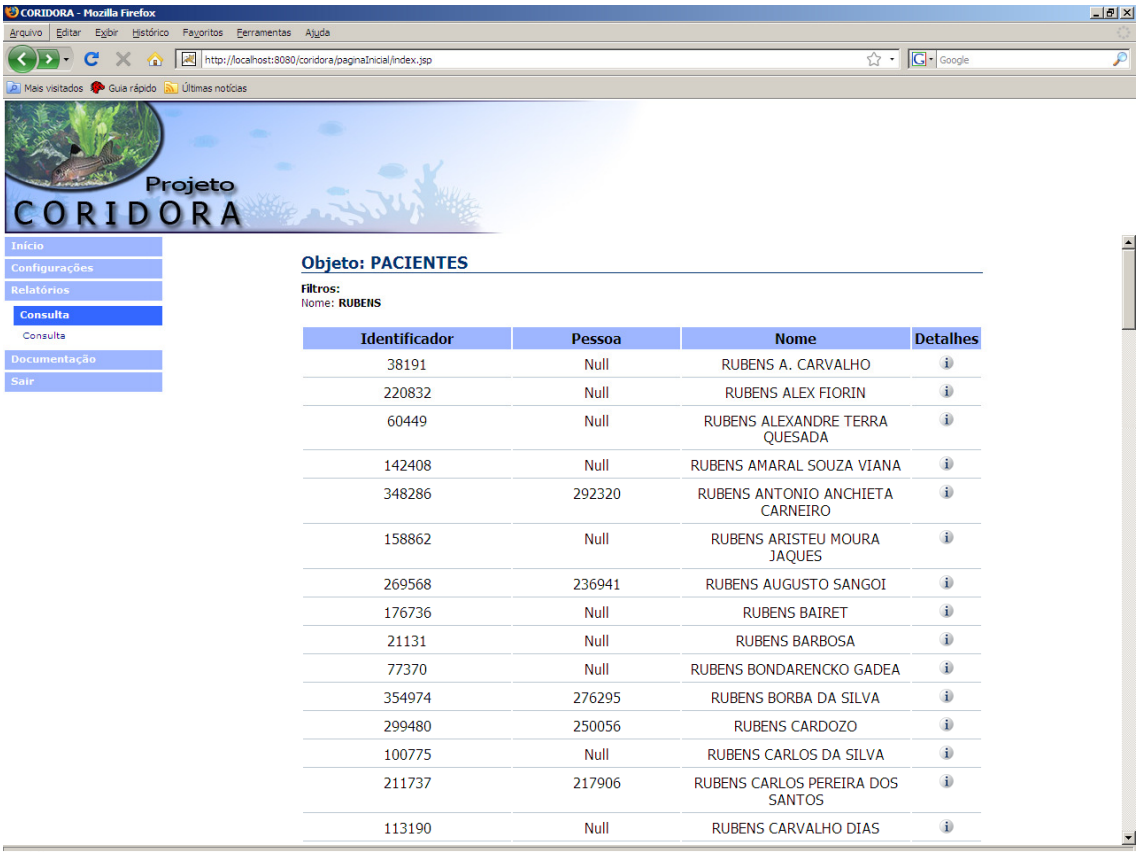

Figura 1. Interface de Consulta Integrada.

Um padrão para estruturar documentos é o XML (*eXtensible Markup Language*), proposta pelo W3C como uma linguagem de marcação textual cuja e tem sido aplicada para interoperabilidade, integração, estruturação e armazenamento de informações [W3C 2009]. Esta linguagem oferece uma abordagem para descrição, processamento e publicação de informações representadas por conteúdo, estrutura e apresentação. Desta forma, documentos XML são considerados coleções de documentos textuais com *tags* adicionais e relacionamentos entre as *tags*. A ferramenta proposta trabalha com fontes de dados XML validado pelo XSD (XML *Schema Definition*) descrito na Figura 2.

O arquivo XML (Figura 2) é composto por dois elementos que representam dois objetos providos de fontes de dados diferentes. Cada objeto contém um conjunto de elementos que representam seus atributos. Para cada atributo é necessário um identificador que será utilizado para relacionar os atributos equivalentes nos dois objetos, um nome que é utilizado como descrição no momento onde são exibidos os dados, um peso que é utilizado pela ferramenta para definir a relevância de cada atributo no processo de comparação e por fim um elemento que contém o conjunto dos valores de cada atributo.

```xml
<?xml version="1.0" encoding="utf-8"?>
<xs:schema targetNamespace="http://tempuri.org/XMLSchema.xsd" elementFormDefault="qualified"
  xmlns="http://tempuri.org/XMLSchema.xsd" xmlns:mstns="http://tempuri.org/XMLSchema.xsd"
  xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Teo" type="TeoType"/>
  <xs:complexType name="TeoType">
    <xs:sequence>
      <xs:element name="Objeto1" type="ObjectType" />
      <xs:element name="Objeto2" type="ObjectType" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="ObjectType">
    <xs:sequence>
      <xs:element name="Atributo" type="AtributoType" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="AtributoType">
    <xs:sequence>
      <xs:element name="id" type="xs:int" />
      <xs:element name="nome" type="xs:string" />
      <xs:element name="peso" type="xs:float" />
      <xs:element name="valores" type="xs:string" maxOccurs="unbounded" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

Figura 2. XSD do arquivo XML.

A figura 3 mostra uma representação do documento XML através de uma árvore. O elemento TEO representa a Tabela de Equivalência de Objetos [Meneghetti, Paes, Kantorski 2007a] que contém quais objetos são equivalentes. O resultado da consulta apresentado na figura 1 somado às informações dos metadados registrados no ambiente CORIDORA resulta no documento XML que será utilizado para a identificação de similaridade entre os objetos.

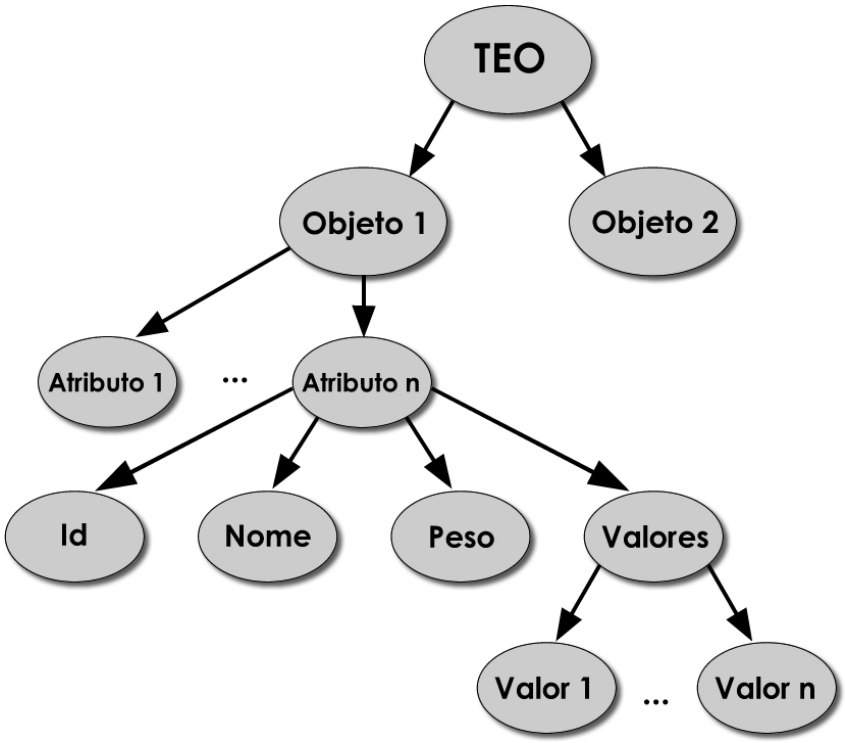

Figura 3. Arquivo XML representado em uma árvore.

O padrão para acesso e processamento de documentos XML é o XML DOM (*Document Object Model*). DOM representa elementos, atributos e textos como nós de uma árvore. Com a API DOM é possível processar um documento XML, iniciando pelo elemento raiz e navegando nas árvores nos demais elementos pais e filhos. Além da API DOM existe a API denominada SAX que permite a manipulação de documentos XML.

Com o documento XML criado, o usuário precisa informar apenas a similaridade referente à probabilidade com que deseja que os dados sejam equivalentes conforme a figura 4. Ao clicar em "Consultar" a ferramenta importa esses dados em formato de árvore através da API DOM e um *hashmap* de vetores é criado através do elemento 'Objeto1' onde cada posição contém um vetor com os dados de cada atributo. O vetor é acessado através do identificador definido no arquivo XML e a partir deste *hashmap* os objetos referentes ao elemento "Objeto1" são montados.

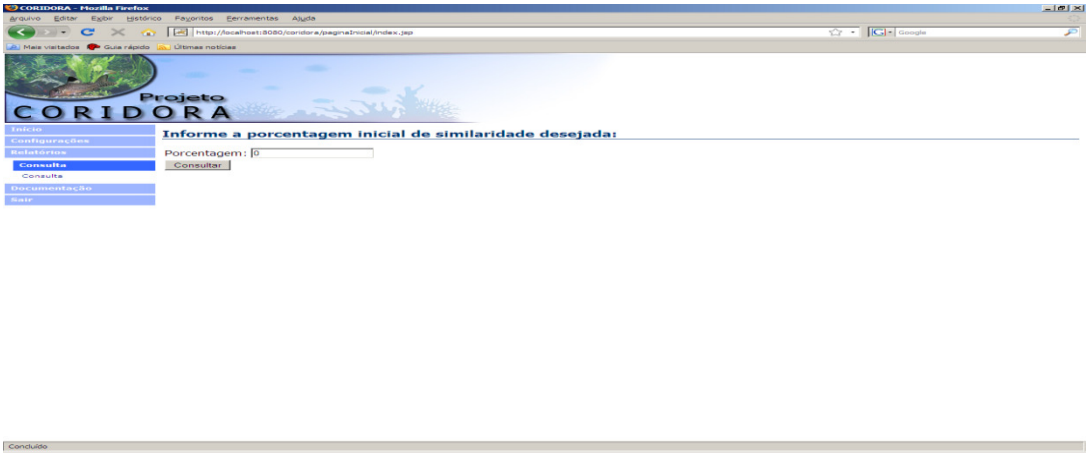

Figura 4. Interface onde deve ser informada a similaridade

Com os objetos criados, para cada valor contido nos atributos do elemento 'Objeto2' é montado um segundo objeto que é utilizado na execução de um processo de comparação que consiste em calcular uma similaridade utilizando-se algoritmos de similaridade definidos pela ferramenta. Foram escolhidos alguns algoritmos através de um estudo realizado levando em consideração aqueles mais citados na literatura [Chapman 2005].

Para atributos com valores numéricos e que possuem apenas um caractere a similaridade é 0 ou 1, onde 1 significa que são idênticos. Para os demais tipos atributos o algoritmo de similaridade calcula o valor *v*. Esse valor então é multiplicado pelo peso *p* definido para o atributo em questão. O processo é repetido até que todos os atributos estejam calculados, os valores obtidos são somados e a porcentagem de similaridade entre os dois objetos é calculada através da soma dos pesos. A fórmula na figura 5 descreve o cálculo da similaridade entre dois objetos:

$$sim(o_i, o_j) = \left(\frac{(\sum_{k=1}^{n} \mathrm{v}k * \mathrm{p}k)}{\sum_{k=1}^{n} \mathrm{p}k} \right)$$

Figura 5. Fórmula para cálculo da similaridade entre dois objetos

Onde *n* representa o número total de atributos equivalentes para os dois objetos, *v* representa o valor obtido com o cálculo de similaridade entre os dois valores e *p* representa o peso definido pelo projetista para o atributo *k*. O valor de *p* considera a

importância do atributo no conjunto de todos os atributos presentes no elemento *Atributo* do arquivo XML.

A tabela 1 descreve dois objetos equivalentes e provenientes de fontes de dados distintas que representam uma mesma pessoa, com seus atributos e dados. Para efeitos de comparação considere o objeto1 como Paciente e o objeto2 como Funcionário. O cálculo de similaridade entre os dois objetos é realizado da seguinte maneira: ((0*0) + (9*0.8) + (1*0.15) + (10*0,86)) / (0+9+1+10) = 0,798.

Tabela 1. Objetos distintos com dados equivalentes

	Id	**Data Nascimento**	**Profissão**	**Nome**
Paciente	258	25/05/1975	Estudante	Adalberto C. Carvalho
Funcionário	325	25/05/75	Desenvolvedor	Adalberto Cruz Carvalho
Peso	0	9	1	10
Similaridade (*strings*)	0	0,8	0,15	0,86
Algoritmo	-	Levengshtein	Levengshtein	Smith-Waterman
$Sim(o_i, o_j) = 0{,}798$				

Pode ser observado que mesmo quando a maior parte dos atributos possui valores consideravelmente diferentes, ainda assim, com a utilização de pesos é possível identificar a equivalência dos dados, pois o atributo "Nome" juntamente com o atributo "Data Nascimento" com o maior peso dentre os demais, é mais conveniente para identificar uma mesma pessoa mesmo quando em contextos diferentes.

A interface de exibição dos dados apresentada na figura 6 mostra para cada processo de comparação os dados originais, o algoritmo de similaridade utilizado, o valor obtido através deste, o peso de cada atributo e o percentual de similaridade calculada para os dois objetos.

É importante verificar que a similaridade é calculada entre objetos e não entre atributos. Embora os algoritmos sejam aplicados nos atributos dos objetos, a similaridade considera o peso de cada atributo no objeto mais a similaridade entre os atributos para calcular a similaridade global entre os objetos.

O processo de seleção do algoritmo de similaridade, que é aplicado nos valores dos atributos textuais, atualmente utiliza aqueles contidos no pacote *SimMetrics* [Chapman 2005]:

- Levengshtein – Este algoritmo pode ser parafraseado como "o menor número de inserções, remoções e substituições para igualar duas strings" [Navarro 2001]. São definidos escores diferentes para cada possível operação: *match* (casamento, igualdade dos caracteres); *mismatchs* (substituições); inserções, remoções. Onde são avaliadas todas as operações na tentativa de chegar ao maior escore. Este algoritmo demonstrou melhor resultado para comparações onde as strings possuem quantidades de caracteres semelhantes.
- Smith-Waterman – Este algoritmo é bastante utilizado para realizar alinhamentos locais de seqüências, isto é, determina regiões semelhantes entre as seqüências de caracteres existentes na string, e compara segmentos de todos os possíveis comprimentos e aperfeiçoa a semelhança medida para atingir o maior

escore. Este algoritmo demonstrou melhor resultado para as comparações quando as strings são compostas por mais de uma palavra.

- Jaro-Winkler – Este algoritmo variante do *Jaro Distance Metric* e é utilizado principalmente na área de *record linkage* (detecção de duplicidades). Esta extensão modifica os pesos dos pares identificados que partilham de um prefixo comum, porém não possuem um bom alinhamento. Demonstrou melhor resultado para as comparações quando a string é composta de uma palavra e um caractere, normalmente como acontece nas abreviações.

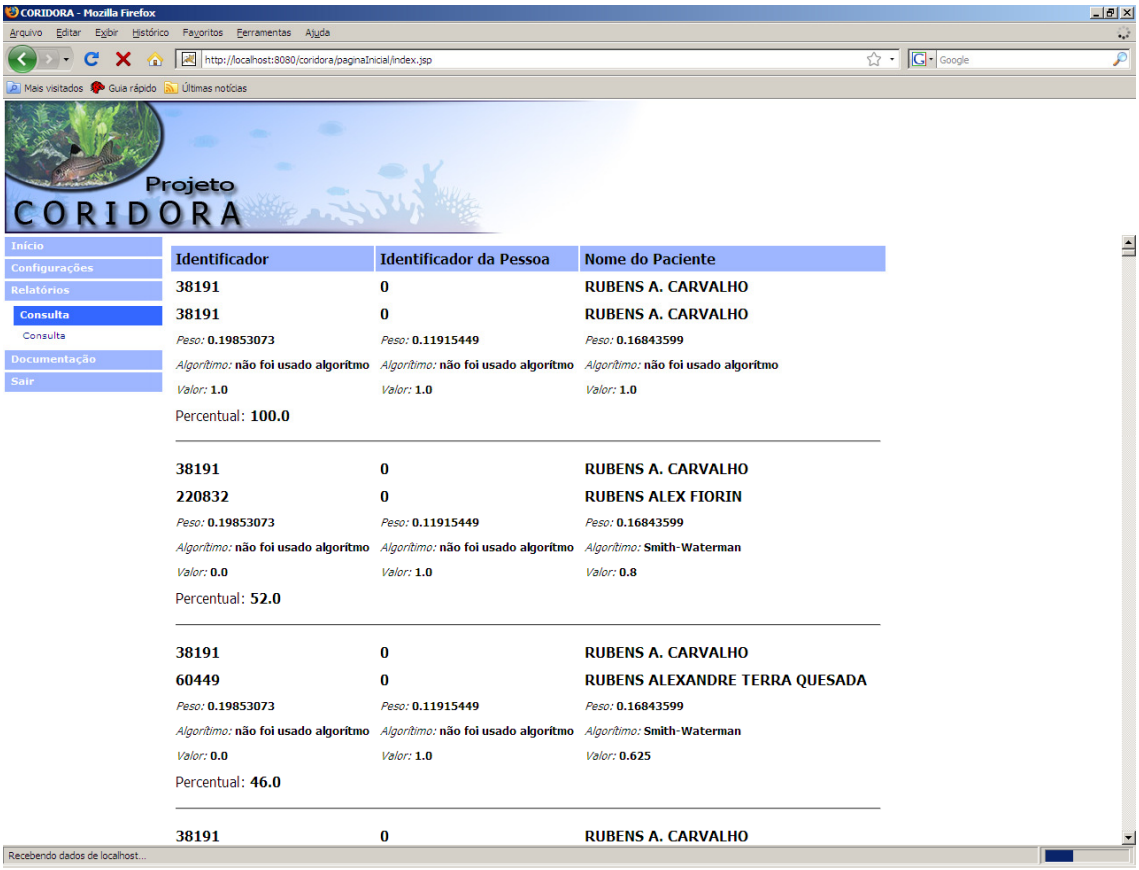

Figura 6. Interface de exibição das comparações

4. Trabalhos Relacionados

Vários trabalhos mostram o interesse da comunidade científica em explorar informações localizadas em fontes heterogêneas, sejam elas estruturadas, não estruturadas ou semi-estruturadas.

O trabalho *Duplicate Record Detection: A Survey* [Elmagarmid 2007] que consiste em uma pesquisa sobre algumas técnicas existentes para a busca de duplicatas em bancos de dados. Este trabalho parte da análise da heterogeneidade léxica, não se preocupando com a heterogeneidade estrutural, ou seja, analisa os dados partindo do princípio em que as estruturas são equivalentes. Neste artigo conforme descrito na seção 3, os dados provenientes de qualquer fonte de dados seja ela estruturada ou semi-estruturada, precisam estar disponibilizados em formato XML para que seja possível o cálculo da similaridade entre os objetos.

O trabalho de [Tejada 2001] apresenta um sistema de identificação de objetos chamado *Active Atlas* que aprende regras de mapeamento para um domínio específico de aplicação para determinar os mapeamentos dos objetos. O objetivo do trabalho proposto por [Tejada 2001] é aumentar a possibilidade de identificação de objetos com a participação mínima do usuário.

Trabalhos que envolvem a integração de documentos semi-estruturados e a sua heterogeneidade estrutural pode ser citado o *Structure-based inference of xml similarity for fuzzy duplicate detection* [Leitão 2007] onde baseado em conceitos de lógica *fuzzy*, propõe uma metodologia para identificar mesmas entidades com estruturas diferentes dentro de arquivos no padrão XML. Esta metodologia visa lidar com os dados em árvore e não somente identificar as duplicatas nos nós filhos, mas também calcular através de redes *Bayesianas*, as probabilidades dos nós descendentes também serem duplicados.

5. Considerações Finais e Trabalhos Futuros

Este trabalho apresentou uma forma de solução do problema de redundância de informações geradas no resultado do acesso integrado em fontes de dados heterogêneas. Desta forma, quando uma busca é realizada nas diversas fontes é possível unificar informações similares de fontes diferentes por meio da aplicação de algoritmos de similaridade. A similaridade de um objeto é calculada baseada em um peso, previamente definido, para cada atributo que o compõe e pelo valor assumido pelo atributo. Para um mesmo atributo (equivalente entre dois objetos) são comparados os seus respectivos valores e determinada a semelhança entre eles através de uma função. É importante salientar que a similaridade não é calculada entre os atributos de um objeto e, sim, entre os objetos. Isto é possível porque é realizada a avaliação de todos os atributos dos objetos.

Atualmente os pesos dos atributos dos objetos são definidos pelo projetista responsável pelo mapeamento das fontes no ambiente CORIDORA, ou seja, o projetista necessita de conhecimento sobre o esquema para aumentar a exatidão dos resultados. Técnicas como aprendizagem de máquina e descoberta de conhecimento podem ser aplicadas para verificar a possibilidade de determinar os pesos dos atributos de maneira semi-automática ou automática, diminuindo a participação do projetista.

Deve ser realizada uma avaliação da solução proposta considerando fontes com grande quantidade de dados para verificar questões relativas a desempenho, revocação e precisão nos resultados.

Referências

Chapman, Sam. (2005) “String Similarity Metrics for Information Integration”, In: Natural Language Processing Group, Department of Computer Science, University of Sheffield, Sheffield, UK.

Elmagarmid A. K., Ipeirotis, P. G., Verykios V. S. (2007) “Duplicate Record Detection: A Survey” The IEEE Transations on knowledge and Data Engineering (TKDE) Vol. 19 No. 1 January 2007, pp. 1-16.

Leitão, L., Pável, C., Weis M. (2007) “Structure-based inference of xml similarity for fuzzy duplicate detection”, In: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management - Lisboa, Portugal.

Meneghetti, F. B., Paes, F. G., Kantorski, G. Z. (2007a) “CORIDORA Mapping: Uma Ferramenta Web para Mapeamento de Equivalências Semânticas em Bancos de Dados Heterogêneos”. In: Simpósio de Informática, 2007, Uruguaiana – RS. XII Simpósio de Informática, Nov.

Meneghetti, F. B., Paes, F. G., Kantorski, G. Z. (2007b) “Ferramenta CORIDORA Mapping para Mapeamento de Esquemas em Bancos de Dados Heterogêneos”. In: Seminário de Informática, Torres – RS. VII Seminário de Informática, Nov.

Meneghetti, F. B., Paes, F. G., Kantorski, G. Z. (2008) “Uma Interface Web para Identificação de Equivalências em Bancos de Dados Heterogêneos”. In: Escola Regional de Banco de Dados. Florianópolis –SC, 2008

Navarro, G. (2001) “A Guided Tour to Approximate String Matching”. University of Chile. ACM Computing Surveys, Vol. 33, No. 1, Março 2001, pp. 31-88.

Paes, F. G. (2008) “Consulta Integrada a Bancos de Dados Heterogêneos na Web”. In: Trabalho de Conclusão de Curso, ULBRA, 2008.

Ribeiro, Cora Helena Francisconi Pinto. (1995) “Banco de Dados Heterogêneos: Mapeamento dos Esquemas Conceituais em um Modelo Orientado a Objetos (CPGCC)”. Porto Alegre: UFRGS, 1995. 165p.

Suder, R. L., Dornelles, C. F. (2006) “Integração de Dados em Múltiplos Níveis”. In: Escola Regional de Banco de Dados. Passo Fundo – RS, 2006.

Tejada, S., Knoblock, C.A., Minton, S. (2001) “Learning object identification rules for information integration”. Information Systems, Vol. 26, No 8, pp 607-633, 2001.

W3C, (2009) “Extensible Markup Language (XML)”, http://www.w3.org/XML Dezembro 200.

Wiederhold, G. (1993) “Intelligent integration of information” SIGMOD Record (1993), 434-437.