UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

Karine B. de Oliveira

NAZCA: UM MÉTODO DE SIMILARIDADE BASEADO NO CONTEXTO PARA MELHORIA DO CASAMENTO DE ESTRUTURAS HETEROGÊNEAS

Florianópolis

2014

Karine B. de Oliveira

NAZCA: UM MÉTODO DE SIMILARIDADE BASEADO NO CONTEXTO PARA MELHORIA DO CASAMENTO DE ESTRUTURAS HETEROGÊNEAS

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação para a obtenção do Grau de Mestre em Ciência da Computação.
Orientador: Prof. Dra. Carina F. Dorneles

Florianópolis

2014

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Oliveira, Karine Barbosa de
Nazca: Um método de similaridade baseado no contexto para melhoria do casamento de estruturas heterogêneas / Karine Barbosa de Oliveira ; orientadora, Carina Friedrich Dorneles - Florianópolis, SC, 2014.
66 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Ciência da Computação.

Inclui referências

1. Ciência da Computação. 2. Casamento em nível de estrutura. 3. Busca por similaridade. 4. Informações do contexto. 5. Escore de similaridade. I. Dorneles, Carina Friedrich. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Ciência da Computação. III. Título.

Karine B. de Oliveira

NAZCA: UM MÉTODO DE SIMILARIDADE BASEADO NO CONTEXTO PARA MELHORIA DO CASAMENTO DE ESTRUTURAS HETEROGÊNEAS

Esta Dissertação foi julgada aprovada para a obtenção do Título de "Mestre em Ciência da Computação", e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Florianópolis, 26 de fevereiro 2014.

Prof. Ronaldo dos Santos Mello, Dr.
Coordenador do Curso

Prof. Dra. Carina F. Dorneles
Orientador

Banca Examinadora:

Prof. Ronaldo dos Santos Mello, Dr.
Presidente

Prof.ª Patrícia Vilain, Dr.ª

Prof. Roberto Willrich, Dr.

Prof.ª Viviane Pereira Moreira, Dr.ª

A Deus, minha família e ao Fernando com amor.

AGRADECIMENTOS

Agradeço a Deus, autor da minha vida e dos meus sonhos, inclusive este. Obrigada por alinhar os meus pensamentos e por ser a inspiração para continuar. Pela sabedoria encontrada na maior fonte de conhecimentos acessível ao ser humano, a Sua Palavra.

Obrigada aos meus pais, que mesmo sem compreender, apoiaram o meu trabalho e intercederam por mim em cada dificuldade. Obrigada a todos os familiares que de alguma forma torceram para essa conclusão. Ao Fernando, meu amor, o que dizer para agradecer por tanto apoio, por tantas palavras motivadoras, por tantos abraços e ideias que me influenciaram e fizeram deste trabalho o que é. O seu amor que me fez compreender que vale a pena acreditar e prosseguir, te amo e obrigada por fazer parte de tudo isso (e por ajudar a corrigir o inglês dos meus artigos).

Não poderia deixar de agradecer aos meus amigos e colegas que contribuíram com conhecimento, diversão e apoio. Até as meninas que não são da área e não entendem muito bem porque tanta preocupação, obrigada por todos os recadinhos e apoio em oração, vocês também foram muito importantes nessa caminhada.

À minha querida orientadora, professora Carina, obrigada por acreditar em mim e por toda a ajuda. Por corrigir meus artigos nos finais de semana ensolarados, por falar comigo no *Skype*, *Whatsapp* e em todos os meios de comunicação que puderam atrapalhar seus momentos especiais, mesmo em viagem. Obrigada por contribuir com os seus conhecimentos e boas ideias e também obrigada por apoiar as minhas. Aos demais professores que fizeram parte desta etapa, muito obrigada.

Nenhuma grande descoberta foi feita jamais sem um palpite ousado.

Isaac Newton

RESUMO

O casamento de esquemas em nível de estrutura é um processo que pode ser aplicado em diversas áreas que envolvem a manipulação de dados heterogêneos. A ideia principal é casar elementos de estruturas que podem ser encontradas em diferentes fontes de dados, como por exemplo, elementos XML, classes de objetos, tabelas relacionais, *web forms* entre outras. Este processo é considerado um desafio devido ao grande número de representações heterogêneas de estruturas semanticamente similares. Neste trabalho, descreve-se um método de casamento de esquemas em nível de estruturas aplicado em um processo de busca. O objetivo é utilizar não só a própria estrutura no processo de casamento, mas também dados adicionais armazenados nas fontes de dados, que podem ser suficientemente representativos para caracterizar a estrutura. Estes dados podem ser chamados de "informações contextuais" e servem como base para ajustar o escore final de similaridade entre a estrutura da consulta e as estruturas encontradas nas fontes de dados. O método proposto é composto pelos seguintes componentes: i) funções de similaridade atômicas para elementos do esquema; ii) algoritmo para detecção das informações contextuais; e iii) árvore de decisão para o ajuste final de similaridade. Foram realizados experimentos que demonstram a efetividade do método com melhoria da precisão em relação ao algoritmo usado como *baseline*.

Palavras-chave: Casamento em nível de estrutura. Escore de similaridade. Informações do contexto. Busca por similaridade. Índice.

ABSTRACT

Structure-level matching is an important matching operator in various applications areas involving heterogeneous data. The main ideia is to matching combinations of elements that appear together in a structure, which can be found in different data models such as XML elements, object classes, relational tables, structures of web forms, and so one. This is a challenge due to large number of distinct representations of structures semantically similar. In this work, we describe a structure-level matching method developed to search for structures representations in data sources, taking into account the similarity score between structure elements and its context. The main goal is to use any internal information stored in the data source as context beyond the structure information, which can be representative enough to characterize the structure representation itself, for adjusting the similarity score between structure elements. The proposed method consists of the following components: i) atomic similarity functions to schema elements; ii) detection algorithm of contextual information; e iii) decision tree for final similarity score adjusts. We also present experiments showing the effectiveness of our method.

Keywords: Structure level matching. Similarity score. Context Information. Similarity search. Index.

LISTA DE FIGURAS

Figura 1 Classificação de abordagens para casamento de esquemas (RAHM; BERNSTEIN, 2001) 27
Figura 2 Representação de uma estrutura inserida no contexto: Job - Web Form. 31
Figura 3 Representação de uma estrutura inserida no contexto: Job - XML. 32
Figura 4 Exemplo de consulta com contexto e representação da estrutura 33
Figura 5 Tabela comparativa com outras abordagens semelhantes 41
Figura 6 Visão Geral da Proposta 43
Figura 7 Exemplo de Índice de Contexto (CI) 44
Figura 8 Motor de Busca por similaridade 45
Figura 9 Exemplo Similaridade Básica 47
Figura 10 Grafo de Contexto 49
Figura 11 Árvore de Decisão 52
Figura 12 Avaliação dos termos de CI 58
Figura 13 Preparação dos dados para execução do algoritmo Similarity Flooding 59
Figura 14 Visão geral do algoritmo Similarity Flooding (MELNIK; GARCIA-MOLINA; RAHM, 2002) 60
Figura 15 Revocação X Precisão - Método Nazca $\delta f'$ e Similaridade Básica α 61
Figura 16 Método Nazca X Similarity Flooding 62

LISTA DE TABELAS

Tabela 1 Armazenamento de ε 48
Tabela 2 Precisão da Árvore de Decisão 53
Tabela 3 Termos por Domínio 57

LISTA DE ABREVIATURAS E SIGLAS

XML	Extensible Markup Language	29
EI	Índice de Elementos	43
CI	Índice de Contexto	43

LISTA DE SÍMBOLOS

e	Estrutura composta por elementos armazenados em EI	46
D	Documento	46
t	termo	46
Co	Vetor de termos do contexto	46
q	Consulta	46
ne	Nome da Estrutura	46
E	Conjunto de elementos de uma estrutura	47
α	Similaridade Básica	47
ε	Relevância Contextual	48
β	Score de Contexto	49
ir	Taxas de Insignificância	51
δ	Escore de Similaridade Intermediário	51
μ	Fator de Correção	51
$\delta' f$	Score final de similaridade	53

SUMÁRIO

1 INTRODUÇÃO 27
2 DEFINIÇÃO DO PROBLEMA E CONCEITOS BÁSICOS 31
2.1 DEFINIÇÃO DO PROBLEMA 31
2.2 CONCEITOS BÁSICOS 34
3 TRABALHOS RELACIONADOS 37
3.1 CASAMENTO DE ESQUEMAS 37
3.2 RESOLUÇÃO DE ENTIDADES 38
3.3 DISCUSSÃO 39
4 NAZCA: UM MÉTODO DE BUSCA POR SIMILARIDADE BASEADO NO CONTEXTO 43
4.1 VISÃO GERAL 43
4.1.1 Definições Básicas 45
4.2 BUSCA POR SIMILARIDADE 46
4.3 ESCORE DE CONTEXTO 47
4.3.1 Ajuste da similaridade baseado no contexto 50
4.3.2 Árvore de Decisão 52
5 ANÁLISE EXPERIMENTAL 55
5.1 CONJUNTO DE DADOS E IMPLEMENTAÇÃO 55
5.2 AVALIAÇÃO DOS TERMOS DO CONTEXTO 56
5.2.1 Resultados 57
5.3 AVALIAÇÃO DE CONSULTAS 57
5.3.1 Preparação dos dados 58
5.3.2 Resultados 60
6 CONCLUSÃO E TRABALHOS FUTUROS 63
REFERÊNCIAS 65

1 INTRODUÇÃO

Abordagens para casamento de esquema têm sido amplamente estudadas na literatura sob diversos aspectos. Essas abordagens podem ser aplicadas em diferentes áreas, tais como, integração de esquemas, processamento de consultas semânticas, construção automática de bases de conhecimento (RAHM; BERNSTEIN, 2001; WANG et al., 2012), entre outras. Uma das mais respeitadas classificações na área, que leva em conta esses aspectos, reproduzida na Figura 1, foi feita por (RAHM; BERNSTEIN, 2001), e apresenta os mais variados problemas focados por trabalhos propostos na literatura. O presente trabalho foca no casamento de esquemas em nível de estrutura, conforme enfatizado na Figura 1. Neste caso, considera-se que a estrutura de um objeto do mundo real pode ser representada de diferentes formas em fontes de dados distintas, e essas diferenças tem motivado o desenvolvimento de soluções baseadas em similaridade (DORNELES; GONçALVES; MELLO, 2011). Entretanto, os resultados obtidos ainda não são precisos o suficiente, apresentando grande quantidade de falsos positivos e falsos negativos. Segundo (RAHM; BERNSTEIN, 2001), o casamento de esquemas em nível de estrutura caracteriza-se por considerar os elementos que aparecem juntos em uma estrutura.

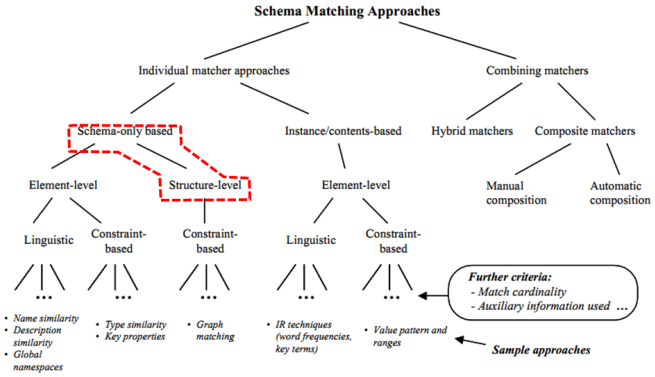

Figura 1 – Classificação de abordagens para casamento de esquemas (RAHM; BERNSTEIN, 2001).

Abordagens conhecidas para esse tipo de casamento consideram somente as informações disponíveis no próprio esquema em seu processo, ou seja, a estrutura e seus elementos, e em alguns casos os rela-

cionamentos com outras estruturas. Além dessas, outras informações podem ser consideradas nesse processo, pois auxiliam na caracterização das estruturas. Tais informações podem ser consideradas contexto, que pode ser entendido como qualquer informação interna ou externa ao Banco de Dados que pode ser usada para caracterizar o dado propriamente dito (STEFANIDIS; KOUTRIKA; PITOURA, 2011). Considerando que algumas representações de estruturas informadas como entrada possuem elementos muito distintos dos elementos das estruturas armazenadas nas fontes de dados indexadas, a proposta baseia-se no uso de informações adicionais presentes nessas fontes de dados visando melhorar a precisão da busca e reduzir os falsos positivos e falsos negativos no resultado.

Alguns trabalhos da literatura tem como objetivo descobrir entidades do mundo real em páginas web, como por exemplo (WENINGER; JOHNSTON; HAN, 2013; HE et al., 2013). A comunidade tem despendido esforços para encontrar soluções que sejam capazes de encontrar páginas-entidade ou páginas-objeto. Weninger et al. (WENINGER; JOHNSTON; HAN, 2013) define uma página-entidade como uma *Web Page* que descreve uma entidade específica, e fazer uso dos dados presentes nestas páginas é uma oportunidade de criar conhecimento útil para muitas aplicações reais, tais como comparações de preços de site de venda, pesquisa vertical, reconhecimento de entidade e sugestão de consulta. A busca por essas entidades através da consulta de um usuário por exemplo, se torna difícil considerando que a consulta pode ter o objetivo de encontrar uma determinada entidade, porém a estrutura informada difere das existentes na base de dados consultada. Nesses casos, a busca utilizando casamento em nível de estrutura e também outras informações do contexto pode ser empregada.

Para exemplificar o cenário descrito anteriormente considera-se uma consulta pela estrutura *job*, que contenha os elementos *job(name, city, salary)* nos seguintes conjuntos de dados que possuem dados adicionais representados entre [ ] *dataset 1 = job(description, city, salary, company), [career, jobs name, industry, manager, employees], dataset 2 = job(name, location, company), [career, jobs category, industries, skills, experiences]* representa um desafio pelo fato de que na consulta são informados apenas os elementos que representam a estrutura, enquanto as fontes de dados possuem a representação da estrutura e as informações adicionais que representam o contexto. Desta forma, o problema consiste em casar um objeto de dados composto por uma estrutura de esquema com um objeto de dados composto pela estrutura de esquema mais um conjunto de palavras que define seu contexto.

Neste caso, o método de similaridade deve ser capaz de comparar dois parâmetros heterogêneos e resultar em um escore de similaridade que represente a semelhança entre os objetos.

Neste trabalho, é proposto um método de casamento de esquemas em nível de estruturas, chamado *Nazca*[1], que envolve: i) funções de similaridade atômicas para elementos do esquema; ii) algoritmo para detecção das informações contextuais; e iii) árvore de decisão para o ajuste final de similaridade. Especificamente, o trabalho tem como principal objetivo a aplicação do método proposto na busca por estruturas que estejam distribuídas em diversas bases de dados, tais como, bancos de dados relacionais, formulários *web*, documentos XML, entre outros, e que podem estar armazenadas de diferentes formas. No método proposto, a entrada é a consulta que representa a estrutura desejada, e o resultado é um *ranking* das estruturas similares encontradas no conjunto de fontes de dados indexadas. De forma geral, o método se baseia em uma métrica de similaridade que utiliza informações do contexto no processo de casamento de esquemas, e funciona através dos seguintes passos: i) indexação das bases de dados a serem consultadas; ii) cálculo do escore de similaridade entre a os elementos da estrutura usada na consulta e as estruturas dos esquemas armazenados; iii) cálculo do escore de contexto da estrutura da consulta em relação à estrutura armazenada; iv) cálculo de um escore de similaridade intermediário entre a consulta e as estruturas das bases de dados; v) cálculo do escore de similaridade final, através de uma árvore de decisão usada para refinar o escore intermediário; e vi) geração do *ranking* de resultados, ordenado pelo escore de similaridade final. A proposta apresenta algumas limitações como a dependência das estruturas dos índices de elementos e de contexto. Além disso, só pode ser aplicada a fontes de dados de onde possam ser extraídas informações textuais de estruturas e contexto, o que não ocorre com figuras e documentos no formato *pdf* por exemplo.

Foram executados dois tipos de experimentos para validação do método. O primeiro experimento busca avaliar o conteúdo das fontes de dados indexadas, que permite identificar se o método proposto para extração do contexto das fontes de dados realmente recupera conteúdo relevante para representar o contexto das estruturas nelas contidas. O segundo experimento, que foi feito com formulários *web* visa a avaliação de consultas através das métricas de revocação e precisão. Ambos os

[1] Em referência às linhas de Nazca no Peru, onde as imagens desenhadas no solo apenas podem ser compreendidas com uma vista aérea que dá a compreensão do contexto http://www.nascaperu.com/en/tourist-attractions-nasca/nasca-lines.

experimentos demonstram que o ajuste dos scores baseado no contexto torna os resultados mais precisos para as buscas.

As principais contribuições deste trabalho são: i) desenvolvimento de um método para casamento de estruturas baseado em informações do contexto no ajuste dos scores de similaridade;ii) melhoria da precisão no processo de casamento de estruturas utilizando informações adicionais ao esquema e iii) redução da quantidade de falsos positivos e falsos negativos nos resultados das buscas utilizando este método.

Esta dissertação está organizada da seguinte forma, o Capítulo 2 apresenta a descrição do problema e alguns conceitos básicos para compreensão da proposta, como Funções de Similaridade e Contexto de maneira geral. O Capítulo 3 apresenta alguns trabalhos desenvolvidos nas áreas de Resolução de Entidades e Casamento de Esquemas, discutindo suas semelhanças e diferenças com a proposta apresentada nesta dissertação. O Capítulo 4 apresenta o método *Nazca*, bem como a metodologia e as equações utilizadas para o desenvolvimento do método. O Capítulo 5 descreve os experimentos realizados e os resultados obtidos. O Capítulo 6 apresenta as conclusões deste trabalho e possíveis trabalhos futuros.

2 DEFINIÇÃO DO PROBLEMA E CONCEITOS BÁSICOS

Neste capítulo é definido o problema do casamento de esquemas em nível de estrutura abordado neste trabalho, usando alguns exemplos de busca por representação de estruturas. Na proposta deste trabalho, o objetivo é buscar por uma estrutura específica em um conjuntos de dados, tais como bancos de dados relacionais, documentos XML, documentos web, entre outros.

2.1 DEFINIÇÃO DO PROBLEMA

Uma situação real de estrutura inserida em seu contexto é uma página contendo um formulário web. A Figura 2 apresenta um *formulário web* disponível em um site de pesquisas por vagas de emprego. Nesta figura, está destacada uma possível representação da estrutura *"job"*, que possui os elementos:*Job Title*, *Company Name* e *City/State*. Além disso, esse web site apresenta outras informações como dicas para a carreira e outros artigos sobre a vida profissional que podem ser identificados através de termos nessa fonte de dados, por exemplo, *resumes*, *jobs*, *company*, *industries*, *employers*, *job-hunting*, *manager* e *careers*. Essas informações adicionais podem ser usadas para melhor caracterizar a estrutura *"job"* na fonte de dados.

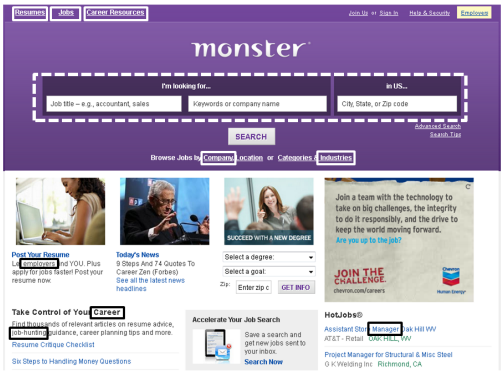

Figura 2 – Representação de uma estrutura inserida no contexto: Job - Web Form.

A representação da estrutura *"job"*, também pode ser encontrada em outras fontes de dados, como documentos XML por exemplo. A Figura 3 mostra outra possível representação desta estrutura, alguns dos seus elementos são: *job-category*, *required-skills*, *required-education*,*required-experience*, *salary-range* e *benefits* (destacados com linhas pontilhadas). Além da representação da estrutura, alguns termos do contexto também estão destacados, como por exemplo, *company*, *industry*, *contact*, *email* e *location* (destacados com linhas contínuas). As fontes de dados mostradas nas Figuras 2 e 3, que representam a estrutura *"job"* em uma página web permitem identificar algumas diferenças entre a representação da estrutura do mesmo objeto do mundo real, nesse caso *"job"*, no entanto a presença de outras informações, permitem identificar o contexto das estruturas representadas.

Figura 3 – Representação de uma estrutura inserida no contexto: Job - XML.

Para exemplificar a proposta deste trabalho, as Figuras 2 e 3 apresentam duas possíveis fontes de dados para indexação e posterior busca. Considerando uma busca pela estrutura *"job"*, é desejável que as duas fontes de dados citadas sejam recuperadas com alto escore de similaridade. Para isso, assume-se que a consulta seja escrita de tal

forma que uma estrutura seja informada, como por exemplo, *job (field, position, city, company)*. Para este exemplo, se as fontes de dados apresentadas nas Figuras 2 e 3 estiverem indexadas, elas deveriam retornar em uma alta posição no ranking final, devido ao seu conteúdo relevante para a consulta. Entretanto, se utilizada apenas a similaridade de elementos da estrutura, observa-se que os elementos na Figura 3 apresentam baixa similaridade com os elementos informados na consulta e provavelmente este documento ficaria entre os falsos negativos do resultado da consulta. A análise dos termos desta fonte de dados pode ajudar a caracterizar a estrutura, aumentando assim o seu escore de similaridade em relação à consulta.

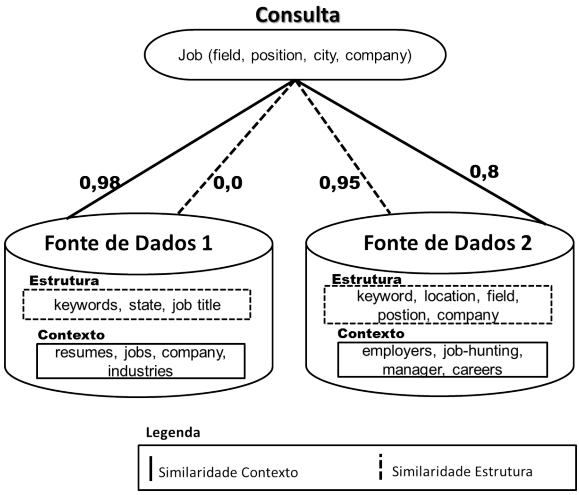

Figura 4 – Exemplo de consulta com contexto e representação da estrutura

A Figura 4 apresenta um exemplo de consulta e fontes de dados utilizadas na busca. Como pode ser visto, na consulta são informados apenas os elementos que representam a estrutura, enquanto as fontes de dados possuem duas representações: a representação da estrutura e o contexto. O principal desafio é encontrar o escore de similaridade entre a consulta e o contexto da fonte de dados, definindo como usar este valor para ajustar o escore final. No exemplo apresentado na Figura 4 pode-se observar que para a Fonte de Dados 2 o valor do escore de similaridade de elementos é 0.95, o que o faria retornar em uma alta posição no ranking, já a Fonte de Dados 1 tem como valor do escore da

similaridade de elementos 0.0, assim, esta fonte de dados retornaria em uma baixa posição no ranking se somente este escore fosse considerado. De outra forma, se o escore de contexto for considerado, a Fonte de Dados 1 poderia retornar em uma alta posição no ranking, considerando que apresenta um valor de 0,98 para o escore de contexto. A Fonte de Dados 2 também pode ser considerada relevante para a consulta, pois o valor do escore de contexto é 0,80.

2.2 CONCEITOS BÁSICOS

A seguir serão apresentados alguns conceitos importantes para a compreensão da proposta:casamento de esquemas, funções de similaridade e contexto.

Casamento de esquemas: casamento de esquemas é o problema de geração de correspondências entre elementos de dois esquemas (BERNSTEIN; MADHAVAN; RAHM, 2011; RAHM; BERNSTEIN, 2001). O casamento de esquemas é utilizado em integração de dados, consultas aproximadas, pesquisa por similaridade, ou seja, sempre que se pretende analisar dados provenientes de uma ou várias fontes de dados para determinar se representam ou não o mesmo objeto do mundo real. O processo de casamento nem sempre é trivial, visto que os dados podem apresentar heterogeneidade de representação, tanto na estrutura quanto em seus valores. Portanto esse processo deve ser capaz de analisar estrutura e valor para atingir com maior precisão seus objetivos (DORNELES et al., 2009; DORNELES; GONçALVES; MELLO, 2011).

Em (BERNSTEIN; MADHAVAN; RAHM, 2011) há uma visão geral da evolução da pesquisa na área de casamento de esquemas. São demonstradas diferentes técnicas, algoritmos e ferramentas conhecidas para este processo. Entre as principais técnicas empregadas em casamento de esquemas estão:

- **Casamento Linguístico**: baseado em nomes ou descrição de elementos, strings e substrings e técnicas de recuperação de informação.
- **Informações auxiliares**: baseado em dicionários, acrônimos entre outras informações.
- **Casamento baseado em instâncias**: elementos de esquemas são considerados similares se as suas instâncias são similares, baseado em estatísticas, metadados, ou treinamento de classificadores.

- **Casamento baseado em estruturas**: elementos do esquema são similares se eles aparecem em estruturas similares, tem relacionamentos similares, ou tem relacionamentos com elementos similares.

- **Casamento baseado em restrições**: baseado em tipos de dados, intervalo de valores, regras de unicidade e chaves estrangeiras.

- **Casamento baseado em regras**: baseado em regras de correspondência que são expressas em lógica de primeira ordem.

- **Casamento híbrido**: que utiliza diferentes critérios para o casamento.

- **Casamento de grafos**: baseado na comparação de relacionamentos entre elementos de grafos.

- **Similaridade de conteúdo**: onde as instâncias de elementos de um esquema são agrupadas em um documento que é então combinado com outros documentos com base na medida de recuperação de informação TF-IDF.

- **Link de documentos**: onde os conceitos de duas ontologias são considerados similares se as entidades desses conceitos são similares.

Além de técnicas para casamento de esquemas (BERNSTEIN; MADHAVAN; RAHM, 2011) também cita algumas abordagens que combinam tipos de algoritmos. Estratégias de *workflow*, executam diferentes algoritmos de casamento e ao final da execução combinam os seus resultados. De forma diferente, em outras abordagens os diferentes passos do algoritmo são executados paralelamente ou diferentes partições do esquema são analisadas em paralelo. Mesmo com a evolução das técnicas de casamento de esquemas ao longo dos anos, os algoritmos de execução automática (sem intervenção do usuário) ainda não atingiram uma precisão significativa nos resultados. Assim, o problema de casamento de esquemas ainda permanece em aberto possibilitando oportunidades de pesquisa e desenvolvimento de novas técnicas que tornem o processo cada vez mais preciso.

Resolução de Entidades: processo que identifica registros que referem-se à mesma entidade do mundo real e que combina as diferentes representações em uma única (WANG et al., 2011; WHANG; BENJELLOUN; GARCIA-MOLINA, 2009; PLOCH, 2011). O processo de resolução

de entidades difere do casamento de esquemas, pois considera somente as instâncias dos dados, como por exemplo, nomes de cidades e não as estruturas que são inerentes de esquemas, como por exemplo, a estrutura de uma tabela relacional que armazena informações de cidades.

Funções de similaridade: Uma função de similaridade retorna um escore para um par de valores, esses valores são considerados similares se este valor ultrapassa um limiar pré-definido. Quando os dados analisados pelo processo de casamento são atômicos, como por exemplo (strings, valores numéricos, datas, etc.) pode ser usada uma função de similaridade apropriada para o tipo de dado. Em casos onde os valores são complexos, como tuplas em bancos de dados relacionais, web forms e documentos XML, podem ser adotadas abordagens que combinam as funções de similaridade para dados atômicos, com outras técnicas mais sofisticadas, como aprendizado de máquina por exemplo (DORNELES; GONçALVES; MELLO, 2011). Especificamente para strings, existem funções de similaridade como Levenshtein (LEVENSHTEIN, 1966), Jaro-Winkler (JARO, 1976), Q-Gram (ELMAGARMID; IPEIROTIS; VERYKIOS, 2007), entre outras. As diferentes funções de similaridade resultam valores diferentes para as mesmas entradas. A escolha do melhor método depende do domínio em que o cálculo de similaridade será aplicado (LEVIN; HEUSER, 2010).

Contexto: de maneira geral, o contexto pode ser entendido como um conhecimento que auxilia na identificação do que é ou não relevante em um dado momento e lugar. O contexto em sistemas é usado para fornecer serviços ou informações mais relevantes para os usuários na realização das suas tarefas. Segundo (DEY, 2001), contexto é qualquer informação que caracteriza a situação de uma entidade (pessoa, lugar ou objeto) considerada relevante para a interação entre uma pessoa e uma aplicação. Trazendo este conceito para Bancos de Dados, contexto é qualquer informação externa ao Banco de Dados que pode ser usada para caracterizar a situação de um usuário ou qualquer informação interna ao Banco de Dados que pode ser usada para caracterizar o dado propriamente dito (STEFANIDIS; KOUTRIKA; PITOURA, 2011). O contexto em Banco de Dados pode ser usado tanto para personalizar informações armazenadas, quanto para recuperar informações mais precisas para as consultas.

3 TRABALHOS RELACIONADOS

Casamento de esquemas e resolução de entidades são assuntos discutidos por diferentes segmentos de pesquisa envolvendo similaridade e o uso de outras informações para melhorar o processo. O desafio é tornar o resultado da similaridade mais preciso, especialmente em *casamento de esquemas*, onde a resolução de entidades é uma tarefa essencial. Abaixo são apresentados alguns trabalhos envolvendo *casamento de esquemas* e *resolução de entidades* com diferentes abordagens de solução.

3.1 CASAMENTO DE ESQUEMAS

Um algoritmo para casamento de esquemas é o *Similarity Flooding* (MELNIK; GARCIA-MOLINA; RAHM, 2002), que é um algoritmo que pode ser usado para o casamento de diversas estruturas. Tais estruturas, que são chamadas de modelos, podem ser esquemas de dados, instâncias de dados ou a combinação de ambos. Os elementos dos modelos representam artefatos tais como tabelas relacionais e colunas por exemplo. O algoritmo é baseado na seguinte idéia:

1. Os modelos são convertidos em grafos rotulados;
2. Esses grafos são usados em um cálculo iterativo que irá determinar quais nodos do primeiro grafo são similares aos nodos do segundo grafo.
3. A computação da similaridade é baseada na ideia de que dois elementos de dois modelos distintos são similares quando seus elementos adjacentes são similares.

O objetivo geral do *Similarity Flooding* (MELNIK; GARCIA-MOLINA; RAHM, 2002) é projetar uma ferramenta que ajuda a manipular e manter esquemas, instâncias e casar os resultados. O processo é feito com o auxílio dos usuários, que através de uma interface gráfica, ajusta o casamento proposto adicionando ou removendo linhas que conectam os elementos dos dois esquemas. A precisão do algoritmo é avaliada pela quantidade de ajustes necessários.

Outra abordagem de casamento de esquemas é o *SimRank* (JEH; WIDOM, 2002), que pode ser aplicada em qualquer domínio com relacionamentos entre objetos, chamado de contexto estrutural. A medida

de similaridade do contexto estrutural em que os objetos ocorrem é baseada no seu relacionamento com outros objetos. A medida define que "dois objetos são similares se estão relacionados a objetos similares", e é baseada em modelo de grafos. *SimRank* pode ser combinada com outras métricas de similaridade específicas do domínio em que está sendo aplicada. Vários aspectos dos objetos podem ser usados para determinar similaridade e geralmente dependem do domínio. Essa é uma abordagem geral que explora os relacionamentos entre objetos, encontrados em muitos domínios de interesse. Por exemplo, duas páginas estão relacionadas se há hiperlinks entre elas.

3.2 RESOLUÇÃO DE ENTIDADES

Em (WANG et al., 2011), a resolução de entidades é feita baseando-se em uma tabela com um conjunto de registros e em regras de casamento de registros com funções de similaridade e *thresholds* desconhecidos, o objetivo é encontrar as melhores funções de similaridade e os melhores *thresholds*, para combinar as entidades. Como entrada, o usuário informa exemplos positivos (ex.: registros que representam a mesma entidade) e exemplos negativos (ex.: registros que não representam a mesma entidade). Esses exemplos são usados para identificar as melhores funções de similaridade, utilizando-as para eliminar redundâncias.

Outra abordagem utilizada na solução do problema da resolução de entidades é a de pares genéricos (WHANG; BENJELLOUN; GARCIA-MOLINA, 2009). Neste caso, para determinar se dois registros representam a mesma entidade do mundo real, são usadas duas funções, uma de *match* e outra de *merge*. Um especialista do domínio escreve essas funções, que também podem ser desenvolvidas através de técnicas de aprendizado de máquina. A função de *match* avalia se um par de registros representa a mesma entidade do mundo real. Se a função *match* retornar uma resposta verdadeira, então a função de *merge* é usada para criar um registro composto pelos dois anteriormente analisados. Depois de combinados os registros, a função de *match* pode ser aplicada novamente a fim de comparar com um terceiro registro, por exemplo.

Considerando a complexidade do problema da resolução de entidades, (PLOCH, 2011) explora os relacionamentos entre as entidades. O processo envolve a construção de uma base de conhecimento a partir das informações coletadas dos artigos da *Wikipedia*, como por exemplo, título e links da página, de onde são extraídos e armazenados nomes de

entidades. A base de conhecimento serve para rotular corretamente as formas diferentes de se referir à mesma entidade e também formas iguais de se referir a entidades diferentes da *Wikipedia*. Depois de identificadas as formas de referência a entidades dentro de um artigo. Através de uma estrutura de grafo criada com base nos links de cada página, é possível identificar a quais entidades elas referem-se, auxiliando na desambiguação.

O foco do Flint (BLANCO et al., 2008), além da resolução de entidades, é a busca de entidades. O objetivo é pesquisar, coletar e indexar automaticamente páginas web que contenham representações de instâncias de uma determinada entidade do mundo real. A busca é feita através da entrada de algumas páginas web rotuladas, que são usadas como sementes, de onde o sistema infere a descrição conceitual da entidade e pesquisa por outras páginas contendo dados que representem instâncias dessa entidade.

LINDEN (SHEN et al., 2012) é um novo *framework* desenvolvido para relacionar entidades nomeadas em um texto com conceitos presentes em uma base de conhecimento que unifica *Wikipedia* e *WordNet* utilizando o rico conteúdo semântico para criar uma base de conhecimento. O processo de construção é feito coletando um dicionário de formas de representação das entidades em quatro fontes na *Wikipedia*, páginas de conteúdo referente às entidades, páginas redirecionadas, páginas de desambiguação e *hyperlinks* em artigos da *wikipedia*. A partir desses dados é armazenada a informação de quantidade para cada entidade alvo no dicionário. Usando este dicionário, uma lista de entidades candidatas é gerada para cada menção da entidade e todas as possíveis entidades correspondentes dessa menção são incluídas na lista gerada. A informação de quantidade é usada para definir a *probabilidade de link* para cada entidade candidata. Então, uma rede semântica é construída pela definição da associatividade e similaridade semântica. Além disso, a *coerência* é definida para cada entidade candidata para mensurar a *coerência global* do mapeamento das entidades do documento. Então, um *ranking* de entidades candidatas é feito para cada menção da entidade combinando as medidas: *probabilidade de link*, *associatividade semântica*, *similaridade semântica* e *coerência global*.

3.3 DISCUSSÃO

Alguns trabalhos recentes baseiam-se no conhecimento do usuário para informar exemplos de falsos positivos e falsos negativos ou

em construções de bases de conhecimento para auxiliar em processos de resolução de entidades e casamento de esquemas. Tanto o *feedback* do usuário, quanto a construção de bases de conhecimento, podem se tornar tarefas exaustivas pois nem sempre o usuário está disposto à interagir e nem sempre a tarefa de construção de bases de conhecimento é simples A solução proposta neste trabalho difere dos citados anteriormente, pois não usa o conhecimento do usuário e nem base de conhecimento para melhorar o escore de similaridade. O objetivo é usar um vetor de termos da própria fonte de dados como contexto para melhorar o ajuste do seu escore de similaridade a fim de tornar mais preciso o processo de casamento de estruturas.

Alguns estudos na literatura representam o contexto das entidades em um documento (BAGGA; BALDWIN, 1998; MANN; YAROWSKY, 2003; PEDERSEN; PURANDARE; KULKARNI, 2005). *Bagga e Baldwin* (BAGGA; BALDWIN, 1998) usam um conjunto de palavras para representar o contexto de entidades, aplicando a técnica da clusterização aglomerativa baseada no cosseno do vetor de similaridade. *Mann e Yarowsky* (MANN; YAROWSKY, 2003) estendem o trabalho de *Bagga e Baldwin* adicionando recursos avançados de fatos biográficos.*Pedersen et al.* (PEDERSEN; PURANDARE; KULKARNI, 2005) apresenta uma abordagem não-supervisionada para ambiguidade de nomes, agrupando as instâncias de um determinado nome, cada um dos quais associado com uma entidade distinta subjacente. Os recursos empregados para representar o contexto são bigramas estatisticamente significativos que ocorrem no mesmo contexto do nome ambíguo e uma matriz de coocorrências com os escores de semelhança. Neste caso, a comparação entre os objetos é homegênea, pois ambos são representados por "nome + contexto".

A principal diferença dos trabalhos de (BAGGA; BALDWIN, 1998; MANN; YAROWSKY, 2003; PEDERSEN; PURANDARE; KULKARNI, 2005) com o *Nazca* é em relação ao que se passa como parâmetro de entrada. No *Nazca*, a entrada é uma consulta que representa apenas a estrutura de um objeto; esta consulta é comparada com outra estrutura e o seu contexto. O desafio é inferir o contexto da estrutura representada na consulta para calcular a similaridade baseada no contexto das fontes de dados, ou seja, não é possível utilizar técnicas onde são necessárias comparações de diferentes representações de contexto.

A Figura 5 mostra as diferenças entre os principais aspectos de alguns importantes trabalhos relacionados. Esses trabalhos foram comparados em relação a: *dados de entrada, intervenção do usuário, itens comparativos, processo de combinação, objetivo do processo e base de*

		Casamento de Esquemas				Resolução de Entidades		
		MétodoNazca	Melnik et al. 2002	Jeh and Widom 2002	Li et al. 2009	Bagga 1998	Mann and Yarowsky 2003	Wang et al. 2012
Dados de Entrada	Grafo		X	X				
	Ontologia				X			
	Consulta	X					X	
	Árvore DOM							X
	Documentos					X		
Intervenção do Usuário			X					X
Itens comparativos (Matchers)	Instância					X	X	
	Estrutura	X	X		X			
	Relacionamentos		X	X				
	Informações da fonte de dados	X			X	X	X	X
Processo de Combinação	Linguístico				X			
	Machine Learning							X
	Similaridade de Strings	X	X					
	Métricas de similaridade do domínio			X				
	Vetor					X	X	
Objetivo do Processo	Casamento de objetos		X	X	X	X		
	Score de Similaridade	X						
	Clusterização						X	X
Base de Conhecimento								X

Figura 5 – Tabela comparativa com outras abordagens semelhantes

conhecimento. A solução proposta neste trabalho difere das demais, pois não necessita de esquemas completos, intervenção do usuário ou bases de conhecimento para calcular o escore de similaridade. No entanto a proposta é usar um vetor de termos do contexto extraídos da própria fonte de dados no cálculo do escore de similaridade tornando o seu resultado mais preciso. Além disso, a maioria dos trabalhos relacionados utiliza características inerentes de um esquema completo, como relacionamentos entre entidades, o que não encontra-se em *web forms* por exemplo, onde a representação da estrutura está isolada, porém podem ser encontradas outras informações (contexto), que pode ser utilizado no cálculo de similaridade. Outra característica encontrada em trabalhos que baseiam-se no contexto é que as entradas também tem alguma representação adicional à entidade ou ao esquema. No caso de Nazca, o usuário informa uma consulta, sem qualquer informação adicional e o método utiliza essa consulta combinada com o contexto das fontes de dados para o seu processo de casamento.

4 NAZCA: UM MÉTODO DE BUSCA POR SIMILARIDADE BASEADO NO CONTEXTO

Este capítulo apresenta o método Nazca[1], mostrando uma visão geral, alguns conceitos e definições, assim como os detalhes do processamento de consultas descrevendo todas as fases do processo com a utilização de dados do contexto.

4.1 VISÃO GERAL

Conforme mencionado anteriormente, o objetivo da proposta é encontrar representações de estruturas similares à informada na consulta levando em consideração os seus elementos e o contexto da estrutura contida no documento da fonte de dados. Para isso são necessárias duas etapas, *indexação* e *processamento de consultas*. A Figura 6 mostra a arquitetura geral dessa abordagem.

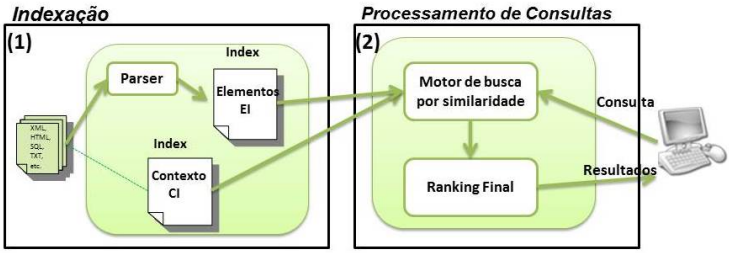

Figura 6 – Visão Geral da Proposta

- `Indexação:` Nesta etapa são necessárias duas estruturas de índice: `Índice de Elementos` (EI) e `Índice de Contexto` (CI):

  - `Índice de Elementos (EI)`: indexa os elementos das estruturas armazenadas nas fontes de dados. A extração desses elementos não faz parte do escopo deste trabalho e pode ser feita com o auxílio de algum algoritmo para extração

[1] Em referência às linhas de Nazca no Peru, onde as imagens desenhadas no solo apenas podem ser compreendidas com uma vista aérea que dá a compreensão do contexto http://www.nascaperu.com/en/tourist-attractions-nasca/nasca-lines.

automática de atributos como por exemplo, (FURCHE et al., 2012).

– `Índice de Contexto (CI)`: armazena as informações contextuais da estrutura em uma fonte de dados. Essas informações são representadas através de um vetor de termos que pode ser construído utilizando uma ferramenta de indexação, como Lucene[2] por exemplo. A construção deste índice é feita em três etapas:

1. identificação dos termos que representam informações contextuais no documento (retirando-se a representação da estrutura);
2. remoção de dados irrelevantes do conjunto de informações contextuais, tais como *stop words*, números e caracteres de pontuação;
3. indexação dos termos em uma estrutura com os seguintes campos: *identificador do documento*, *caminho do arquivo do documento*, *termo* e *domínio*. A Figura 7 ilustra esta estrutura de armazenamento;

Doc. Id	arquivo	termo	domínio
135	D:\HTML2\223\page.html	flight	airfare
158	D:\HTML2\257\page.html	flight	airfare
159	D:\HTML2\258\page.html	flight	airfare
761	D:\HTML2\1113\page.html	book	books
762	D:\HTML2\1114\page.html	history	books
538	D:\HTML2\797\page.html	book	books

Figura 7 – Exemplo de Índice de Contexto (CI)

- `Processamento de Consultas:` durante o processamento das consultas, mostrado na Figura 8, a entrada é uma consulta do usuário e são usados os índices *EI* e *CI* para o cálculo do escore de similaridade e posterior geração de um *ranking* final de documentos.

  Na etapa de *processamento de consultas*, detalhada na Figura 8, a entrada é uma consulta contendo nome e elementos, representando uma estrutura. O motor de buscas utiliza os índices EI e CI para buscar por estruturas similares à consulta. Inicialmente os escores de similaridade de elementos e contexto são obtidos e

[2] http://lucene.apache.org/java/docs/index.html

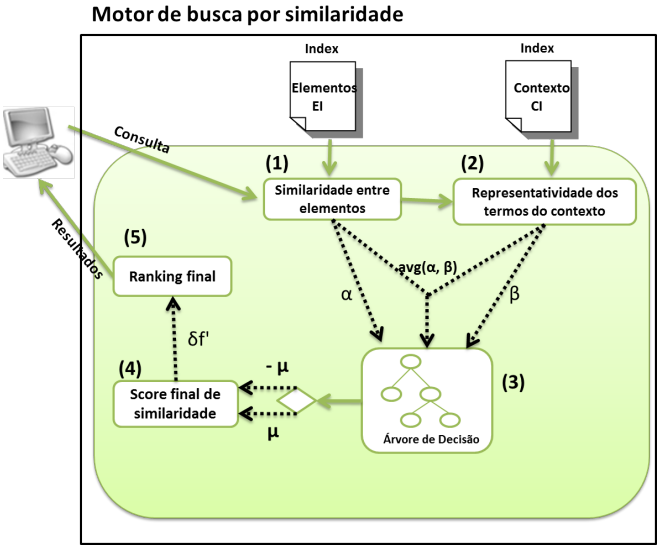

Figura 8 – Motor de Busca por similaridade

então uma *Árvore de Decisão* é empregada pra ponderar esses valores e o resultado é o escore final de similaridade. O processo é concluído quando um *ranking* de documentos é construído baseado nesse valor.

4.1.1 Definições Básicas

Antes do detalhamento da proposta, é importante definir alguns conceitos básicos para o método proposto, como, *termo*, *documento*, *estrutura* e *contexto*, que são essenciais para entender o método proposto. Uma fonte de dados é representada por qualquer fonte de informação, que pode ser, banco de dados relacional, arquivos em disco, informações disponíveis nos mais diversos formatos na web.

Definição 4.1.1 (Termo)**.** *Um termo t é definido como uma palavra, ou palavras consecutivas em um documento, mais especificamente, é um conjunto de caracteres com significado para o idioma a que pertence.*

Termos em um documento podem ser palavras que representam o domínio do documento. Alguns exemplos de termos apresentados na Figura 3 são *manager*, *job*,*salary*, *benefits* e *location*. Esses termos in-

dicam empiricamente que o documento provavelmente está relacionado com o mercado de trabalho. Nesta proposta esses termos são classificados em *elementos*, que caracterizam a estrutura e *contexto*, que caracterizam o documento.

Definição 4.1.2 (Documento, Estrutura e Contexto)**.** *Um documento D é definido como um conjunto contendo um número finito de termos t, ou seja,$D = \{t_1, t_2 \ ... \ t_n\}$. Uma estrutura e é definida como um subconjunto de D com um número finito de termos representando elementos, tal que $e \subset D$. Seja D um documento, o contexto Co é definido como um subconjunto de termos obtidos pela subtração entre D e e, ou seja, $Co = D - e$, e $Co \subset D$.*

4.2 BUSCA POR SIMILARIDADE

Nesta seção é explicado o funcionamento da busca por similaridade de acordo com o método proposto. São descritas algumas definições importantes para a compreensão dos cálculos empregados para obter o escore final de similaridade.

Definição 4.2.1 (Consulta)**.** *Seja ne_i o nome de uma estrutura e_i e $E_i = (e_1, ..., e_n)$ o conjunto de n elementos de e_i, uma consulta q_j é definida por: $q_j = ne_j(e_1, ..., e_m)$.*

Um exemplo de consulta pela estrutura Job, poderia ser construído da seguinte forma, q1=*job(field, industry, city, company)* de acordo com a Definição 4.2.1.

O cálculo de similaridade entre elementos (*Similaridade Básica*) é feito através da comparação de cada um dos elementos da consulta com cada um dos elementos das estruturas armazenadas no EI. A função de similaridade utilizada para esta comparação nos experimentos é Q-Grams (ELMAGARMID; IPEIROTIS; VERYKIOS, 2007). Se o escore de similaridade para uma comparação for menor do que 1.0, são empregados os sinônimos dos elementos com o auxílio da ferramenta *WordNet*[3], desta forma é possível tornar este cálculo mais exato.

Definição 4.2.2 (Similaridade Básica)**.** *Seja q_i uma consulta composta pelo conjunto de elementos $E_i = (e_1, ..., e_n)$, e_j a j-ésima representação de estrutura armazenada no EI, composta pelo conjunto de elementos $E_j = (e_1, ..., e_m)$. O escore de similaridade entre elementos é dado por:*

[3] http://wordnet.princeton.edu/

$$\alpha_{(q_i,e_j)} = \frac{\sum_1^n (max(sim(e_i^n, [e_1, ..., e_m])))}{max(m, n)} \quad (4.1)$$

onde $(1 \leq i \leq n), (1 \leq j \leq m)$, sim() é uma função de similaridade que retorna um valor entre 0 e 1, como por exemplo Levenshtein (LEVENSHTEIN, 1966), Jaro-Winkler (JARO, 1976) ou Q-Grams (ELMAGARMID; IPEIROTIS; VERYKIOS, 2007). Cada elemento da consulta q_i é comparado com todos os elementos da estrutura e_j na fonte de dados, e a função *max()* retorna o escore máximo de similaridade para o elemento mais similar. Os escores resultantes para cada um dos elementos da consulta são somados e o valor resultante da soma é normalizado por *max(m, n)*, ou seja, considera-se na divisão a estrutura com maior número de elementos.

A Equação 4.1 gera um escore de similaridade entre os elementos da consulta e os elementos da estrutura armazenada no EI. Considerando que o valor do *escore de similaridade básica* (α) pode resultar em falsos positivos e falsos negativos para a consulta, a proposta é o ajuste desse valor baseado nos termos extraídos do contexto dos documentos das fontes de dados e armazenados no CI.

Para exemplificar, considera-se uma consulta q_1 *Job(field, industry, city, company)* e uma estrutura e_1 *(state, field, company)* armazenada em uma fonte de dados. O cálculo resultante da utilização de uma função de similaridade de strings poderia ser: $\frac{1.0+0.0+0.6+1.0}{4} = 0.65$.

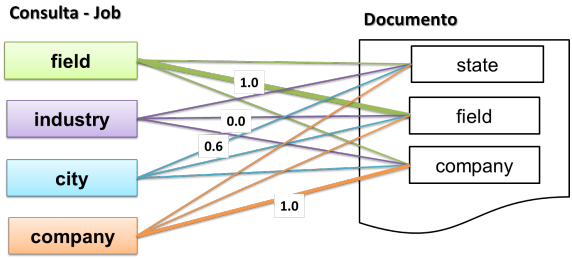

Figura 9 – Exemplo Similaridade Básica

4.3 ESCORE DE CONTEXTO

Para determinar o escore de contexto, inicialmente é calculada a relevância dos termos indexados no CI definindo quantitativamente

a representatividade dos termos de cada documento indexado para a consulta informada. A avaliação dos termos é baseada no cálculo de *Relevância Contextual* ε, que determina a importância do termo para uma determinada consulta. Esta avaliação é baseada em (SILVA et al., 2007), porém com diferentes propósitos. O valor de ε é calculado considerando o número de resultados obtidos com a busca de um termo junto com o nome de uma estrutura, no caso o nome informado na consulta, em relação ao número total de ocorrências desse termo no CI.

Definição 4.3.1 (Relevância Contextual)**.** *Seja $O_{(t,ne_i)}$ o número de ocorrências do termo t com o nome da estrutura ne_i de uma consulta e T_t o número total de ocorrências do termo t nas fontes de dados indexadas no CI, a Relevância Contextual ε resulta em um valor entre 0 e 1 e é calculada por:*

$$\varepsilon_{(t,ne_i)} = \frac{(O_{(t,ne_i)})}{T_t} \tag{4.2}$$

O valor de *relevância contextual* (ε) para cada termo no CI é armazenado em um índice com as seguintes colunas: *doc_id*, *nome_estrutura*, *termo* e $\varepsilon_{(t,ne_i)}$, um exemplo pode ser visto na Tabela 1.

doc_id	**nome_estrutura**	**termo**	$\boldsymbol{\varepsilon_{(t,ne_i)}}$
1245	*auto*	*passport*	0,61
1245	*auto*	*airport*	0,10
1245	*auto*	*reservation*	0,29
1245	*auto*	*maintenance*	0,52
1245	*flight*	*passport*	0,85
1245	*flight*	*maintenance*	0,39
1245	*flight*	*airport*	0,94
1245	*flight*	*reservation*	0,78

Tabela 1 – Armazenamento de ε

Para o cálculo do escore final de similaridade são levados em consideração os termos do contexto, para isso é calculado o valor do *score de contexto* (β), que indica o grau de relevância total dos termos indexados no CI de um determinado documento relacionados a uma determinada estrutura informada na consulta. Este valor é a média de ε de todos os termos de cada documento em CI.

Definição 4.3.2 (Escore de Contexto)**.** *Seja n o número total de ter-*

mos indexados para um determinado documento D_i, ne_j o nome da estrutura informado em uma determinada consulta e $\varepsilon_{((t_k,ne_j))}$ o valor da relevância contextual para o termo t_k em relação ao nome da estrutura ne_j para o documento D_i. O escore de contexto β para D_i e ne_j é calculado por:

$$\beta_{(D_i,ne_j)} = \frac{\sum_{i=1}^{n} \varepsilon_{(t,ne_i)}}{n} \tag{4.3}$$

Para exemplificar visualmente os valores obtidos pelo cálculo da *relevância contextual* (ε), utiliza-se o *G*rafo de Contexto. Este grafo mostra quantitativamente a conexão dos termos com determinados nomes de estruturas utilizados em consultas. A Figura 10 exemplifica uma parte do *Grafo de Contexto* para alguns termos associados às estruturas *auto* e *flight*. Os nodos preenchidos representam os nomes das estruturas e os nodos vazados representam termos indexados em CI. Os pesos das arestas são determinados pelos valores de *Relevância Contextual* (ε) para os termos em relação às estruturas. Na Figura 10 observa-se a relevância do termo *airport* para a estrutura *flight*, sendo $T_{airport} = 898$, $O_{(airport,flight)} = 845$. Aplicando-se a Equação 4.2, o valor de ε obtido é $\varepsilon_{(airport,flight)} = 0,94$. Para a estrutura *flight* e o termo *maintenance*, os valores são $T_{maintenance} = 460$ e $O_{(maintenance,flight)} = 180$, aplicando-se a Equação 4.2, o valor da ε obtido é $\varepsilon_{(maintenance,flight)} = 0,39$. Desta forma, de acordo com os valores, pode-se concluir que o termo *airport* apresenta uma relevância maior para a estrutura *flight* do que o termo *maintenance* de acordo com a abordagem proposta.

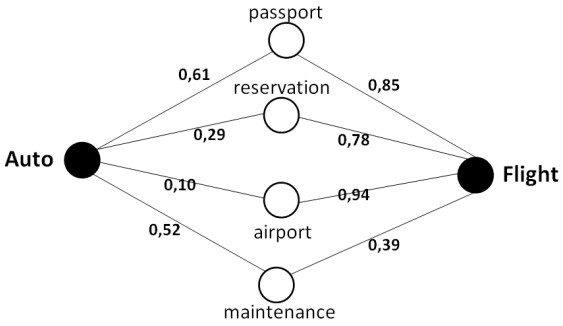

Figura 10 – Grafo de Contexto

A construção do grafo de contexto com os termos indexados de um determinado documento em relação a uma determinada consulta, é feita através do Algoritmo 1. Este algoritmo executa uma busca por cada termo do contexto de determinado documento $D_t : (t_0, ..., t_n)$ em todo o CI, obtendo uma quantidade de resultados (T_t) e posteriormente é feita uma nova busca pelo termo junto com o nome da estrutura informado na consulta obtendo outra quantidade de resultados (O). Os números de resultados obtidos para essas duas buscas são usados para calcular o peso do termo para a consulta (ε). Após esse cálculo para cada um dos termos do contexto de uma fonte de dados, todos esses valores são utilizados no cálculo do escore de contexto (β) que determina a importância dos termos do contexto indexados para a fonte de dados em relação à consulta.

Algorithm 1 Cálculo do Score de Contexto

1: CI : Índice de Contexto
2: $D_t : (t_0, ..., t_n)$: Vetor de termos da fonte de dados
3: ne_j : Nome da estrutura informado na consulta
4: L : Vetor de ε para o documento D_t
5: $i = 0, j = 0$;
6: **for all** $t_i \in D_t$ **do**
7: $T_t \leftarrow$ número total de ocorrências de t_i em CI;
8: $O \leftarrow$ número total de co-ocorrências de t_i e ne_j em CI;
9: $\varepsilon \leftarrow T_t/O$;
10: $L \leftarrow L \cup \varepsilon$ add ε to L
11: $i++$;
12: **end for**
13: $\beta \leftarrow$ average of L

4.3.1 Ajuste da similaridade baseado no contexto

Antes de obter o *escore final de similaridade*, é calculado *score de similaridade intermediário* $\delta_{(q_i,e_j)}$. Esse cálculo envolve os valores obtidos de: *similaridade básica* $\alpha_{(q_i,e_j)}$, *score de contexto* $\beta_{(D_i,ne_j)}$ e as *taxas de insignificância* ir_α e ir_β.

Definição 4.3.3 (Taxa de Insignificância)**.** *Seja* $\alpha_{(q_i,e_j)}$ *o escore de similaridade básica e* $\beta_{(D_i,ne_j)}$ *o escore de contexto, a taxa de insignificância é calculada por:*

$$ir_{\alpha} = 1 - \alpha_{(q_i,e_j)}$$
$$ir_{\beta} = 1 - \beta_{(D_i,ne_j)} \tag{4.4}$$

As taxas de insignificância servem para balancear a função do *score de similaridade intermediário*. Esses valores indicam o peso dos valores de $\alpha_{(q_i,e_j)}$ e $\beta_{(D_i,ne_j)}$ no cálculo.

O *escore de similaridade intermediário* representa o valor obtido do cálculo de similaridade básica (α), ajustado pelo valor do escore de contexto (β) de um documento em relação a uma determinada consulta considerando as *taxas de insignificância*. Desta forma através deste valor determina-se quantitativamente a importância do documento para a consulta de acordo com a proposta de utilização dos termos do contexto.

Definição 4.3.4 (Escore de similaridade intermediário)**.** *Seja $\alpha_{(q_i,e_j)}$ a similaridade básica entre a consulta q_i e os elementos da estrutura e_j, seja $\beta_{(D_i,ne_j)}$ o valor correspondente ao escore de contexto para a fonte de dados D_i em relação ao nome da estrutura ne_j informado na consulta , e ir a taxa de insignificância, o escore de similaridade intermediário é calculado por:*

$$\delta_{(\alpha,\beta)} = \frac{(\alpha_{(q_i,e_j)} * ir_{\alpha_{(q_i,e_j)}}) + (\beta_{(D_i,ne_j)} * ir_{\beta_{(D_i,ne_j)}})}{(ir_{\alpha_{(q_i,e_j)}}) + (ir_{\beta_{(D_i,ne_j)}})} \tag{4.5}$$

Para exemplificar, considera-se uma consulta q_1 pela estrutura e_1 *job(field, industry, city, company)* que obteve os seguintes valores em relação ao documento D_i: $\alpha_{(q_1,e_1)} = 0.65$, $\beta_{(D_i,ne_1)} = 0.80$, $ir_{\alpha} = 0.35$ e $ir_{\beta} = 0.20$. Assim, o valor do *score de similaridade intermediário* obtido é $\delta_{(q_1,e_1)} = 0.70$. Para tornar o valor de similaridade final mais preciso, o valor de $\delta_{(q_i,e_j)}$ sofre uma correção baseada no valor do fator de correção μ. O fator de correção μ é um valor adicionado ou subtraído do valor de $\delta_{(q_i,e_j)}$ ajustando o escore obtido para cima ou para baixo.

Definição 4.3.5 (Fator de Correção)**.** *Seja $\delta_{(\alpha,\beta)}$ obtido previamente, o fator de correção μ é um valor percentual, calculado por:*

$$\mu = \frac{(\delta_{(\alpha,\beta)} * ((1 - \delta_{(\alpha,\beta)}) * 100))}{100} \tag{4.6}$$

Considerando ainda o exemplo acima, o *fator de correção* obtido

é $\mu = \frac{(0.70*((1-0.70)*100))}{100}$, $\mu = 0.21$. Este valor representa um ajuste no valor do *score de similaridade intermediário* para chegar ao *score final de similaridade*, podendo aumentar ou reduzir a posição do documento no ranking. Este ajuste a partir do *fator de correção* é definido por uma árvore de decisão, cuja construção e utilização é descrita a seguir com maiores detalhes.

4.3.2 Árvore de Decisão

As árvores de decisão constituem uma técnica muito poderosa e amplamente utilizada em problemas de classificação, geralmente empregadas na análise multivariável (MITCHELL, 1997). Segundo (TAN; STEINBACH; KUMAR, 2005) a classificação pode ser utilizada para modelagem descritiva e modelagem preditiva. Na modelagem preditiva, um modelo de classificação é utilizado para classificar exemplos cujas classes são desconhecidas, que é o caso do fator de correção μ.

Devido às características das árvores de decisão como modelos de classificação, a sua utilização é adequada para o problema de classificação do fator de correção μ em duas classes: positivo ou negativo. Esta classificação depende de três variáveis (δ, β e α), que são consideradas os atributos preditivos do algoritmo. Esses atributos estão presentes no cálculo do *score de similaridade intermediário* $\delta_{(q_i,e_j)}$ e por isso os seus pesos podem influenciar os valores uns dos outros.

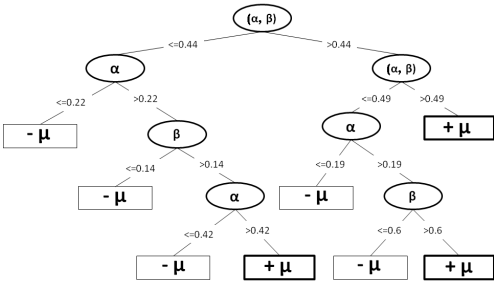

Figura 11 – Árvore de Decisão

A Figura 11 mostra um exemplo de uma árvore de decisão construída utilizando o algoritmo de aprendizado de máquina *J48 Classifier* (QUINLAN, 1993), que é um dos mais utilizados na literatura, por ter mostrado ótimos resultados em problemas de classificação. Esta

árvore de decisão foi construída baseada em um conjunto de treinamento contendo 2475 registros rotulados. A predição começa a partir do nodo raiz e percorre todos os níveis até as folhas, cada nodo indica um teste a ser executado com algum dos atributos preditivos e usa um *threshold* para classificar o resultado do teste como verdadeiro ou falso. Os atributos preditivos são $\alpha_{(q_i,e_j)}$, $\beta_{(D_i,ne_j)}$ e $avg(\alpha_{(q_i,e_j)}$.

Tabela 2 – Precisão da Árvore de Decisão

Taxa FP	**Precisão**	**Revocação**	**Medida-F**	**Classe**
0.004	0.967	0.762	0.853	μ
0.238	0.962	0.996	0.979	- μ

Os valores estatísticos de avaliação de precisão da árvore podem ser vistos na Tabela 2, que contém os valores da Taxa FP (taxa de falsos positivos), *precisão*, *revocação* e *Medida-F* para cada classe de predição (μ e $-\mu$). Além disso, o erro médio absoluto é 0.058, portando de acordo com este valor, esta árvore de decisão apresenta alta precisão.

Finalmente, o valor do escore final de similaridade é resultante da aplicação do fator de correção μ, classificado pela árvore de decisão, sobre o valor do *score de similaridade intermediário* $\delta_{(q_i,e_j)}$.

Definição 4.3.6 (Score Final de Similaridade)**.** *Seja* $\delta_{(\alpha,\beta)}$ *o escore intermediário de similaridade e* μ *o fator de correção, o escore final de similaridade* $\delta f'_{(\alpha,\mu)}$ *é calculado por:*

$$\delta f'_{(\delta,\mu)} = \delta \pm \mu \tag{4.7}$$

Com base no valor obtido com o cálculo do escore final de similaridade é criado o ranking de documentos para a consulta realizada.

No próximo capítulo, são apresentados os experimentos e resultados obtidos, realizados para demonstrar a viabilidade da proposta.

5 ANÁLISE EXPERIMENTAL

Com o objetivo de avaliar o método Nazca de maneira geral, um conjunto de experimentos foi executado. Este capítulo descreve a preparação dos dados, a realização desses experimentos e seus resultados. Para avaliar a precisão da proposta, os experimentos dividem-se em *avaliação dos termos indexados no CI*, para avaliar a qualidade da extração do contexto das fontes de dados e *avaliação de consultas*, para avaliar a qualidade do método proposto.

5.1 CONJUNTO DE DADOS E IMPLEMENTAÇÃO

Para a realização dos experimentos propostos, utilizou-se uma parte da base de dados do *DeepPeep* (BARBOSA et al., 2010), que é um repositório de 46.474 formulários *web*, indexados em sete domínios diferentes.Considerou-se quatro desses domínios na realização dos experimentos, *auto*, *book*, *airfare* e *movie*.

Cada formulário está armazenado em uma estrutura contendo os seus campos e também a URL da página *web* em que se encontra. As informações dos campos de cada formulário foram utilizadas para a construção do Índice de Elementos (EI). A partir das URLs, o conteúdo HTML de cada uma das páginas foi recuperado. Posteriormente este documento HTML sofreu um tratamento, onde toda a informação que representa a estrutura foi retirada, no caso todos os elementos que representam o formulário web. Assim o conteúdo restante, que representa o contexto, foi indexado no CI, retirando-se as informações irrelevantes, como: *tags* HTML, *stop words*, números e caracteres de pontuação. Para cada termo indexado no CI, foi identificado o documento de origem e o domínio, conforme o exemplo citado no Capítulo 4 na Figura 7.

Foi implementado um protótipo do método Nazca na linguagem Java. Os formulários do *DeepPeep* (BARBOSA et al., 2010) que são armazenados em um banco de dados relacional *MySql* [1] foram utilizados como *Índice de Elementos*. Para a criação do *Índice de Contexto*, utilizou-se os endereços das páginas onde encontram-se esses formulários. A partir desses endereços foi feita a busca e o download dessas páginas *web* onde encontram-se cada um dos formulários. Após isso foi

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

utilizado o *Jericho HTML Parser*[2] para extrair e separar em arquivos distintos as informações relevantes do contexto. Essas informações foram indexadas através da ferramenta *Apache Lucene*[3] criando assim o *Índice de Contexto*.

A implementação do algoritmo de busca por estruturas similares baseada no contexto utilizou os dados armazenados no banco de dados *DeepPeep* e também no *Índice de Contexto* criado. Para a implementação do cálculo da *Similaridade Básica* utilizou-se a função Q-grams da biblioteca SimMetrics [4]. Para auxiliar no resultado final do cálculo de similaridade foi construída uma árvore de decisão com o auxílio do software e da biblioteca de mineração de dados *Weka*[5]. Esta árvore de decisão foi construída baseada em um conjunto de treinamento contendo 2475 registros rotulados e utilizou-se a implementação do algoritmo *J48 Classifier* (QUINLAN, 1993) desta biblioteca.

5.2 AVALIAÇÃO DOS TERMOS DO CONTEXTO

Os experimentos desta seção, tem como objetivo avaliar a qualidade dos termos indexados no CI, que representam o contexto das estruturas das fontes de dados e assim, validar o processo de indexação destes termos. Para esta avaliação foi utilizado o *Freebase* (BOLLACKER et al., 2008)[6], uma base de dados escalável usada para estruturar o conhecimento humano no geral. Os dados do *Freebase* são criados, estruturados e mantidos colaborativamente. Atualmente o *Freebase* contém mais de 125.000.000 tuplas, mais de 4.000 tipos, e mais de 7.000 propriedades, por isso foi considerado uma referência para a avaliação do contexto. Dos domínios disponíveis no *Freebase*, os utilizados nos experimentos são: *book* com 1.893.860 instâncias, *film* com 232.312 instâncias, *automotive* com 13.439 instâncias e *travel* com 28.130 instâncias. A Tabela 3 mostra as correspondências entre esses domínios do *Freebase* e os domínios do CI. Para esta avaliação é considerado neste trabalho um valor maior ou igual a 50 % de termos encontrados no *Freebase* como significativo para que o contexto indexado seja considerado representativo do domínio.

Os experimentos de avaliação do contexto foram executados da seguinte forma: cada um dos termos indexados no CI em cada domínio

[2] http://jericho.htmlparser.net/docs/index.html
[3] http://lucene.apache.org/java/docs/index.html
[4] http://sourceforge.net/projects/simmetrics/
[5] http://www.cs.waikato.ac.nz/ml/weka/index.html
[6] http://www.freebase.com/

Tabela 3 – Termos por Domínio

Domínio Freebase	**Domínio CI**	**Total CI**	**Total encontrados Freebase**	**Não encontrados no Freebase**
film	*movie*	47.867	**30108**	17.759
automotive	*auto*	86.350	**47.029**	39.321
travel	*airfare*	65.357	**24.814**	40.543
book	*book*	65.535	**43.920**	21.615

foram pesquisados no *Freebase* em um domínio similar, conforme Tabela 3. A quantidade desses termos que foi encontrada no *Freebase* foi comparada com a quantidade total de termos indexados no CI para o domínio similar. Para que os termos indexados sejam considerados representativos do contexto de cada domínio, espera-se que a maior parte deles seja encontrada no *Freebase*, que considerou-se uma referência de conteúdo para esses experimentos.

5.2.1 Resultados

Os resultados absolutos obtidos com o experimento proposto são mostrados na Tabela 3 e a Figura 12 mostra o gráfico de barras com os valores percentuais. De acordo com esses valores, nos domínios *movie*, *auto* e *book*, os termos do CI, representando o contexto são significativos devido à grande quantidade deles que foi encontrada no *Freebase* em domínios similares. O oposto ocorre no domínio *airfare*, o que pode ter sido causado pelo fato de que o domínio mais similar encontrado no *Freebase* foi *travel* e de acordo com o esquema esse domínio é mais geral na hierarquia, contendo subdomínios, no entanto *airfare* não é um subdomínio de *travel*, desta forma muitos termos específicos não foram encontrados no *Freebase*.

5.3 AVALIAÇÃO DE CONSULTAS

A fim de avaliar a precisão das consultas realizadas a partir do método Nazca, foram executados experimentos comparativos. No primeiro experimento foram comparados os resultados da implementação *baseline*, que consiste somente no escore de similaridade básica (α), com

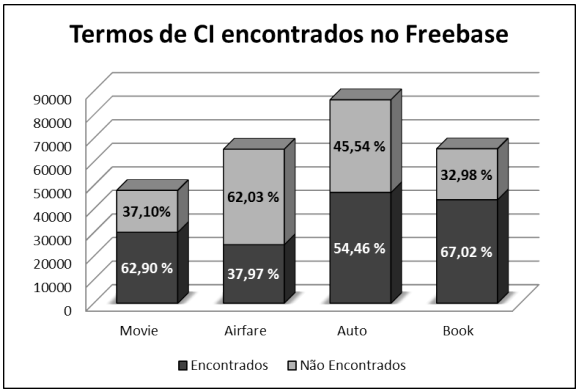

Figura 12 – Avaliação dos termos de CI

o método Nazca, que consiste no escore final de similaridade ($\delta f'$) definido anteriormente. Os demais experimentos foram feitos comparando o método Nazca com outra abordagem, que é o *Similarity Flooding* (MELNIK; GARCIA-MOLINA; RAHM, 2002).

5.3.1 Preparação dos dados

A avaliação foi feita a partir da execução de 10 consultas para cada domínio, totalizando 40 consultas. As consultas foram criadas manualmente, bem como o levantamento de resultados relevantes, neste levantamento considerou-se as estruturas (formulários web) armazenados nas fontes de dados e também o conteúdo das suas respectivas páginas que compõe os termos indexados no *Índice de Contexto*. que foram executadas para cada variável de comparação (*similaridade básica*, método Nazca e *Similarity Flooding*). Para cada avaliação comparativa das consultas, dois rankings foram criados, um para cada valor de similaridade, (*Similaridade Básica* X *Método Nazca*) e (*Método Nazca* X *Similarity Flooding*).

A abordagem do método *Nazca* é focada no casamento baseado em estruturas, por isso para os experimentos comparativos foi utilizada a abordagem do algoritmo de *Similarity Flooding* (MELNIK; GARCIA-MOLINA; RAHM, 2002), esse algoritmo utiliza os elementos e os relacionamentos de dois esquemas distintos no processo de casamento. A escolha da abordagem do *Similarity Flooding* justifica-se por ser um algoritmo de casamento de esquemas que obteve resultados significativos

para a área e inclusive originou outras abordagens, como o *SimRank* (JEH; WIDOM, 2002). Além disso, é possível adequar o formato dos dados utilizados no método *Nazca* ao formato de grafos, que é utilizado pelo algoritmo, possibilitando a execução dos dois algoritmos sobre o mesmo conjunto de dados e entradas. Isto não é possível em alguns dos trabalhos que utilizam termos como contexto, como por exemplo (BAGGA; BALDWIN, 1998; MANN; YAROWSKY, 2003; PEDERSEN; PURANDARE; KULKARNI, 2005) devido ao fato de utilizarem informações de contexto tanto na fonte de dados, quanto na entrada, no caso do método *Nazca* não há informações do contexto na entrada, apenas na fonte de dados.

No algoritmo do *Similarity Flooding*, os dados de entrada são compostos por dois grafos que representam os esquemas, o algoritmo então produz um mapeamento entre os nodos correspondentes nos dois grafos. Para realizar esses experimentos, a mesma base de dados utilizada como EI foi preparada considerando cada registro de formulário web e cada consulta de entrada como um grafo. Um exemplo ilustrativo da preparação destes dados pode ser visto na Figura 13, na representação da consulta o nome da estrutura é representado pelo nodo preenchido e os seus elementos pelos nodos vazados. Na base dados o nodo preenchido é representado pelo nome do domínio e os nodos vazados representam os campos ou elementos de cada um dos formulários web.

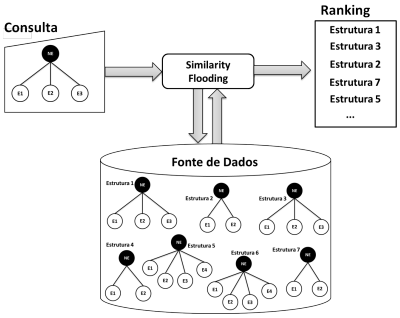

Figura 13 – Preparação dos dados para execução do algoritmo Similarity Flooding

Para cada consulta o algoritmo do *Similarity Flooding* retorna um mapeamento entre os nodos dos dois grafos de entrada, onde cada par de mapeamento possui um valor de similaridade, conforme Figura 14. A fim de comparar com a abordagem proposta, calculou-se a média

aritmética desses valores para posteriormente criar o ranking de saída e calcular as medidas de revocação e precisão.

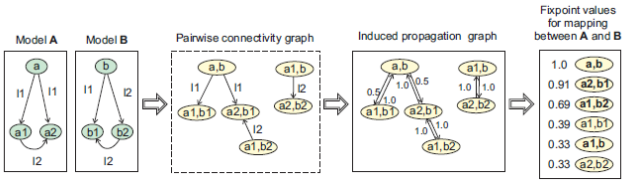

Figura 14 – Visão geral do algoritmo Similarity Flooding (MELNIK; GARCIA-MOLINA; RAHM, 2002)

Para a avaliação das consultas, utilizou-se medidas clássicas da área de recuperação de informação: *precisão* e *revocação* (SALTON; MCGILL, 1986). A *precisão* identifica a fração dos resultados obtidos para uma consulta que correspondem a resultados relevantes, ou seja, representam a mesma estrutura informada na consulta. A *revocação* corresponde à fração dos resultados relevantes para uma consulta que foram recuperados pelo método Nazca. Seja R_i o conjunto de resultados relevantes para a consulta e T_i o total de resultados relevantes recuperados, as fórmulas de revocação e precisão estão definidas a seguir:

$$precisão = \frac{R_i \cap T_i}{T_i} \qquad \text{revocação} = \frac{R_i \cap T_i}{R_i}$$

As curvas foram geradas de acordo com (BAEZA-YATES; RIBEIRO-NETO, 1999), encontrando a precisão em cada um dos pontos de revocação. A partir desses dados para cada uma das consultas, os gráficos foram feitos de acordo com as medidas de precisão mínima em um dos pontos.

5.3.2 Resultados

Na Figura 15 pode-se observar os resultados obtidos com as consultas executadas por domínio, comparando o *escore de similaridade básica* α com o *escore de similaridade final* $\delta f'$ obtido pelo método *Nazca*. Em todos os domínios o método *Nazca* obteve melhores resultados em relação ao valor do *escore similaridade básica*. Nos domínios *airfare* e *book*, a curva de similaridade básica obteve os piores resultados, indicando que as fontes de dados recuperadas no topo do ranking são falsos positivos. Isto ocorre porque nesses domínios, os elementos

das estruturas armazenadas no EI não são suficientemente representativos para as consultas. Assim, o uso de outras informações das fontes de dados, auxilia na caracterização dessas estruturas, por isso os melhores resultados podem ser vistos com os valores de escore de similaridade $\delta f'$ obtido pelo método Nazca.

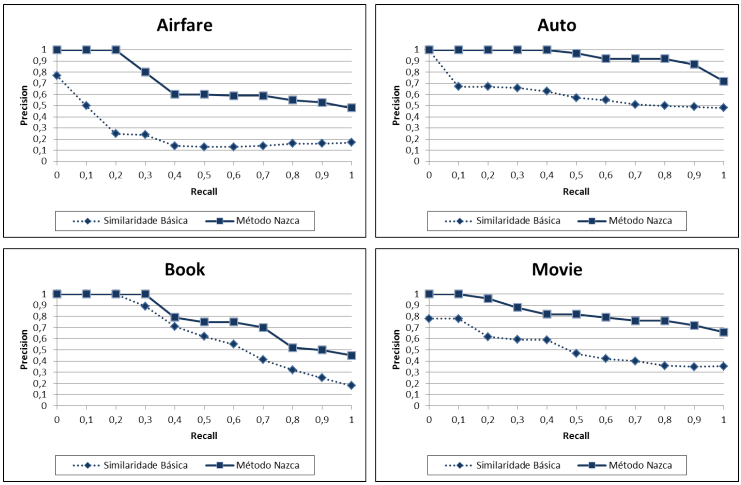

Figura 15 – Revocação X Precisão - Método Nazca $\delta f'$ e Similaridade Básica α

Em geral, os bons resultados obtidos com o método *Nazca* devem-se ao fato de que para as fontes de dados recuperadas, os valores do *escore de similaridade básica* não são suficientes para caracterizar a estrutura representada na fonte de dados em relação à consulta. Por exemplo, o objetivo da consulta *car(year, make, model)* é encontrar estruturas que representem o objeto *car* no mundo real, com os elementos, *year*, *make* e *model*. Entretanto, muitos dos resultados obtidos por esta consulta, representam na verdade objetos do tipo *auto parts* (peças) no mundo real, a sutil diferença entre os propósitos da consulta e a relevância dos resultados obtidos com a consulta pode ser identificada pelas informações do contexto.

A segunda avaliação é feita comparando os resultados do método Nazca com *Similarity Flooding*. Cada gráfico da Figura 16 mostra as curvas de revocação e precisão para cada domínio representando as consultas executadas com o método Nazca (linhas contínuas) e a abordagem *Similarity Flooding* (linhas pontilhadas).

A média dos valores de precisão de todos os domínios para o

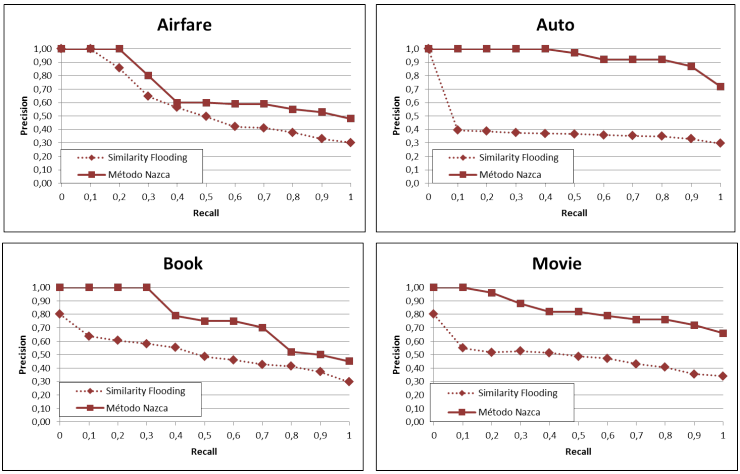

Figura 16 – Método Nazca X Similarity Flooding

método Nazca estão entre 0.5 e 0.7, cujos valores mostram melhor precisão se comparados aos mesmos obtidos para o Similarity Flooding, que obteve média de 0.3 nos valores de precisão. Um aumento significativo nos resultados pode ser observado em todos os domínios para a utilização do método Nazca, especialmente no domínio auto, que representa um aumento de 0,44 na precisão. Esta diferença pode ser observada porque das estruturas indexadas deste domínio, uma quantidade significativa representa "*auto parts*" (peça) e não "*auto*" (automóvel) no mundo real, assim os elementos dessas estruturas são muito similares. Considerando que o algoritmo do *Similarity Flooding* não utiliza outras informações além dos elementos da estrutura e relacionamentos, que são representativos apenas em esquemas completos, desta forma muitos registros são falsos positivos. Isto pode ser observado porque os relacionamentos são pouco significativos para casamentos em nível de estrutura, onde apenas a representação da estrutura é analisada, como ocorre nas consultas realizadas. Por outro lado, a utilização das informações de contexto utilizadas pelo método Nazca sugerem esta significativa diferença.

6 CONCLUSÃO E TRABALHOS FUTUROS

As inúmeras possibilidades de representações heterogêneas de estruturas similares em diferentes fontes de dados, motivam o estudo de soluções para o casamento desses dados. De acordo com a classificação de abordagens de casamentos de esquemas proposta por (RAHM; BERNSTEIN, 2001), o casamento em nível de estruturas caracteriza-se pela combinação dos elementos presentes nas estruturas. Como proposta de solução, o objetivo principal do método Nazca é a busca por estruturas similares, utilizando no processo informações adicionais das fontes de dados consideradas como contexto da estrutura.

O método Nazca difere dos trabalhos relacionados encontrados na literatura em alguns aspectos, que podem ser observados na tabela da Figura 5. A solução proposta não necessita de esquemas completos, ou seja, relacionamentos entre estruturas ou entidades, intervenção do usuário no processo de casamento ou bases de conhecimento para calcular o escore de similaridade. A partir de uma consulta informada como entrada é possível calcular a sua similaridade com a estrutura presente em uma fonte de dados utilizando um vetor de termos do contexto para tornar o processo mais exato.

O método Nazca é uma abordagem de casamento de esquemas em nível de estruturas que emprega o conceito de contexto da fonte de dados. O processo proposto para as buscas está dividido em duas etapas: a) indexação, em que o ambiente é preparado para as buscas através da criação dos índices de elementos e de contexto; b) processamento de consultas, etapa em que as consultas do usuário são processadas pelo motor de buscas a partir da estrutura de índices criada na etapa de indexação.

Alguns experimentos com os termos indexados no CI (índice de contexto) mostram que o processo de extração de informações do contexto das fontes de dados é efetivo, buscando as informações relevantes. Os experimentos de avaliação dos resultados das consultas mostram a precisão da abordagem através das métricas de revocação e precisão. Nas comparações com o algoritmo *Similarity Flooding* (MELNIK; GARCIA-MOLINA; RAHM, 2002), o método Nazca obteve os melhores resultados em todos os domínios analisados.

Existem aspectos que representam desafios na utilização do contexto para o processo de casamento de esquemas em nível de estrutura. Entre eles destacam-se:

1. Avaliar e melhorar desempenho e tempo de execução das etapas

de indexação e processamento de consultas.

2. Utilizar outros métodos de aprendizado de máquina além de árvores de decisão e realizar experimentos comparativos de precisão.

3. Criação de interface gráfica de consultas acessível ao usuário.

As pesquisas realizadas para a construção da abordagem proposta nesta dissertação resultaram e uma publicação na *VIII Escola Regional de Banco de Dados (ERBD 2012)* (Oliveira e Dorneles, 2012), com experimentos avaliando a qualidade do contexto indexado. Um poster no *SAC 2014 (29th Symposium On Applied Computing)*, com a apresentação da proposta e resultados obtidos. Um artigo completo no *ICCIT 2014: International Conference on Computer and Information Technology* com a apresentação do método Nazca e os experimentos de avaliação do contexto e das consultas.

REFERÊNCIAS

BAEZA-YATES, R. A.; RIBEIRO-NETO, B. *Modern Information Retrieval*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1999. ISBN 020139829X.

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 17th International Conference on Computational Linguistics - Volume 1*. Stroudsburg, PA, USA: Association for Computational Linguistics, 1998. p. 79–85. <http://dx.doi.org/10.3115/980845.980859>.

BARBOSA, L. et al. Creating and exploring web form repositories. In: *Proceedings of the 2010 international conference on Management of data*. New York, NY, USA: ACM, 2010. (SIGMOD '10), p. 1175–1178. ISBN 978-1-4503-0032-2. <http://doi.acm.org/10.1145/1807167.1807311>.

BERNSTEIN, P. A.; MADHAVAN, J.; RAHM, E. Generic schema matching, ten years later. *PVLDB*, v. 4, n. 11, p. 695–701, 2011. <http://dblp.uni-trier.de/db/journals/pvldb/pvldb4.html/BernsteinMR11>.

BLANCO, L. et al. Flint: Google-basing the web. In: *Proceedings of the 11th international conference on Extending database technology: Advances in database technology*. New York, NY, USA: ACM, 2008. (EDBT '08), p. 720–724. ISBN 978-1-59593-926-5. <http://doi.acm.org/10.1145/1353343.1353435>.

BOLLACKER, K. et al. Freebase: a collaboratively created graph database for structuring human knowledge. In: *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM, 2008. (SIGMOD '08), p. 1247–1250. ISBN 978-1-60558-102-6. <http://doi.acm.org/10.1145/1376616.1376746>.

DEY, A. K. Understanding and using context. *Personal Ubiquitous Comput.*, Springer-Verlag, London, UK, UK, v. 5, n. 1, p. 4–7, jan. 2001. ISSN 1617-4909. <http://dx.doi.org/10.1007/s007790170019>.

DORNELES, C. F.; GONçALVES, R.; MELLO, R. dos S. Approximate data instance matching: a survey. *Knowl. Inf. Syst.*, Springer-Verlag

New York, Inc., New York, NY, USA, v. 27, n. 1, p. 1–21, abr. 2011. ISSN 0219-1377. <http://dx.doi.org/10.1007/s10115-010-0285-0>.

DORNELES, C. F. et al. A strategy for allowing meaningful and comparable scores in approximate matching. *Inf. Syst.*, v. 34, n. 8, p. 673–689, 2009.

ELMAGARMID, A. K.; IPEIROTIS, P. G.; VERYKIOS, V. S. Duplicate record detection: A survey. *IEEE Trans. on Knowl. and Data Eng.*, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 19, n. 1, p. 1–16, jan. 2007. ISSN 1041-4347. <http://dx.doi.org/10.1109/TKDE.2007.9>.

FURCHE, T. et al. Amber: Automatic supervision for multi-attribute extraction. *CoRR*, abs/1210.5984, 2012. <http://dblp.uni-trier.de/db/journals/corr/corr1210.html/abs-1210-5984>.

HE, Y. et al. Crawling deep web entity pages. In: *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*. New York, NY, USA: ACM, 2013. (WSDM '13), p. 355–364. ISBN 978-1-4503-1869-3. <http://doi.acm.org/10.1145/2433396.2433442>.

JARO, M. A. Unimatch: A record linkage system: User's manual. *US Bureau of the Census*, 1976.

JEH, G.; WIDOM, J. Simrank: a measure of structural-context similarity. In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2002. p. 538–543. ISBN 1-58113-567-X. <http://doi.acm.org/10.1145/775047.775126>.

LEVENSHTEIN, V. I. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, American Institute of Physics., v. 10, n. 8, p. 707, 1966. <http://www.mendeley.com/research/binary-codes-capable-of-correcting-insertions-and-reversals/>.

LEVIN, F. H.; HEUSER, C. A. Using genetic programming to evaluate the impact of social network analysis in author name disambiguation. In: *AMW*. [S.l.: s.n.], 2010.

MANN, G. S.; YAROWSKY, D. Unsupervised personal name disambiguation. In: *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*. Stroudsburg, PA,

USA: Association for Computational Linguistics, 2003. (CONLL '03), p. 33–40. <http://dx.doi.org/10.3115/1119176.1119181>.

MELNIK, S.; GARCIA-MOLINA, H.; RAHM, E. Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In: *Proceedings of the 18th International Conference on Data Engineering*. Washington, DC, USA: IEEE Computer Society, 2002. (ICDE '02), p. 117–. <http://dl.acm.org/citation.cfm?id=876875.879024>.

MITCHELL, T. M. *Machine Learning*. 1. ed. New York, NY, USA: McGraw-Hill, Inc., 1997. ISBN 0070428077, 9780070428072.

PEDERSEN, T.; PURANDARE, A.; KULKARNI, A. Name discrimination by clustering similar contexts. In: *International Conference on Intelligent Text Processing and Computational Linguistics*. [S.l.]: Springer, 2005. p. 220–231.

PLOCH, D. Exploring entity relations for named entity disambiguation. In: *Proceedings of the ACL 2011 Student Session*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011. (HLT-SS '11), p. 18–23. ISBN 978-1-932432-89-3. <http://dl.acm.org/citation.cfm?id=2000976.2000980>.

QUINLAN, J. R. *C4.5: programs for machine learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN 1-55860-238-0.

RAHM, E.; BERNSTEIN, P. A. A survey of approaches to automatic schema matching. *The VLDB Journal*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, v. 10, n. 4, p. 334–350, dez. 2001. ISSN 1066-8888. <http://dx.doi.org/10.1007/s007780100057>.

SALTON, G.; MCGILL, M. J. *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, Inc., 1986. ISBN 0070544840.

SHEN, W. et al. Linden: linking named entities with knowledge base via semantic knowledge. In: *Proceedings of the 21st international conference on World Wide Web*. New York, NY, USA: ACM, 2012. p. 449–458. ISBN 978-1-4503-1229-5. <http://doi.acm.org/10.1145/2187836.2187898>.

SILVA, A. S. D. et al. Labeling data extracted from the web. In: *Proceedings of the OTM Confederated international conferences:*

CoopIS, DOA, ODBASE, GADA, and IS. Berlin, Heidelberg: Springer-Verlag, 2007. p. 1099–1116. ISBN 3-540-76846-7, 978-3-540-76846-3. <http://dl.acm.org/citation.cfm?id=1784607.1784701>.

STEFANIDIS, K.; KOUTRIKA, G.; PITOURA, E. A survey on representation, composition and application of preferences in database systems. *ACM Trans. Database Syst.*, ACM, New York, NY, USA, v. 36, n. 3, p. 19:1–19:45, ago. 2011. ISSN 0362-5915. <http://doi.acm.org/10.1145/2000824.2000829>.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. *Introduction to Data Mining*. Us ed. Addison Wesley, 2005. Hardcover. ISBN 0321321367. <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/0321321367>.

WANG, D. Z. et al. Automatic knowledge base construction using probabilistic extraction, deductive reasoning, and human feedback. In: *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2012. (AKBC-WEKEX '12), p. 106–110. <http://dl.acm.org/citation.cfm?id=2391200.2391220>.

WANG, J. et al. Entity matching: how similar is similar. *Proc. VLDB Endow.*, VLDB Endowment, v. 4, n. 10, p. 622–633, jul. 2011. ISSN 2150-8097. <http://dl.acm.org/citation.cfm?id=2021017.2021020>.

WENINGER, T.; JOHNSTON, T. J.; HAN, J. The parallel path framework for entity discovery on the web. *ACM Trans. Web*, ACM, New York, NY, USA, v. 7, n. 3, p. 16:1–16:29, set. 2013. ISSN 1559-1131. <http://doi.acm.org/10.1145/2516633.2516638>.

WHANG, S. E.; BENJELLOUN, O.; GARCIA-MOLINA, H. Generic entity resolution with negative rules. *The VLDB Journal*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, v. 18, n. 6, p. 1261–1277, dez. 2009. ISSN 1066-8888. <http://dx.doi.org/10.1007/s00778-009-0136-3>.