UNIVERSIDADE FEDERAL DO AMAZONAS

INSTITUTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

UM ESTUDO SOBRE O USO DE INFORMAÇÕES DE INSTÂNCIAS PARA O CASAMENTO DE ESQUEMAS NO DOMÍNIO DE COMÉRCIO ELETRÔNICO

Manaus

Setembro de 2017

UNIVERSIDADE FEDERAL DO AMAZONAS

INSTITUTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

LUÍSA DOS REIS E SILVA

UM ESTUDO SOBRE O USO DE INFORMAÇÕES DE INSTÂNCIAS PARA O CASAMENTO DE ESQUEMAS NO DOMÍNIO DE COMÉRCIO ELETRÔNICO

Dissertação apresentada ao Programa de Pós-Graduação em Informática do Instituto de Computação da Universidade Federal do Amazonas como requisito parcial para a obtenção do grau de Mestre em Informática.

ORIENTADOR: ALTIGRAN SOARES DA SILVA

Manaus

Setembro de 2017

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

Silva, Luísa dos Reis e
S586u Um Estudo sobre o Uso de Informações de Instâncias para o Casamento de Esquemas no Domínio de Comércio Eletrônico / Luísa dos Reis e Silva. 2017
68 f.: il. color; 31 cm.

Orientador: Altigran Soares da Silva
Dissertação (Mestrado em Informática) - Universidade Federal do Amazonas.

1. Casamento de Esquemas. 2. Comércio Eletrônico. 3. Instâncias. 4. Integração de Dados. 5. Aprendizado de Máquina. I. Silva, Altigran Soares da II. Universidade Federal do Amazonas III. Título

PODER EXECUTIVO
MINISTÉRIO DA EDUCAÇÃO
INSTITUTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

UFAM

FOLHA DE APROVAÇÃO

"Um Estudo sobre o Uso de Informações de Instâncias para o Casamento de Esquemas no Domínio de Comércio Eletrônico"

LUÍSA DOS REIS E SILVA

Dissertação de Mestrado defendida e aprovada pela banca examinadora constituída pelos Professores:

Prof. Altigran Soares da Silva - PRESIDENTE

Profa. Tayana Uchôa Conte - MEMBRO INTERNO

Prof. Moisés Gomes de Carvalho - MEMBRO EXTERNO

Profa. Viviane Pereira Moreira - MEMBRO EXTERNO

Manaus, 12 de Setembro de 2017

Av. Rodrigo Otávio, 6.200 - Campus Universitário Senador Arthur Virgílio Filho - CEP 69077-000 - Manaus, AM, Brasil
Tel. (092) 3305 1193 E-mail: secretariappgi@icomp.ufam.edu.br www.ppgi.ufam.edu.br

Dedico este trabalho à minha família.

Agradecimentos

Agradeço a Deus por sempre me ajudar e me guiar. Sem Ele eu não estaria aqui. À minha família pelo amor, carinho, paciência e apoio. Aos membros do laboratório de BDRI por colaborarem na obtenção de informações, coleta de dados, esclarecimento de dúvidas e apresentação de novas abordagens. Agradeço ao meu orientador, D.Sc. Altigran Soares da Silva, pela oportunidade de desenvolver essa pesquisa, pelo apoio e orientação. Por fim, agradeço a todos que de alguma forma me ajudaram nesse processo.

"Mas buscai primeiro o reino de Deus, e a sua justiça, e todas estas coisas vos serão acrescentadas."

(Bíblia Sagrada, Mateus 6:33)

Resumo

Integração de dados é a tarefa de combinar dados de diversas fontes e representá-los em um único conjunto de dados. Uma tarefa fundamental para integração de dados é o casamento de esquemas, definido como a tarefa de encontrar correspondências semânticas entre elementos de dois esquemas distintos. Recentemente, esse problema tem sido estudado no domínio de comércio eletrônico, por ser um domínio de grande importância prática no dia-a-dia das pessoas. Vários métodos têm sido propostos na literatura com o objetivo de automatizar essa tarefa. Os métodos utilizam diferentes tipos de informação, como informação dos nomes e da estrutura dos elementos dos esquemas analisados. Neste trabalho, procuramos identificar se as informações de instâncias são mais significativas para os métodos de casamento de esquemas no domínio de comércio eletrônico. Para tanto, verificamos o comportamento de três métodos de casamento de esquemas ao adicionarmos essas informações: COMA, que utiliza heurísticas fixas para combinação de *matchers*; ALMa, que utiliza Aprendizado Ativo; e RFSM, que utiliza aprendizado de máquina supervisionado. Nos experimentos, percebemos que ao utilizar informação de instância os métodos apresentaram melhorias nos seus resultados, principalmente na precisão e medida-f. Verificamos também que os métodos não necessitam ter uma frequência alta dessa informação para que elas contribuam com os resultados.

Palavras-chave: Casamento de Esquemas, Comércio Eletrônico, Instâncias, Integração de Dados, Aprendizado de Máquina.

Abstract

The Data integration task seeks to combine data from various sources and represent them in a single data set. Schema Matching is a key task to solve this problem, and is defined as the task of finding semantic correspondences between elements of two distinct schemes. Recently, this problem have been studied in the e-commerce domain, since it has great practical importance in people's daily lives. Several methods have been proposed in the literature aiming to automate the schema matching task. These methods use different types of information, such as information on the names and structure of the elements of the analyzed schemas. In this research, we try to identify if the information of instances are more significant to the schema matching methods in the e-commerce domain. With this purpose, we verify the behavior of three schema matching methods by adding the instances information: COMA, which uses fixed heuristics to match matchers; ALMa, which uses Active Learning; And RFSM, which uses supervised machine learning. In the experiments, we noticed that by using instances information all methods presents improvements, mainly in precision and measure-f. We also verify that the methods do not require a high frequency of this information to contribute with the results.

Keywords: Schemma Matching, E-Commerce, Instances, Data Integration, Machine Learning.

Lista de Figuras

1.1 Exemplo Casamento de Esquemas. 2

3.1 Distribuição de Ofertas nas bases de dados 19
3.2 Distribuição de Atributos nas bases de dados 21
3.3 Distribuição das Classes de Atributos por categoria no *BDRI*. 22
3.4 Distribuição das Classes de Atributos por categoria no *Dexter*. 22

4.1 Experimentos com a Função de Similaridade Categorical. 38
4.2 Experimentos com a Função de Similaridade Cosseno. 38
4.3 Experimentos com a Função de Similaridade Fledex Value-Based. 39
4.4 Experimentos com a Função de Similaridade Jaccard. 39
4.5 Experimentos com a Função de Similaridade Jaro-Winkler. 40
4.6 Experimentos com a Função de Similaridade Jensen Shannon. 40
4.7 Experimentos com a Função de Similaridade KLD. 41
4.8 Experimentos com a Função de Similaridade TFIAF. 41
4.9 Experimentos com a Função de Similaridade Camberra. 42
4.10 Experimentos com a Distância Euclideana. 42
4.11 Experimentos com a Função de Similaridade Fledex Content. 43
4.12 Experimentos com a Função de Similaridade Manhattan. 43
4.13 Experimentos com a Função de Similaridade Numerical. 44

5.1 Resutados do F-Measure na base de dados *BDRI* 51
5.2 Resutados do F-Measure na base de dados *Dexter* 52
5.3 Frequência de *Matchers* na base de dados *BDRI*. 54
5.4 Frequência de *Matchers* base de dados *Dexter*. 54
5.5 Resultados do método COMA na BDRI 56
5.6 Resultados do método COMA na Dexter 57
5.7 Resultados do método ALMa na BDRI 58
5.8 Resultados do método ALMa na Dexter 59

5.9 Resultados do método RFSM para quantidade da variáveis 60
5.10 Resultados do método RFSM na BDRI 61
5.11 Resultados do método RFSM na Dexter 62

Lista de Tabelas

3.1 Detalhes das Categorias da Base de Dados. 17
3.2 Exemplo de Atributos Locais e Pares Atributos-Valores. 17
3.3 Detalhes das Categorias da base de dados Dexter. 17
3.4 Exemplo de Atributos Globais e Locais. 19
3.5 Detalhes dos Atributos no BDRI. 20
3.6 Detalhes dos Atributos na Dexter. 20
3.7 Características das bases de dados. 23

4.1 Exemplo - Frequência de Termos. 28
4.2 Exemplo - Cálculo tf e idf para A_x 29
4.3 Exemplo - Cálculo tf e idf para A_y 29
4.4 Exemplo 2 - Frequência de Termos. 30
4.5 Exemplo 2 - Cálculo tf e idf para A_y 30
4.6 Exemplo 2 - Cálculo tf e idf para A_x 31
4.7 Funções de Similaridade Avaliadas 36
4.8 Métodos de Agregação Avaliados 37
4.9 Amostras Utilizadas para Validar os Métodos de Agregação. 40
4.10 Resultados da Validação do Single Link. 42
4.11 Resultados da Validação do Complete Link. 43
4.12 Resultados da Validação do Avarage Link. 44
4.13 Resultados da Validação do Valor mais Frequente. 44
4.14 Resultados da Validação do AVG_E. 45
4.15 Resultado da Validação das Funções de Similaridade 45
4.16 Resultado da Validação das Funções de Similaridade 45

5.1 Resultados dos experimentos na base de dados *BDRI* 51
5.2 Resultados dos experimentos na base de dados *Dexter* 52

Sumário

Agradecimentos **xi**

Resumo **xv**

Abstract **xvii**

Lista de Figuras **xix**

Lista de Tabelas **xxi**

1 Introdução **1**

2 Trabalhos Relacionados **7**

2.1 Métodos de Casamento de Esquemas Baseados em Heurísticas Fixas . . 7

2.2 Métodos de Casamento de Esquemas Baseados em Aprendizagem de Máquina 9

2.3 Métodos Baseados em Instâncias 12

2.4 Métodos Utilizados neste Trabalho 13

3 Bases de Dados para Experimentação **15**

3.1 Construção da Base de Dados 15

3.1.1 Base de dados *BDRI* 16

3.1.2 Base de Dados *Dexter* 17

3.2 Análise das Bases de Dados 18

3.2.1 Distribuição de Ofertas 18

3.2.2 Atributos Locais e Globais 19

3.2.3 Distribuição de Atributos 20

3.2.4 Classificação dos Atributos 20

3.2.5 Bases de Dados por Tarefas de Casamento de Esquemas 23

3.3 Considerações Finais 23

4 Abordagem para Utilização de Informações de Instâncias 25
4.1 Visão Geral . . . 25
4.2 Funções de Similaridade Utilizadas . . . 26
4.2.1 Categóricos e Multicategóricos . . . 26
4.2.2 Numéricos . . . 31
4.2.3 Dimensionais . . . 31
4.2.4 Booleanos . . . 33
4.3 Métodos de Agregação . . . 34
4.4 Seleção das Funções de Similaridade . . . 36
4.4.1 Revisão da Literatura . . . 36
4.4.2 Validação das Funções de Similaridade . . . 37
4.5 Considerações Finais . . . 44

5 Resultados Experimentais 47
5.1 Métricas de Avaliação . . . 47
5.2 Configuração dos Experimentos . . . 48
5.2.1 Métodos Utilizados nos Experimentos . . . 48
5.2.2 Matchers Utilizados . . . 48
5.2.3 Configurações Utilizadas nos Experimentos . . . 49
5.3 Resultados Experimentais . . . 50
5.4 Utilização dos Matchers . . . 51
5.5 Estudo de Parâmetros utilizados nos Métodos de Casamento de Esquema 54
5.6 Considerações Finais . . . 57

6 Conclusões e Trabalhos Futuros 63

Referências Bibliográficas 65

Capítulo 1

Introdução

De forma geral, integração de dados é a tarefa de combinar dados de diversas fontes distintas com o objetivo de fornecer uma visão unificada desses dados. Realizar integração de dados é uma tarefa complexa, uma vez que existem muitas formas diferentes de modelar os mesmos conceitos em banco de dados [Doan et al., 2012]. Uma tarefa fundamental para integração de bancos de dados é a identificação de estruturas lógicas que representam os mesmos conceitos do mundo real. Nesse contexto, o casamento de esquemas é a tarefa de identificar correspondências semânticas entre elementos de dois esquemas distintos [Rahm & Bernstein, 2001, Doan et al., 2012].

Por exemplo, na Figura 1.1 ilustramos os esquemas dos catálogos de produtos de três lojas distintas: *Lojas Americanas*, *Extra* e *Ricardo Eletro*. Estes catálogos apresentam informações sobre uma mesma câmera fotográfica "Nikon D5300". Os atributos correspondentes entre os esquemas são indicados pelas linhas tracejadas. Entre os esquemas, pode-se encontrar atributos com as mesmas formas de representação, como "Resolução" nos esquemas das *Lojas Americanas* e *Extra*. Também pode-se encontrar atributos com representações diferentes, mas com a mesma semântica, como "Resolução" nas *Lojas Americanas* e "Megapixel (mp)" do *Ricardo Eletro*. Outro caso ocorre quando um único atributo de um esquema representa mais de um atributo em outro esquema, como ocorre com o atributo "Dimensões aproximadas do produto - cm (axlxp)" nas *Lojas Americanas*, que corresponde a "Altura", "Largura" e "Profundidade" no *Extra*.

Um método para casamento de esquemas deve retornar um mapeamento entre os atributos correspondentes entre dois esquemas fornecidos como entrada. No exemplo, para os casos das *Lojas Americanas* e *Ricardo Eletro*, o mapeamento seria: Resolução - Megapixel (mp); Slide Show - Modo Slide show. Esse mapeamento de esquemas pode ser utilizado em aplicações como geração de esquemas globais, reescrita de consultas

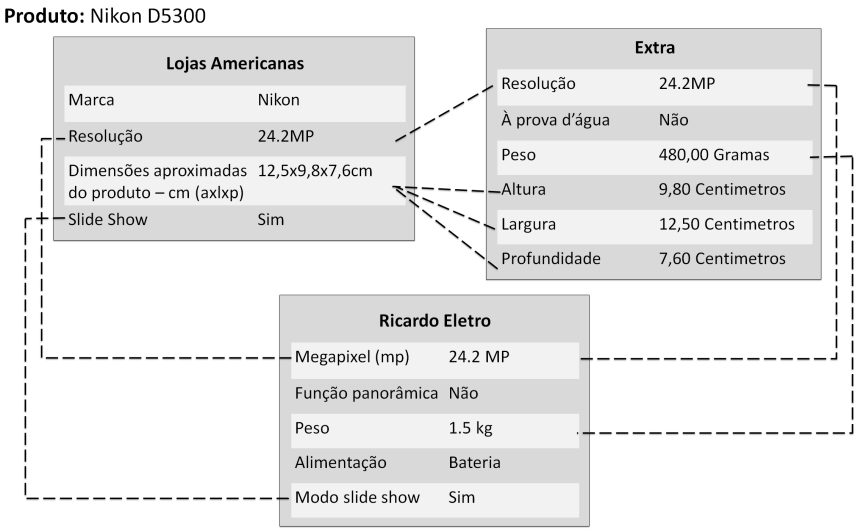

Figura 1.1. Exemplo Casamento de Esquemas.

em fontes heterogêneas e eliminação de dados duplicados [Doan et al., 2012].

Recentemente, o problema de casamento de esquemas tem sido estudado no domínio de comércio eletrônico por ser este um domínio muito importante atualmente, o que pode ser comprovado pelo grande número de lojas online existentes. Nesse domínio, o casamento de esquemas auxilia em tarefas como geração de catálogos de produtos unificados, síntese de produtos [Nguyen et al., 2011], normalização de atributos de produto da Web [Wong et al., 2011], entre outras.

De forma geral, os esquemas usados para descrever produtos em sites de comércio eletrônico possuem um alto grau de heterogeneidade entre si. Sites de comércio eletrônico têm como característica a descrição de uma grande quantidade de produtos organizados em diversas categorias. Apesar de um mesmo site seguir um padrão para todas as categorias de produtos, cada site de comércio eletrônico tem a sua própria forma de descrever seus produtos [Köpcke et al., 2012]. Além disso, apesar de estarem descrevendo o mesmo atributo, nem sempre os esquemas de dois sites terão correspondências entre todos os seus atributos, uma vez que a quantidade de atributos utilizadas para descrever um produto é diferente nos dois esquemas [Wong et al., 2011].

Ao longo dos anos, diversos métodos têm sido propostos com o objetivo de automatizar a tarefa de casamento de esquemas [Melnik et al., 2002, Nguyen et al., 2011, Bernstein et al., 2011, Do & Rahm, 2002]. Estes métodos exploram diferentes tipos de informação, como características linguísticas de nomes de atributos e informações auxiliares, tais como dicionários de dados, reuso de correspondências encontradas em mapeamentos anteriores e uso de informações fornecidas pelo usuário. Os métodos mais conhecidos na literatura utilizam heurísticas fixas [Do & Rahm, 2002,

Melnik et al., 2002]. Entretanto, essa abordagem não é flexível o suficiente para lidar com a variedade de domínios e de elementos que compõem os esquemas.

Para contornar esse problema foram propostas abordagens que utilizam aprendizagem de máquina. Essa estratégia obtém bons resultados, uma vez que se adequa a vários tipos de domínio. Entretanto, a maioria dos métodos necessita de um conjunto de exemplos rotulados para realizar o seu treinamento. Alguns métodos que utilizam essa abordagem são o LSD [Doan et al., 2001], YAM [Duchateau et al., 2009], ALMa [Rodrigues et al., 2015] e RFSM [Rodrigues, 2017].

De forma geral, é esperado que qualquer método para casamento de esquemas tenha um grau de imperfeição no seu processo [Gal, 2006], considerando a enorme ambiguidade e heterogeneidade encontrada na descrição dos seus dados. Não se espera que um único mecanismo de mapeamento consiga identificar o mapeamento correto para qualquer conceito possível em um banco de dados. Por isso, a tarefa de casamento de esquemas requer muitas vezes a intervenção de especialistas humanos que possuam amplo conhecimento sobre o domínio e a semântica dos esquemas envolvidos [Gal, 2006]. Mesmo sendo realizada por um especialista, essa tarefa ainda é propensa a erros e praticamente inviável quando aplicada a grandes esquemas.

Grande parte dos métodos existentes para casamentos de esquema utilizam funções, conhecidas como *matchers*, que estimam um valor de similaridade entre dois elementos de dois esquemas distintos. Geralmente, o resultado retornado por um *matcher* é um valor entre 0 e 1, em que 0 indica que o par de elementos não é uma correspondência, e 1 representa um alto nível de correspondência. Os métodos aplicam uma série de *matchers* no conjunto de pares de atributos de dois esquemas e combinam esses resultados para estabelecer quais pares representam uma correspondência e serão retornados como o mapeamento [Rahm & Bernstein, 2001, Rodrigues et al., 2015].

De forma geral, os *matchers* podem ser classificados em categorias [Rahm & Bernstein, 2001]. Algumas são: *matchers* baseados em esquema, *matchers* baseados em instâncias, *matchers* baseados em estrutura e *matchers* baseados em restrições. *Matchers* baseados em esquema consideram informações sobre elementos de esquema, como nomes e descrições, e também utilizam algum conhecimento sobre o domínio da aplicação. *Matchers* baseados em instâncias determinam a semelhança entre elementos do esquema ao identificar as características dos valores das suas instâncias. Nesse tipo de *matcher*, apenas os valores das instâncias são considerados, não utilizando, portanto, informação de esquema. *Matchers* baseados em estrutura referem-se a combinações de elementos correspondentes que aparecem juntos em uma estrutura. *Matchers* baseados em restrições utilizam informações como faixas de valores numéricos ou médias e padrões de caracteres.

Uma abordagem para casamento de esquemas em comércio eletrônico é apresentada por Nguyen et al. [Nguyen et al., 2011]. Os autores propõem um método que procura solucionar o problema de síntese de produtos em catálogo de produtos para comércio eletrônico. Síntese de produtos é definido como a tarefa de identificar novos produtos através das ofertas disponibilizadas pelas lojas e adicioná-las ao catálogo. O casamento de esquemas é realizado em uma das etapas da síntese de produtos, ocorrendo entre os elementos do esquema de uma oferta e os elementos presentes no catálogo de produtos do sistema. Esse método leva em consideração o histórico de associações entre ofertas e produtos contidas no sistema e utiliza as informações de instância das ofertas para realizar o casamento de esquema. Apesar da tarefa de casamento de esquemas ter obtido bons resultados nos seus experimentos, o método se restringe a agrupar informações em um único catálogo.

Considerando as características do comércio eletrônico e as dificuldades conhecidas nos métodos de casamento de esquemas, observamos a necessidade de estudar mais detalhadamente esse domínio. Especificamente, considerando o trabalho apresentado por Nguyen et al. [Nguyen et al., 2011], acreditamos que o uso de *matchers* baseados em instância podem ser mais informativos para os métodos de casamento de esquema nesse domínio. *Matchers* de instância também obtiveram bons resultados nos trabalhos apresentados por Bilke e Naumann [Bilke & Naumann, 2005] e De Carvalho et al. [De Carvalho et al., 2013].

Neste trabalho procuramos analisar se o uso de informações de instâncias contribui nos resultados dos métodos de casamento de esquema, em comparação com os resultados sem considerar essas informações. Para tanto, avaliamos o comportamento de três métodos de casamento de esquemas ao adicionarmos essas informações.

Especificamente, apresentamos neste trabalho os seguintes resultados:

- Geração de duas bases de dados para experimentação em casamento de esquemas no domínio de comércio eletrônico;

Antes de realizar os experimentos, construímos duas bases de dados no domínio de comércio eletrônico: *BDRI*, gerada a partir de lojas brasileiras, e *Dexter*, gerada a partir de lojas estrangeiras e contém informações em inglês. As bases de dados possuem características representativas dos sites de comercio eletrônico, tais como grande volume de dados e variedade de categorias e lojas. Além disso, as bases de dados também contém problemas típicos de catálogos de comércio eletrônico, tais como, atributos que só existem em uma loja; atributos com nomes similares, mas significados diferentes, como dimensões do produto e dimensões da embalagem; representação do

mesmo atributo por nomes diferentes, como revestimento e cor; atributos com valores binários, entre outros.

- Uma abordagem para utilização de informações de instâncias em métodos de casamento de esquemas para comércio eletrônico;

Propusemos uma abordagem para utilização de informações de instância nesse domínio e realizamos experimentos com as bases de dados em três métodos de casamento de esquemas: COMA [Do & Rahm, 2002], ALMa [Rodrigues et al., 2015, Rodrigues, 2013] e RFSM [Rodrigues, 2017], que utilizam abordagens heurística, aprendizado ativo e aprendizado de máquina supervisionado, respectivamente.

- Avaliação experimental do uso de informações de instâncias nos métodos COMA [Do & Rahm, 2002], ALMa [Rodrigues et al., 2015, Rodrigues, 2013] e RFSM [Rodrigues, 2017], que utilizam abordagens heurística, aprendizado ativo e aprendizado de máquina supervisionado, respectivamente.

Nossos resultados mostram que a utilização de *matchers* de instância contribuíram nos resultados dos métodos. Nos experimentos pode-se verificar melhoras nos resultados da precisão e medida-F. Observamos, também, que a frequência de *matchers* de instância utilizados pelos métodos não se sobressai aos *matchers* de esquema. Isso indica que estes métodos não precisam utilizar muitos *matchers* de instância, mas suas informações trazem de fato, vantagens para os resultados.

Além deste capítulo introdutório, este trabalho está organizado como segue. No Capítulo 2 apresentamos alguns dos trabalhos que consideramos como mais relevantes para problema de casamento de esquema. No Capítulo 3 apresentamos as bases de dados de comércio eletrônico geradas como parte deste trabalho e que foram utilizadas para experimentação. No Capítulo 4, é apresentada a nossa abordagem para utilização das informações de instâncias nos métodos de casamento de esquemas. No Capítulo 5 são apresentados os experimentos realizados e os resultados obtidos. Por fim, o Capítulo 6 encerra o texto desta dissertação com a apresentação das conclusões e sugestões para trabalhos futuros.

Capítulo 2

Trabalhos Relacionados

Neste capítulo apresentamos uma visão geral de alguns métodos e técnicas existentes para o problema de casamento de esquemas que consideramos significativos para o nosso trabalho. O problema de casamento de esquemas tem sido estudado por muitos anos e muitas estratégias têm sido propostas para sua solução. Na literatura, além de livros-texto sobre o tema [Bellahsene et al., 2011, Doan et al., 2012], existem artigos que apresentam uma revisão detalhada sobre vários métodos propostos para a solução deste problema [Bernstein et al., 2011, Rahm & Bernstein, 2001].

Grande parte dos métodos de casamento de esquemas utilizam um conjunto de funções, conhecidas como *matchers*, que estimam a similaridade entre os elementos de dois ou mais esquemas. Os métodos mais conhecidos para casamento de esquemas utilizam *matchers* que usam informações dos próprios esquemas. No entanto, em alguns problemas representativos a área de comércio eletrônico, informações extraídas das instâncias tem se mostrado muito úteis. Assim, neste capítulo, revisamos também alguns trabalhos da literatura que utilizaram *matchers* de instância. Uma das principais abordagens existentes para casamento de esquemas consiste em combinar o resultado de vários *matchers*, para decidir quais elementos de dois esquemas serão mapeados. Nesse capítulo iremos abordar duas estratégias para combinação de *matchers*: uma baseada em heurísticas fixas e uma baseada em aprendizagem de máquina.

2.1 Métodos de Casamento de Esquemas Baseados em Heurísticas Fixas

Métodos que usam heurísticas fixas seguem um conjunto de passos fixos para encontrar o mapeamento de dois esquemas. Alguns dos métodos mais conhecidos na lite-

ratura que utilizam essa estratégia são Similarity Flooding [Melnik et al., 2002], CUPID [Madhavan et al., 2001] e COMA [Do & Rahm, 2002].

O Similarity Flooding [Melnik et al., 2002] utiliza pontos fixos para procurar similaridades semânticas entre os elementos de dois esquemas. O método representa cada esquema como um grafo dirigido e rotulado. O grafo é usado para calcular, de forma iterativa, os pontos fixos, cujos resultados indicam quais nós de um grafo são similares aos nós do outro grafo. Inicialmente, o método realiza um mapeamento utilizando *matchers* de *string* simples, que realizam comparações de prefixos e sufixos comuns entre os elementos dos esquemas. Após esse mapeamento inicial, é aplicado o algoritmo *SFJoin*, proposto pelo método. O algoritmo baseia-se na suposição de que, quando quaisquer dois elementos de dois esquemas são considerados similares, a similaridade dos elementos adjacentes a eles aumenta. Assim, durante as iterações do método, a similaridade inicial de quaisquer dois nós, calculada do mapeamento inicial, se propaga através dos grafos. No final da execução do algoritmo, é aplicado um limiar para selecionar quais pares serão retornados pelo método. Porém, os autores ressaltam a necessidade do usuário fazer correções nos resultados retornados. Por isso, o método é descrito como uma ferramenta de auxílio ao casamento de esquemas.

O CUPID [Madhavan et al., 2001] combina dois tipos de *matchers* para realizar o casamento de esquemas: *matchers* linguísticos e *matchers* estruturais. *Matchers* linguísticos utilizam elementos nominais, tipos de dados e domínios. Também são utilizados dicionários para identificar abreviações, acrônimos e sinônimos. *Matchers* estruturais procuram relacionamentos semânticos entre os elementos dos esquemas. Por exemplo, em um esquema no formato XML, se dois subelementos são correspondentes, seus elementos raiz também são considerados correspondentes. Primeiramente, o método aplica esses *matchers* nos elementos de dois esquemas. Com o resultado obtido, o método combina os valores dos *matchers* através de média ponderada, onde os pesos são definidos pelo usuário. É importante ressaltar que o método utiliza recursos externos, como dicionários, de forma extensa, o que pode ser problemático, dependendo do domínio, se esses recursos não estiverem disponíveis.

O método COMA [Do & Rahm, 2002] aplica uma série de *matchers* existentes em uma biblioteca para calcular a similaridade entre os elementos de dois esquemas. A matriz resultante desse cálculo é agregada em uma única matriz de acordo com um dos seguintes critérios: *Max*, *Weighted*, *Average* e *Min*. *Max* retorna o valor máximo de similaridade obtido por qualquer *matcher* para um par de elementos dos esquemas. O *Weighted* aplica pesos a cada matchers, de acordo com a importância de cada um, retorna uma soma ponderada dos seus valores de similaridade. *Average* retorna a similaridade média obtida por todos os *matchers*, considerando todos igualmente

importantes. *Min* seleciona o menor valor de similaridade retornado por qualquer matcher.

Após a agregação, o método seleciona a direção da correspondência, entre *LargeSmall*, *SmallLarge* ou *Both*. Considerando dois esquemas $S1$ e $S2$, tais que $|S2| \leq |S1|$, a direção representa a forma com que as correspondências serão analisadas. Na *LargeSmall*, os elementos de $S1$, esquema maior, são classificados em relação a cada elemento de $S2$, esquema menor. A *SmallLarge* ocorre de forma contrária a *LargeSmall*, sendo a classificação dos elementos feita com base nos elementos de $S2$ em comparação aos elementos de $S1$. Em *Both* ambas direções *LargeSmall* e *SmallLarge* são consideras; além disso, um par de elementos será considerado como verdadeiro apenas se for identificado em ambas direções.

Por fim, as correspondências consideradas verdadeiras são selecionadas de acordo com um dos critérios de seleção: *Threshold*, *MaxN* e *MaxDelta*. O *Threshold* os pares que mostram uma semelhança superior a um determinado valor limite. O *MaxN* seleciona os n elementos de $S1$ que obtiveram similaridade maxima. No *MaxDelta*, um dos elementos do primeiro esquema com similaridade máxima é determinado como um candidato a correspondência verdadeira e todos os elementos que diferirem dessa similaridade por, no máximo, um valor de tolerância d, especificado como um valor absoluto ou relativo, também são selecionados.

2.2 Métodos de Casamento de Esquemas Baseados em Aprendizagem de Máquina

Métodos que utilizam aprendizagem de máquina têm obtido bons resultados, uma vez que essa estratégia se adéqua a vários tipos de domínios. Entretanto, a maioria dos métodos dessa categoria precisa de exemplos rotulados para realizar o seu treinamento. Nessa categoria encontram-se os métodos LSD [Doan et al., 2001], YAM [Duchateau et al., 2009], ALMa [Rodrigues et al., 2015, Rodrigues, 2013] e RFSM [Rodrigues, 2017].

LSD [Doan et al., 2001] foi um dos primeiros métodos que utilizaram a abordagem de aprendizagem de máquina para casamento de esquemas. O método solicita exemplos iniciais de mapeamentos semânticos rotulados pelo usuário e os utiliza para treinar uma série de *learners*, que exploram diferentes informações das bases de dados, como os nomes ou a estrutura dos esquemas. Em seguida, um componente chamado de *meta-learner* seleciona e atribui pesos aos *learners* de acordo com a sua performance no treinamento. O modelo retornado nesse processo é usado, então, para realizar o

mapeamento dos pares de elementos restantes. O método apresenta apenas soluções 1-1 e, assim como o CUPID, também depende de informação externa para cada domínio, o que pode apresentar problemas caso essas informações não estejam disponíveis.

O método YAM [Duchateau et al., 2009], é apresentado como uma "fábrica" de métodos de casamento de esquema. O método usa aprendizagem de máquina para gerar *matchers* dedicados para uma tarefa de dados especificada, de acordo com a entrada de dados fornecida pelo usuário. A premissa desse método é de que os algoritmos que combinam medidas de similaridade fornecem resultados diferentes dependendo do cenário dos esquemas em que são aplicados. Ele é dividido em duas etapas: treinamento dos modelos e seleção final dos modelos. Na fase de treinamento dos modelos, o YAM treina com diferentes algoritmos de aprendizagem e testa o modelo gerado com uma base de conhecimento, que contém vinte classificadores, disponibilizados pelo Weka[1], trinta medidas de similaridade; e o histórico de correspondências anteriores entre esquemas no mesmo domínio. Na seleção final dos modelos é feita a eleição do melhor modelo o método dedicado de acordo com a sua acurácia na fase de treinamento. O método escolhido, então, é usado para realizar o casamento de esquemas do cenário em que foi treinado.

O método ALMa [Rodrigues et al., 2015, Rodrigues, 2013] utiliza aprendizado ativo [Cohn et al., 1994, Settles, 2012] para combinar *matchers* em casamento de esquemas. Aprendizado ativo é uma técnica de amostragem de dados que seleciona um subconjunto dos exemplos mais informativos para treinar um classificador. Dado um conjunto de valores resultantes da execução de vários *matchers*, o ALMa utiliza árvores de decisão para identificar quais as correspondências verdadeiras entre os elementos dos esquemas. Dado um conjunto de valores resultantes da execução de vários *matchers*, o ALMa utiliza árvores de decisão para identificar quais as correspondências verdadeiras entre os elementos dos esquemas. O ALMa recebe como entrada um conjunto de dados que se constituem em todos os pares possíveis de elementos entre dois esquemas e, para cada par, um vetor com os resultados de cada *matchers* aplicados neles. Com essas informações, o método executa seu algoritmo de aprendizado ativo. O aprendizagem ativo do método é dividido em quatro etapas: Seleção, Treinamento, Eleição do Comitê e Votação.

Na Seleção, o método seleciona alguns pares para serem rotulados pelo usuário. Após o usuário rotular esses pares, eles são levados para a etapa de Treinamento, onde são usados para compor o conjunto de treino que será usado para gerar um conjunto de árvores de decisão. No final dessa etapa, cada árvore tem uma pontuação

[1]http://www.cs.waikato.ac.nz/ ml/weka

que reflete o seu grau de confiança. Na etapa de Eleição do Comitê as árvores com as maiores confianças são selecionadas para compor o comitê. Este comitê decidirá quais pares, não-rotulados, são correspondências verdadeiras. Por fim, na Votação, o método utiliza os votos fornecidos pelo comitê para decidir se um par é verdadeiro ou não. Se todos os membros do comitê concordarem que um par é verdadeiro, então esse par será considerado uma correspondência e será retirado da base de dados à rotular. Porém, se os membros do comitê estiverem em desacordo, será solicitado que o usuário rotule o par, que será, então, utilizado para retreinar as árvores na próxima rodada. Esse processo se repete até que todas as árvores concordem que não existem mais correspondências, rotulando todos os pares remanescentes como falsos.

O RFSM (Random Forest Schema Matching) [Rodrigues, 2017] é um método de casamento de esquemas baseado em *Random Forest* [Breiman, 2001] que está em desenvolvimento no nosso grupo de pesquisa [Rodrigues, 2017]. Embora o método ainda não tenha sido publicado formalmente, escolhemos utilizá-lo como um método representativo de métodos supervisionados, uma vez que experimentos realizados com esse método apresentaram resultados similares aos melhores métodos baseados em aprendizagem de máquina encontrados na literatura, mas aos quais não tivemos acesso à implementação. O método RFSM utiliza a técnica de aprendizagem de máquina *Random Forest* para, assim como o método ALMa, combinar *matchers* em casamento de esquemas. O método recebe como parâmetros o resultado da aplicação de vários *matchers* nos pares de elementos de dois esquemas. Então, ele seleciona aleatoriamente uma parte desses dados para serem usados no seu treinamento, sendo que os pares selecionados precisam estar previamente rotulados. Em seguida, o método gera as árvores de decisão que irão compor a floresta. Para gerar essas árvores, o método seleciona m *matchers* randomicamente dentre um conjunto de p variáveis disponíveis. Na maioria dos casos, o valor de m é definido como $\sqrt{p}$. Dentre esses m *matchers*, são selecionados os mais informativos para compor os nós. Após a composição da floresta, cada árvore apresentará um voto para cada par de elementos analisados. A classificação de cada par é feita computando o voto majoritário da floresta.

Um método de casamento de esquemas que utiliza programação genética para resolver mapeamentos complexos em casamento de esquemas foi proposto por De Carvalho et al. [De Carvalho et al., 2013]. Mapeamentos complexos são mais trabalhosos, uma vez que os métodos precisam encontrar combinações de atributos em um esquema com combinações de atributos em outro esquema. Um exemplo de mapeamento complexo é o atributo "Dimensões do produto", que se relaciona com a combinação de atributos "Largura", "Altura" e "Profundidade". Utilizando programação genética, cada combinação de atributos é organizada em forma de árvore e, representando os indiví-

duos da população de cada esquema. No método proposto, cada candidato a correspondência representa um indivíduo da população que irá evoluir. A evolução se inicia com uma população simulada de candidatos corretos e evolui para uma população final de candidatos que o método considerará correto. Para avaliar cada indivíduo está organizado como uma tripla $m = \langle d_A, d_B, \alpha \rangle$, onde d_A e d_B são as combinações de atributos analisadas e α é um *matcher* aplicado sobre seus valores. O método utiliza duas estratégias para utilizar o *matcher*: orientada a entidade e orientada a valor. A orientada a entidade utiliza *winkler* como função de similaridade e concatena o resultado da aplicação dessa função a várias instâncias. A orientada a valor aplica o modelo vetorial, conhecido em recuperação de informação [Rijsbergen, 1979], sobre os valores das instâncias, agrupados como uma *bag of words*.

2.3 Métodos Baseados em Instâncias

Em nosso trabalho, procuramos explorar o uso de informações de instâncias para resolver o problema de casamento de esquemas em comércio eletrônico. Entretanto, o domínio de comércio eletrônico é pouco explorado no contexto de casamento de esquemas.

Nguyen et al. [Nguyen et al., 2011] propõem um método para solucionar o problema de síntese de produtos em catálogo de produtos de comércio eletrônico. Síntese de produtos é definida como a tarefa de identificar novos produtos através das ofertas disponibilizadas pelas lojas fornecedoras e adicioná-las ao catálogo. O casamento de esquemas é realizado em uma das etapas da síntese de produtos, ocorrendo entre os elementos do esquema de uma oferta e os elementos presentes no catálogo de produtos do sistema.

O método utiliza um histórico de associações entre ofertas e produtos contido no sistema para realizar o seu treinamento. O método considera que utilizar apenas os nomes dos atributos na tarefa de casamento de esquemas não é suficiente, por isso, utiliza as informações das instâncias, com a premissa de que termos similares possuem contextos similares. O método reúne todas as palavras contidas nos valores de cada atributo em um catálogo de produtos e aplica os *matchers* Jensen-Shannon [Manning & Schütze, 1999] e Coeficiente de Jaccard [Jaccard, 1912]. Após esses cálculos, o método utiliza regressão linear para selecionar os pares corretos e gerar o mapeamento.

Bilke e Naumann [Bilke & Naumann, 2005] não exploram o domínio de comércio eletrônico. Porém, o método utiliza a informações de instâncias para realizar casamento

de esquemas. Os autores propõem o método DUMAS, que procura duplicatas entre as instâncias de dois esquemas para realizar o casamento de esquemas. Duplicatas são definidas como múltiplas representações do mesmo objeto do mundo real dentro de um conjunto de objetos. No método, as duplicatas são utilizadas para identificar automaticamente os atributos correspondentes entre dois esquemas.

O algoritmo primeiro descobre as duplicatas entre as instâncias de dois esquemas e seleciona as K melhores duplicatas entre elas. O método então aplica a medida estatística TF-IDF [Baeza-Yates & Ribeiro, 2012] entre os valores de cada atributo presente nas duplicatas. É gerada uma matriz com os resultados do TF-IDF para cada duplicata e o método aplica um limiar para sumarizar todos os resultados em uma única matriz com os resultados que serão retornados para o usuário.

2.4 Métodos Utilizados neste Trabalho

Nesta pesquisa selecionamos três métodos para avaliar o uso das informações de instâncias para casamento de esquemas em comércio eletrônico: COMA [Do & Rahm, 2002], que utiliza heurísticas fixas; ALMa [Rodrigues et al., 2015], que utiliza aprendizado ativo e RFSM [Rodrigues, 2017], que utiliza aprendizado supervisionado.

Selecionamos esses métodos por permitirem o uso de informação de instância em estratégias diferentes e por terem obtido bons resultados nos seus experimentos originais. Com isso, procuramos verificar se o uso de informações de instâncias resultam em melhorias para os métodos, independente da abordagem utilizada por eles.

Capítulo 3

Bases de Dados para Experimentação

Neste capítulo apresentamos as bases de dados que utilizamos para os experimentos realizados neste trabalho. Optamos por construir essas bases de dados, pois não encontrarmos na literatura uma base de dados pública com dados no domínio comércio eletrônico que apresentasse as características necessárias para avaliar a tarefa de casamento de esquemas. Procuramos, assim, criar bases de dados que fossem representativas das características dos sites de comércio eletrônico, tais como grande volume de dados e uma boa variedade de categorias e lojas. Acreditamos que as bases de dados construídas podem contribuir para estudos sobre casamentos de esquema e integração de dados, por isso a deixamos disponível publicamente [1]. No decorrer do capítulo, será explicado como a coleta dos dados que compõem essas bases foi realizada e será apresentada uma análise das características das bases de dados.

3.1 Construção da Base de Dados

Foram construídas duas bases de dados. A primeira, que iremos chamar de *BDRI*, foi construída no decorrer dessa pesquisa; e a segunda, aqui referida como *Dexter*, foi construída a partir de uma coleção de dados de comércio eletrônico disponibilizada por Qiu et al. [Qiu et al., 2015]. Destacamos que nas duas bases tomou-se o cuidado de converter os valores para as mesmas unidades de medida, bem como de eliminar ruídos nos valores textuais.

Para a construção dos gabaritos usados na avaliação dos métodos de casamento

[1] https://drive.google.com/drive/folders/1dZHPQV3Gu9OAwFDtqL7E256ORYPY3VSA?usp=sharing

de esquemas, foi realizado o mapeamento manual dos pares verdadeiros de cada tarefa de casamento, tanto para a base de dados *BDRI*, quanto para a *Dexter*. A seguir descrevemos a coleta de dados de cada base de dados.

3.1.1 Base de dados *BDRI*

A base de dados *BDRI* foi obtida através da API disponibilizada pelo site de comparação de preços Buscapé[2]. Optou-se por utilizar este site pelo fato dele conter uma vasta quantidade de categorias e lojas afiliadas, uma vez que se trata de um dos maiores sites de comparação de preços do Brasil.

O site organiza as ofertas dos produtos em categorias, como *Telefonia*, e subcategorias, como *Smartphones*. Foram selecionadas as subcategorias que possuíam os maiores números de produtos ofertados, com a finalidade de obter uma maior diversidade de informações. Além disso, para facilitar a coleta e permitir uma melhor análise do comportamento dos métodos durante a execução de experimentos, procurou-se selecionar categorias que estivessem presentes em muitas lojas e com a maior quantidade de lojas em comum. Para tanto, foram selecionadas oito das lojas mais populares do Brasil: *Lojas Americanas*, *Casas Bahia*, *Extra*, *Fast Shop*, *Ricardo Eletro*, *Ponto Frio*, *Shoptime* e *Submarino*. As lojas também foram escolhidas de acordo com a quantidade de ofertas que apresentavam. Além dos dados de lojas individuais, coletamos também dados sobre os produtos ofertados pelo próprio site do *Buscapé*. Assim, o *Buscapé* foi considerado como uma nona loja. Foram selecionadas quatro categorias: Câmera Digital, Celulares e Smartphones, Notebooks e Televisões. No decorrer do texto, as categorias serão referenciadas respectivamente através das siglas: CAM, CEL, NOTE e TV.

A Tabela 3.1 apresenta detalhes de cada categoria de produtos que compõe a *BDRI*. A coluna "Lojas" indica o número de lojas que contêm ofertas em cada categoria. A coluna "Ofertas" apresenta o número total de ofertas de produtos na base de cada categoria, somando as ofertas de todas as lojas. A coluna "Atributos Locais" indica o número total de atributos utilizados na descrição das ofertas, sem considerar repetição de atributos com o mesmo nome e somando todos os atributos que apareceram em cada loja. Finalmente, a coluna "Pares Atributo-Valor" apresenta a quantidade de valores encontrados para os atributos em todas as ofertas das lojas.

Para ilustrar o que são atributos locais e pares de atributo-valor, temos como exemplo a Tabela 3.2. As Lojas Americanas apresenta o atributo "Modelo Processador" e a Casas Bahia apresenta o atributo "Processador", sendo que estes dois atributos

[2]http://developer.buscape.com.br/portal/developer/, acessado em 15 de Novembro de 2015

locais representam o mesmo conceito, que nesse caso é o modelo do processador de um notebook. O par atributo-valor representa o valor de um determinado atributo local para uma oferta de produto disponibilizada por uma loja. Nesse exemplo, o par atributo-valor para o atributo *modelo* nas Lojas Americanas para o produto "Samsung ATIV Book 3 NP370E4K" é $\langle ModeloProcessador; 5005U \rangle$.

Categoria	Ofertas	Atributos Locais	Pares Atributo-Valor
CAM	22733	164	21707
CEL	24415	218	16933
NOTE	41268	185	33862
TV	18424	209	11438

Tabela 3.1. Detalhes das Categorias da Base de Dados.

Loja	Produto	Atributo Local	Par Atributo-Valor
Lojas Americanas	Samsung ATIV Book 3 NP370E4K	Modelo Processador	$\langle ModeloProcessador; 5005U \rangle$
Casas Bahia	Samsung ATIV Book 3 NP370E4K	Processador	⟨Processador; Intel Core i3-5005U Dual Core 2.0 GHz >

Tabela 3.2. Exemplo de Atributos Locais e Pares Atributos-Valores.

3.1.2 Base de Dados *Dexter*

As informações contidas nessa base estão em inglês, tanto para os nomes dos atributos, como nos valores e medidas utilizadas. A base contém dados de três categorias: Câmeras, Monitores e Tvs, que serão referenciados como CAMDX, MONDX e TVDX, respectivamente. Cada categoria contém informações de aproximadamente 334 lojas. Porém, para facilitar a coleta e análise dos dados, selecionamos três lojas bem conhecidas em comércio eletrônico: Alibaba, Amazon e Walmart.

A Tabela 3.3 apresenta os detalhes das categorias da base de dados *Dexter*. Percebe-se que a quantidade de atributos locais é maior que a quantidade de atributos locais da base de dados *BDRI*, mas a quantidade de pares atributo-valor é equivalente.

Categoria	Ofertas	Atributos Locais	Pares Atributo-Valor
CAMDX	1552	879	26075
MONDX	815	321	5219
TVDX	1455	494	6056

Tabela 3.3. Detalhes das Categorias da base de dados Dexter.

3.2 Análise das Bases de Dados

A seguir é apresentado um estudo sobre as bases de dados. Temos como objetivo identificar as principais características de cada categoria para avaliar, posteriormente, seus efeitos nos métodos de casamento de esquema.

3.2.1 Distribuição de Ofertas

Analisamos a distribuição das ofertas nas bases de dados com o objetivo de verificar se ela abrange uma grande quantidade de ofertas. Na Figura 3.1 observamos os gráficos bloxplot da variação das ofertas entre as lojas presentes nas bases de dados *BDRI* e *Dexter*.

O objetivo desses gráficos é verificar a distribuição dos dados, considerando o centro dos dados, através da mediana; a amplitude dos dados; a simetria do conjunto de dados e a presença de *outliers*, ou seja, valores discrepantes.

A amplitude dos dados, indica o intervalo mínimo e máximo de ofertas presentes nas lojas. No gráfico, os valores mínimos e máximos são representados como as caudas do retângulo e o retângulo é delimitado pelo quartil 1 e quartil 3. Nos gráficos, a *BDRI* possui amplitudes variáveis. A categoria NOTE apresenta maior amplitude, ou seja, maior variabilidade de ofertas. A CAM apresenta a menor amplitude. Na base *Dexter*, a amplitude das três categorias é aproximada, indicando uma variabilidade semelhante dos dados nas categorias.

Através da mediana dos dados, representada por uma linha horizontal dentro do retângulo, podemos identificar a simetria dos dados, ou seja, o nível de semelhança das informações analisadas. Uma distribuição simétrica tem a mediana no centro do retângulo. Se a mediana é próxima do quartil 1, então, os dados são positivamente assimétricos. Se a mediana é próxima do quartil 3 os dados são negativamente assimétricos. Nos gráficos, na *BDRI*, as categorias CAM e TV apresentaram assimetrias positivas e a categoria CEL assimetria negativa. NOTE foi a única categoria que apresentou dados simétricos. Na *Dexter* todas as categorias apresentaram assimetria, sendo a CAMDX e MONDX positiva e a TVDX negativa.

As categorias CAM e NOTE da *BDRI* apresentaram um único valor discrepante dos demais, apresentado na figura como um ponto externo ao gráfico. Isso ocorre porque, nessas categorias, uma loja em específico possui mais ofertas que as demais. A *Dexter* não apresentou valores discrepantes.

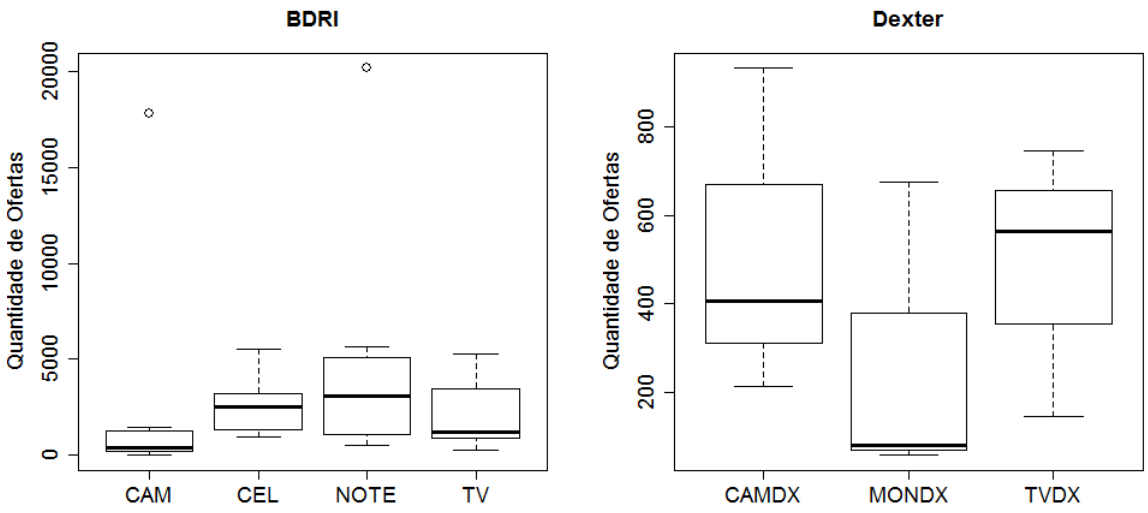

Figura 3.1. Distribuição de Ofertas nas bases de dados

	Atributo Global	**Atributos Locais**
Agrupado	Modelo do Processador	Modelo Processador, Processador
Não Agrupado	Idiomas do Menu	Idiomas do Menu

Tabela 3.4. Exemplo de Atributos Globais e Locais.

3.2.2 Atributos Locais e Globais

Dependendo da loja em que é apresentado, um mesmo atributo pode ser representado de formas diferentes. Para analisar essa situação, realizamos um estudo sobre os atributos. Os atributos com mais de uma representação foram separados em grupos, de forma manual, de acordo com a sua semântica. Na Tabela 3.4 é apresentado um exemplo de como as informações foram mapeadas. Os atributos que corresponderam ao mesmo conceito foram agrupados e receberam um nome genérico, caso de *Modelo do Processador*, e os que não se encaixaram em nenhum grupo permaneceram com o mesmo nome, caso de *Idiomas do Menu*.

O número total de atributos de cada categoria da *BDRI* é apresentado na coluna "Atributos Locais", assim como na Tabela 3.5. O total de grupos é apresentado na coluna "Grupos de Atributos". Os atributos que não se encaixaram em nenhum grupo são apresentados na coluna "Atributos não Agrupados". Na coluna "Atributos Globais" é apresentado o total de grupos formados mais o total de atributos não agrupados. Por fim, a coluna "Atributos por grupo" indica a média de atributos locais que foram agrupadas. A terminologia de Atributos Locais e Globais é a mesma utilizada por Li et al. [Li et al., 2012].

Categoria	Atributos Locais	Grupos de Atributos	Atributos não Agrupados	Atributos Globais	Média de Atributos por Grupo
CAM	164	41	66	107	5
CEL	218	44	27	71	4
NOTE	185	41	13	54	3
TV	209	62	19	81	4

Tabela 3.5. Detalhes dos Atributos no BDRI.

Categoria	Atributos Locais	Grupos de Atributos	Atributos não Agrupados	Atributos Globais	Média de Atributos por Grupo
CAMDX	879	22	784	786	7
MONDX	321	13	283	287	6
TVDX	494	16	430	410	5

Tabela 3.6. Detalhes dos Atributos na Dexter.

Na Tabela 3.6 podemos observar que a quantidade de atributos por categoria na *Dexter* é bem maior que a *BDRI*, bem como a quantidade de atributos globais. No caso da *Dexter*, poucos grupos foram formados, por isso a quantidade de atributos globais é maior.

3.2.3 Distribuição de Atributos

Em seguida analisamos a distribuição de atributos globais entre as lojas de cada categoria das bases de dados. Os resultados podem ser observados na Figura 3.2.

Na *BDRI*, as categorias possuem grande variabilidade de atributos, sendo a maior da categoria TV e a menor da categoria CEL. Porém, todas apresentaram assimetria positiva.

Na *Dexter*, a variabilidade de atributos também é ampla, sendo maior na categoria CAMDX. E todas elas apresentam assimetria positiva, assim como a *BDRI*.

Essa assimetria ocorre nas bases de dados, pois, dependendo da loja, as informações nas ofertas podem ser detalhadas, com uma quantidade maior de atributos; ou mais objetivas, apresentando apenas informações básicas dos produtos.

3.2.4 Classificação dos Atributos

Para classificar os atributos de cada categoria das bases de dados foi usado como base a classificação proposta por Hoffmann et al. [Hoffmann et al., 2015, Hoffmann, 2016]. Os autores definem quatro classes de atributos em Comércio Eletrônico: Categórico, Multicategórico, Dimensional e Numérico. Neste trabalho consideramos também a classe de atributo Booleano.

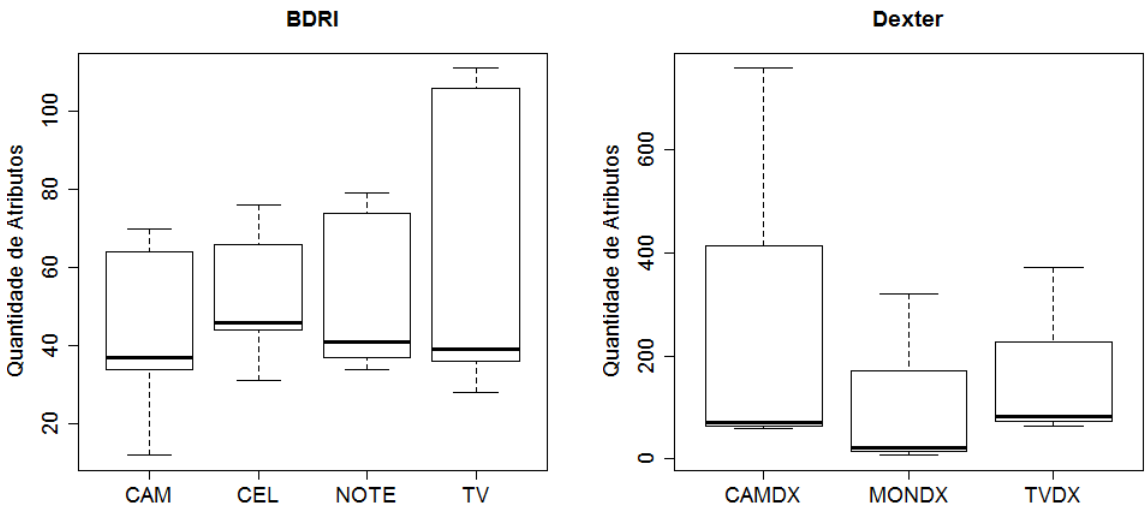

Figura 3.2. Distribuição de Atributos nas bases de dados

Os atributos Categóricos são atributos que possuem um único valor, sendo que este pertence a um conjunto fechado de valores. Este é o caso do atributo "Marca", que pode ter valores como "Sony" ou "Toshiba", por exemplo. Atributos Multicategóricos são aqueles cujos valores são uma lista de valores Categóricos. Por exemplo, o atributo "Idiomas do menu" que possui como valor a lista "alemão, francês, italiano, inglês, português, espanhol, polonês". Atributos Booleanos são aqueles que apresentam apenas valores "sim" ou "não", como ocorre no caso de "touch screen", que identifica uma função disponível para um celular e recebe o valor "sim", para informar que o celular em questão possui essa função. Atributos Dimensionais englobam atributos como "dimensões do produto", que indicam a largura, altura e volume dos produtos. Por fim, atributos Numéricos, que possuem um número como valor. Por exemplo, o atributo "Resolução", de uma câmera recebe o valor "15 MP".

Na Figura 3.3 observamos a cobertura dessas classes de atributos para a *BDRI*. Percebemos que a classe mais frequente é a Categórica, em todas as bases, e a segunda mais frequente é a Multicategórica. Ambos são atributos textuais e, nesses casos, *matchers* que analisam semântica podem gerar bons resultados. Todas as categorias, com exceção da NOTE, possuem uma grande quantidade de atributos Booleanos. Nessa classe a informação do valor do atributo não traz vantagem para o método. Atributos Numéricos também tem uma frequência alta nas categorias. Porém, atributos Dimensionais são os menos utilizados, uma vez que a maioria das ofertas apenas apresenta "Dimensões do Produto" ou "Dimensões da Embalagem". Os atributos Numéricos variam mais, aparecendo como "Voltagem" ou "Quantidade de Pilhas", por exemplo.

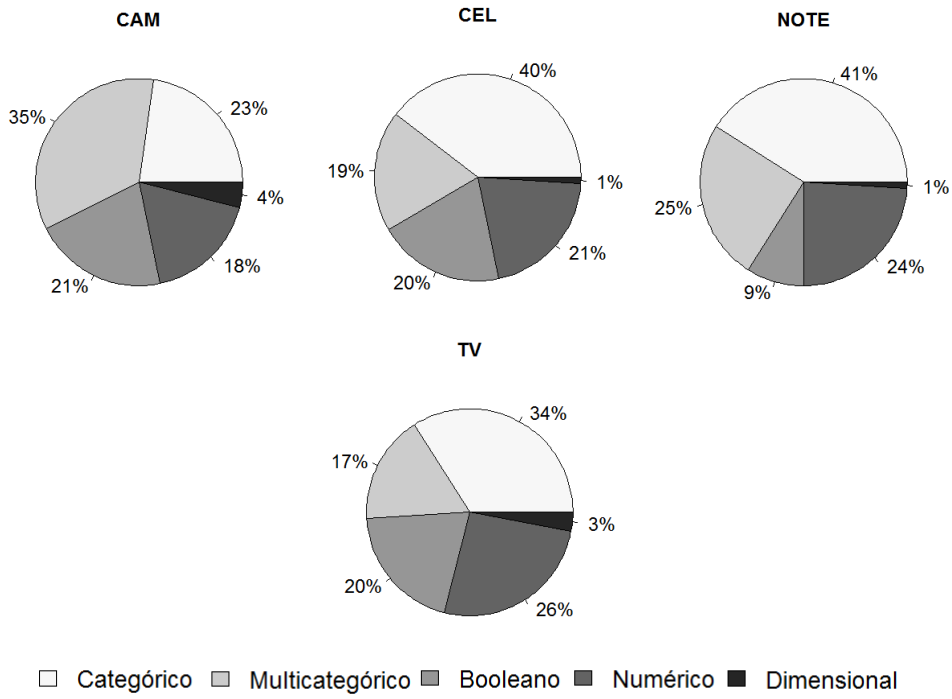

Figura 3.3. Distribuição das Classes de Atributos por categoria no *BDRI*.

Na *Dexter*, observamos na Figura 3.4 que a maior frequência de atributos também são das classes Categórico e Multicategórico, e menor de atributos Dimensionais. Entretanto, a quantidade de atributos Booleanos é menor nessa base. A quantidade de atributos Numéricos atinge apenas 3% na categoria CAMDX, mas nas outras bases ocorrem em maior frequência, de forma semelhante à BDRI.

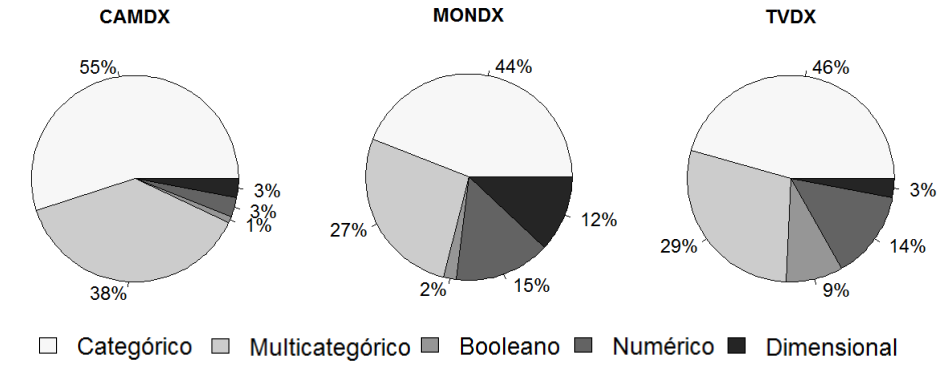

Figura 3.4. Distribuição das Classes de Atributos por categoria no *Dexter*.

Observando as duas bases, podemos perceber que a maioria dos valores apresentados em comércio eletrônico são textuais, Categóricos e Multicategóricos. Além disso, também possuem uma grande quantidade de atributos Numéricos, o que permite uti-

lizar funções que utilizem normalização.

3.2.5 Bases de Dados por Tarefas de Casamento de Esquemas

Os experimentos realizados no Capítulo 5, foram executados por tarefa de casamento. Considerando que os métodos de casamento de esquemas realizam a comparação de dois esquemas por vez e que na nossa base de dados possuimos vários esquemas, fizemos todas as combinações possíveis de esquemas da base, separados por categoria. Cada par de esquemas é considerada uma tarefa de casamento de esquemas. A Tabela 3.7 apresenta um sumário das características das bases de dados nesse contexto.

A base *Dexter* possui um volume de informações maior que a base *BDRI* e possui uma quantidade de atributos maior, como pode ser observado na linha "Média de Atributos por Esquema" na Tabela 3.7. Podemos verificar também que a quantidade de correspondências, ou seja, pares de elementos de esquema, existentes é bem superior a quantidade de correspondências verdadeiras presentes no esquemas. Então, dentre todas as opções possíveis de pares, poucas são as respostas que devem ser retornadas como verdadeiras pelos métodos estudados nessa dissertação, tornando essa uma atividade desafiadora.

Observa-se também que, apesar da base *Dexter* ser maior que a *BDRI*, a média de candidatos a correspondência e a média de atributos verdadeiros nelas é semelhante, o que permite que as bases possam ser analisadas de forma semelhante e que o desempenho dos métodos irá depender primordialmente da quantidade de atributos verdadeiros que ele será capaz de retornar.

	BDRI				**Dexter**		
	CAM	**CEL**	**NOTE**	**TV**	**CAMDX**	**MONDX**	**TVDX**
Número de tarefas de casamento	36	36	36	36	3	3	3
Média de candidatos a correspondência por tarefa	1895	2706	2710	3395	2834	262	1641
Média de correspondências verdadeiras para encontrar	21	23	20	29	38	17	29
Média de atributos em cada esquema	11	13	13	15	297	117	173

Tabela 3.7. Características das bases de dados.

3.3 Considerações Finais

Considerando todas as análises presentes nesse capítulo, concluímos que ambas bases de dados seriam adequadas a execução dos experimentos, uma vez que mostraram-se representativas tanto em questão de categorias, quanto ao volume de informação

presente e de pares possíveis de esquemas e atributos para análise dos métodos de casamento de esquemas estudados nessa dissertação.

Capítulo 4

Abordagem para Utilização de Informações de Instâncias

Nesse capítulo apresentamos a nossa abordagem para utilização das informações de instâncias no casamento de esquemas no domínio de comércio eletrônico.

4.1 Visão Geral

Neste trabalho consideramos que, em cada categoria, cada atributo se enquadra em apenas uma das classes de atributos definidas no Capítulo 3: Categóricos, Multicategóricos, Booleanos, Numéricos e Dimensionais.

Utilizamos conjuntos para representar as informações de instâncias de cada atributo. Dessa forma, para cada atributo A_i, pertencente a loja L na categoria C, temos associado um conjunto $V_i = \{v_1^i, \ldots, v_n^i\}$, com todos os valores v_j^i, sendo $j = \{1...n\}$, desse atributo no catálogo da loja L.

Sendo A_x e A_y dois atributos de esquemas distintos, desejamos verificar a similaridade entre A_x e A_y. Para isso, seguimos os seguintes passos:

1. Verificamos se A_x e A_y pertencem a mesma classe de atributos. Se não pertencem, avaliamos que eles não podem ser similares;

2. Se pertencem à mesma classe, uma função de similaridade específica para essa classe será usada para avaliar a similaridade entre os valores dos atributos contidos nos respectivos conjuntos V_x e V_y;

3. Os resultados obtidos pela comparação dos valores são agregados, para avaliar a similaridade entre os atributos A_x e A_y;

Neste capítulo apresentaremos as funções de similaridades utilizadas no Passo 2, na Seção 4.2, e os métodos de agregação utilizados no Passo 3, na Seção 4.3. Na Seção 4.4 apresentamos um estudo que realizamos com o objetivo de selecionar as funções de similaridade e os métodos de agregação utilizados neste trabalho.

4.2 Funções de Similaridade Utilizadas

Nesta seção descrevemos as funções de similaridade utilizadas para calcular a similaridade entre os valores dos atributos. Essas funções foram selecionadas através de um estudo que será descrito posteriormente na Seção 4.4.

Cada função de similaridade pode ser aplicada para uma ou mais classes de atributos. Representamos esas classes pelas siglas C, M, N, D e B, que simbolizam as classes Categórico, Multicategórico, Numérico, Dimensional e Booleano, respectivamente, de acordo com o Capítulo 3.

Dado o atributo A_x, pertencente ao esquema da loja L_x, e o atributo A_y, pertencente ao esquema da loja L_y; temos associados à eles os conjuntos V_x e V_y, respectivamente. Cada função computa a similaridade entre um par de valores v_i^x e v_j^y, pertencentes a V_x e V_y, respectivamente.

Destacamos que nos experimentos realizados nesse trabalho, foram aplicadas operações de limpeza e padronização para todos os valores dos atributos. Especificamente para os atributos numéricos e dimensionais foram realizadas as devidas conversões de valores para que todos permanecessem na mesma unidade de medida.

A seguir, descrevemos cada uma dessas funções.

4.2.1 Categóricos e Multicategóricos

Para os atributos Categóricos e Multicategóricos selecionamos duas funções de similaridade: Jaro-Winkler e Cosseno.

A função *Jaro-Winkler* [De Carvalho et al., 2013] foi utilizada apenas para atributos categóricos e calcula a distância de edição entre duas *strings*. Quanto menor a distância de edição entre as *strings*, maior a similaridade; sendo 0 equivalente a similaridade máxima e 1 equivalente a nenhuma similaridade. A similaridade é então medida subtraindo o resultado da distância entre as duas *strings* por 1.

Em nosso caso, os valores dos atributos Categórico e Multicategórico são considerados *strings*, mesmo que sejam números.

Antes de apresentar a função Jaro-Winkler, apresentamos os conceitos por ela utilizados. Dois caracteres são considerados *correspondentes* se eles forem iguais

e estiverem na mesma posição ou no máximo na posição resultante do cálculo $\left\lfloor \frac{max(|v_i^x|,|v_j^y|)}{2} \right\rfloor - 1$, onde $|v_i^x|$ ($|v_j^y|$) é o comprimento da *string* $v_i^x(v_j^y)$.

Levando em consideração que as *strings* podem ter caracteres *correspondentes*, mas em ordens diferentes, o número de *transposições* corresponde a quantidade de vezes que esses caracteres terão que ser trocados de lugar para que as *strings* se tornem iguais.

Dado dois valores v_i^x e v_j^y, com $v_i^x \in V_x$ e $v_j^y \in V_y$, a função é apresentada pela fórmula:

$$Jaro - Winkler^C(v_i^x, v_j^y) = 1 - \left[\frac{1}{3} \times \left(\frac{m}{|v_i^x|} + \frac{m}{|v_j^y|} + \frac{m-t}{m}\right)\right]$$

Onde m é o número de caracteres *correspondentes* entre v_i^x e v_j^y; e t é metade do número de *transposições* entre as duas *strings*.

Exemplificando, para os valores $v_i^x = toshiba$ e $v_j^y = semp\ toshiba$ temos: $m = 7$, $|v_i^x| = 7$, $|v_j^y| = 12$ e $t = 7$.

O cálculo da função ficaria:

$$Jaro{-}Winkler^C(toshiba, semp\ toshiba) = 1{-}\left[\frac{1}{3} \times \left(\frac{7}{7} + \frac{7}{12} + \frac{7-7}{7}\right)\right] = 1{-}0.53 = 0.47$$

A segunda função de similaridade utilizada, *Cosseno* [De Carvalho et al., 2013], pode ser aplicada para atributos Categóricos e Multicategóricos, uma vez que contabiliza a frequência de termos em todos os conjuntos. Utilizamos a seguinte fórmula:

$$Cosseno^{C,M}(A_x, A_y) = \frac{\sum_{t \in T} w\,(t, V_x) \times w\,(t, V_y)}{\sqrt{\sum_{t \in T} w\,(t, V_x)^2} \times \sqrt{\sum_{t \in T} w\,(t, V_y)^2}}$$

Considerando T_x como um conjunto com todos os termos que ocorrem em V_x e T_y um conjunto com todos os termos que ocorrem em V_y, T é o conjunto com todos os termos presentes nos dois conjuntos, sendo $T = T_x \cup T_y$.

O peso w é definido pela fórmula:

$$w\,(t, V_x) = tf\,(t, V_x) \times idf\,(t)$$

tf é a frequência normalizada do termo t em um dos atributo e é dada pela fórmula:

$$tf(t, v_i^x) = \frac{freq(t, V_x)}{Max(freq(t, V_x))}$$

Sendo $freq(t, V_x)$ a frequência do termo no conjunto V_x e $Max(freq(t, V_x))$ é a frequência do termo com maior frequência.

Neste trabalho, a função cosseno contabiliza o total de atributos nd como o total de atributos presente em um dos esquemas, $|L_x|$ ou $|L_y|$, mais um, que é o atributo analisado pertencente ao outro esquema. Sua fórmula é indicada por: $nd = (|L_x| \wedge |L_y|) + 1$. O número de atributos em que o termo aparece, $nt(t)$, então, é calculado contabilizando em quantos dos atributos de nd o termo está presente.

Considerando nd e $nt(t)$, o índice idf calcula em quantos atributos um termo pode ser encontrado. Ele é calculado pela fórmula:

$$idf\,(t) = \log\left(1 + \frac{nd}{nt\,(t)}\right)$$

Por causa do cálculo do nd, o cosseno pode ser calculado em duas direções, $A_x \rightarrow A_y$ ou $A_x \leftarrow A_y$, gerando resultados distintos. Nessa função, a agregação será aplicada entre os resultados do cosseno em cada uma das direções.

Exemplificando o cálculo do cosseno, dado os conjuntos V_x, do atributo $A_x = marca$, e V_y, do atributo $A_y = marca$. Calcularemos a função para $Cosseno^C(A_x, A_y)$ e para $Cosseno^C(A_y, A_x)$.

Inicialmente, vamos calcular o $Cosseno^C(A_x, A_y)$. Nessa direção, $nd = 15$, pois $|L_x| = 14$ e $nd = |L_x| + 1$. Os termos e as respectivas frequências de V_x e V_y são apresentados na Tabela 4.1.

Ax = marca		
Termo	**freq(t,Vx)**	**nt(t)**
canon	4	3
fujifilm	1	2
importado	1	1
mitsuca	1	2
nikon	9	4
patriot	1	1
polaroid	3	2
produto	1	1
sandisk	1	2
sony	**27**	3

Ay = marca		
Termo	**freq(t,Vy)**	**nt(t)**
canon	5	3
case	1	1
fujifilm	1	2
nikon	9	4
polaroid	3	2
sony	**22**	3

Tabela 4.1. Exemplo - Frequência de Termos.

Com essas informações podemos calcular o tf e o idf dos termos, conforme observamos nas Tabelas 4.2 e 4.3.

Ax = marca			
Termo	**tf(t,Vx)**	**idf(t)**	**w(t,Vx)**
canon	$\frac{4}{27} = 0.15$	$\log\left(1+\frac{15}{3}\right) = 0.78$	$0.15 \times 0.78 = 0.12$
fujifilm	$\frac{1}{27} = 0.04$	$\log\left(1+\frac{15}{2}\right) = 0.93$	$0.04 \times 0.93 = 0.03$
importado	$\frac{1}{27} = 0.04$	$\log\left(1+\frac{15}{1}\right) = 1.2$	$0.04 \times 1.2 = 0.04$
mitsuca	$\frac{1}{27} = 0.04$	$\log\left(1+\frac{15}{2}\right) = 0.93$	$0.04 \times 0.93 = 0.03$
nikon	$\frac{9}{27} = 0.33$	$\log\left(1+\frac{15}{4}\right) = 0.68$	$0.33 \times 0.68 = 0.23$
patriot	$\frac{1}{27} = 0.04$	$\log\left(1+\frac{15}{1}\right) = 1.2$	$0.04 \times 1.2 = 0.04$
polaroid	$\frac{3}{27} = 0.11$	$\log\left(1+\frac{15}{2}\right) = 0.93$	$0.11 \times 0.93 = 0.1$
produto	$\frac{1}{27} = 0.04$	$\log\left(1+\frac{15}{1}\right) = 1.2$	$0.04 \times 1.2 = 0.04$
sandisk	$\frac{1}{27} = 0.04$	$\log\left(1+\frac{15}{2}\right) = 0.93$	$0.04 \times 0.93 = 0.03$
sony	$\frac{27}{27} = 1$	$\log\left(1+\frac{15}{3}\right) = 0.78$	$1 \times 0.78 = 0.78$

Tabela 4.2. Exemplo - Cálculo tf e idf para A_x

Ay = marca			
Termo	**tf(t,Vy)**	**idf(t)**	**w(t,Vy)**
canon	$\frac{5}{22} = 0.23$	$\log\left(1+\frac{15}{3}\right) = 0.78$	$0.23 \times 0.78 = 0.18$
case	$\frac{1}{22} = 0.05$	$\log\left(1+\frac{15}{3}\right) = 1.2$	$0.05 \times 1.2 = 0.05$
fujifilm	$\frac{1}{22} = 0.05$	$\log\left(1+\frac{15}{2}\right) = 0.93$	$0.05 \times 0.93 = 0.04$
nikon	$\frac{9}{22} = 0.41$	$\log\left(1+\frac{15}{4}\right) = 0.68$	$0.41 \times 0.68 = 0.28$
polaroid	$\frac{3}{22} = 0.14$	$\log\left(1+\frac{15}{2}\right) = 0.93$	$0.14 \times 0.93 = 0.13$
sony	$\frac{22}{22} = 1$	$\log\left(1+\frac{15}{3}\right) = 0.78$	$1 \times 0.78 = 0.78$

Tabela 4.3. Exemplo - Cálculo tf e idf para A_y

Com essas informações, podemos aplicar a função $Cosseno^C(A_x, A_y)$. Aqui serão descritos os cálculos separadamente para facilitar a visualização.

$\sum_{t \in T} w\left(t, V_x\right) \times w\left(t, V_y\right)$

$= \ (0.12 \times 0.28) + (0.03 \times 0.05) + (0.03 \times 0) + (0.23 \times 0) + (0.04 \times 0.04) + (0.10 \times 0) + (0.04 \times 0.13) + (0.03 \times 0) + (0.78 \times 0) + (0.04 \times 0.78)$

$= 0.020 + 0.002 + 0 + 0.010 + 0 + 0.013 + 0 + 0 + 0.606 + 0$

$= 0.65$

$\sqrt{\sum_{t \in T} w\left(t, V_x\right)^2}$

$= \sqrt{0.12^2 + 0.03^2 + 0.03^2 + 0.23^2 + 0.04^2 + 0.10^2 + 0.04^2 + 0.03^2 + 0.78^2 + 0.04^2}$

$= \sqrt{0.01 + 0 + 0 + 0.05 + 0 + 0.01 + 0 + 0 + 0.61 + 0}$

$= \sqrt{0.69} = 0.831$

$$\sqrt{\sum_{t \in T} w\left(t, V_y\right)^2}$$
$$= \sqrt{0.28^2 + 0.18^2 + 0.05^2 + 0.04^2 + 0.13^2 + 0.78^2}$$
$$= \sqrt{0.08 + 0.03 + 0 + 0 + 0.02 + 0.61}$$
$$= \sqrt{0.734}$$
$$= 0.857$$

$$Cosseno^C(A_x, A_y) = \frac{0.65}{0.831 \times 0.857} = \frac{0.65}{0.712} = 0.913$$

Em seguida, é feito o cálculo do $Cosseno^C(A_y, A_x)$. Neste caso, $|L_y| = 10$, então, $nd = 11$. Por causa do novo valor de nd, a frequência dos termos precisa ser recalculada. Os novos valores são apresentados nas tabelas 4.4, 4.5 e 4.6.

Ay = marca		
Termo	**freq(t,Vy)**	**nt(t)**
canon	5	3
case	1	1
fujifilm	1	3
nikon	9	3
polaroid	3	3
sony	**22**	3

Ax = marca		
Termo	**freq(t,Vx)**	**nt(t)**
canon	4	3
fujifilm	1	3
importado	1	1
mitsuca	1	3
nikon	9	1
patriot	1	3
polaroid	3	1
produto	1	2
sandisk	1	3
sony	**27**	1

Tabela 4.4. Exemplo 2 - Frequência de Termos.

Ay = marca			
Termo	**tf(t,Vy)**	**idf(t)**	**w(t,Vy)**
canon	0.23	0.67	0.15
case logic	0.05	1.08	0.05
fujifilm	0.05	0.67	0.03
nikon	0.41	0.67	0.27
polaroid	0.14	0.67	0.09
sony	1	0.67	0.67

Tabela 4.5. Exemplo 2 - Cálculo tf e idf para A_y

Realizando os mesmos calculos do $Cosseno^C(A_x, A_y)$, chegamos aos valores:

$$Cosseno^C(A_y, A_x) = \frac{0.48}{0.72 \times 0.75} = \frac{0.48}{0.54} = 0.89$$

Ax = marca			
Termo	**tf(t,Vx)**	**idf(t)**	**w(t,Vx)**
canon	0.15	0.67	0.10
fujifilm	0.04	0.67	0.02
importado	0.04	1.08	0.04
mitsuca	0.04	1.08	0.04
nikon	0.33	0.67	0.22
patriot	0.04	1.08	0.04
polaroid	0.11	0.67	0.07
produto	0.04	1.08	0.04
sandisk	0.04	0.81	0.03
sony	1	0.67	0.67

Tabela 4.6. Exemplo 2 - Cálculo tf e idf para A_x

4.2.2 Numéricos

Para atributos numéricos, selecionamos a função de similaridade apresentada por Hoffmann et al. [Hoffmann et al., 2015], que neste trabalho chamaremos de *Numerical*. Essa função é definida como a diferença absoluta entre dois valores numéricos e segue a seguinte fórmula:

$$Sim^N(v_i^x, v_j^y) = 1 - \frac{\left|v_i^x - v_j^y\right|}{max\left(v_i^x, v_j^y\right)}$$

Exemplificando, para o $v_i^x = 12\ gb$ e $v_j^y = 32\ gb$, o cálculo da função será:

$$Sim^N(12, 32) = 1 - \frac{|12 - 32|}{max\,(32)} = 1 - \frac{|-20|}{max\,(32)} = 1 - 0.625 = 0.375$$

4.2.3 Dimensionais

Para executar as funções de similaridade nos atributos dimensionais, primeiro aplicamos a normalização dos valores de acordo com a fórmula:

$$\left(v_x^d\right)' = \frac{\left(v_x^d - \mu_x^d\right)}{\sigma_x^d}$$

onde μ_x^d é a média de todos os valores de V_x na dimensão d e σ_x^d é o desvio padrão desse mesmo conjunto de valores para a dimensão d.

Por exemplo, para o conjunto:

$$V_x = \{5.8 \times 5.86 \times 20,\ 5.8 \times 5.86 \times 20,\ 7.5 \times 0.3 \times 12.5,\ 13 \times 8 \times 1,\ 15 \times 9 \times 17\}$$

os valores das médias e dos desvios padrões de cada dimensão são: $\mu_x^1 = 9.42$, $\mu_x^2 = 5.804$, $\mu_x^3 = 14.1$, $\sigma_x^1 = 4.297$, $\sigma_x^2 = 3.367$ e $\sigma_x^3 = 7.94$.

Então, para normalizar um dos elementos de V_x, nesse exemplo $v_x = 15 \times 9 \times 17$, o cálculo será, para cada dimensão:

$$\left(v_x^1\right)' = \frac{(15 - 9.42)}{4.297} = \frac{(5.58)}{4.297} = 1.299$$

$$\left(v_x^2\right)' = \frac{(9 - 5.804)}{3.367} = \frac{(3.196)}{3.367} = 0.949$$

$$\left(v_x^3\right)' = \frac{(17 - 14.1)}{7.94} = \frac{(2.9)}{7.94} = 0.365$$

Então, o valor normalizado de $v_i^x = 15 \times 9 \times 17$ será $(v_i^x)' = 1.299 \times 0.949 \times 0.365$.

As funções de similaridades escolhidas para os atributos dimensionais foram: Distância *Manhattan* [Liu, 2006], Distância *Euclideana* [Liu, 2006] e Distância *Camberra* [Liu, 2006].

A distância *Manhattan* é dada pela fórmula:

$$Manhattan^D((v_i^x)', (v_j^y)') = 1 - \sum_{k=1}^{d} \left|(v_x^k)' - (v_y^k)'\right|$$

Exemplificando, dado os valores $v_i^x = 15 \times 9 \times 17\,cm$ e $v_j^y = 20 \times 22 \times 7,8\,cm$, sendo $(v_i^x)' = 1.299 \times 0.949 \times 0.365$ e $\left(v_j^y\right)' = 1.091 \times 1.535 \times -0.497$ seus valores normalizados, o cálculo da distância para os valores será:

$$\begin{aligned}
&Manhattan^D((v_i^x)', \left(v_j^y\right)') \\
&= 1 - (|1.299 - 1.091| + |0.949 - 1.535| + |0.365 - (-0.497)|) \\
&= 1 - (|0.208| + |-0.586| + |0.862|) \\
&= 1 - 1.656 \\
&= 0.656
\end{aligned}$$

A distância *Euclideana* é dada pela fórmula:

$$Euclideana^D(v_i^x, v_j^y) = 1 - \sqrt{\sum_{k=1}^{d}((v_x^k)' - (v_y^k)')^2}$$

Utilizando os mesmos valores do exemplo anterior, o cálculo dessa função ficará:

$$\begin{aligned}
&Euclideana^D((v_i^x)', \left(v_j^y\right)') \\
&= 1 - \sqrt{(1.299 - 1.091)^2 + (0.949 - 1.535)^2 + (0.365 - (-0.497))^2} \\
&= 1 - \sqrt{(0.208)^2 + (-0.586)^2 + (0.862)^2} \\
&= 1 - \sqrt{0.041 + 0.343 + 0.743} \\
&= 1 - \sqrt{1.127} \\
&= 1 - 1.062 \\
&= 0.062
\end{aligned}$$

Por fim, a distância *Camberra* é dada pela fórmula:

$$Camberra^D(v_i^x, v_j^y) = 1 - \sum_{k=1}^{d} \frac{\left|(v_x^k)' - (v_y^k)'\right|}{|(v_x^k)'| + |(v_y^k)'|}$$

Também utilizando os valores do exemplo anterior, o cálculo da distância será:

$$\begin{aligned}
&Camberra^D((v_i^x)', \left(v_j^y\right)') \\
&= 1 - \left(\tfrac{|1.299-1.091|}{|1.299|+|1.091|} + \tfrac{|0.949-1.535|}{|0.949|+|1.535|} + \tfrac{|0.365-(-0.497)|}{|0.365|+|-0.497|}\right) \\
&= 1 - \left(\tfrac{|0.208|}{2.39} + \tfrac{|-0.586|}{2.484} + \tfrac{|0.862|}{0.862}\right) \\
&= 1 - (0,087 + 0.236 + 1) \\
&= 1 - 1.323 \\
&= 0.323
\end{aligned}$$

4.2.4 Booleanos

Para os atributos booleanos, decidimos não aplicar *matchers* baseados em instância. Nas instâncias das bases de dados utilizadas, esses atributos apresentam apenas os valores "sim" ou "não", o que não contribui nos cálculos dos *matchers* de instância. Podem ocorrer casos como, por exemplo, de um atributo chamado "grava data/hora" com valor "sim" na loja Ponto Frio e "microfone embutido (cameras)" também com valor "sim" na loja Fastshop. Ambos possuem o mesmo valor, então, um *matcher* baseado em instância vai retornar um valor alto para a similaridade desses dois atributos, apesar

deles não serem correspondentes. Para evitar isso, decidimos não aplicar *matchers* baseados em instância nesses atributos.

Utilizamos a classificação dessa classe de atributo para evitar a aplicação dos *matchers* de instância entre atributos booleanos com outras classes de atributos. Assim, identificamos antecipadamente que não há correspondência entre eles e, portanto, atribuimos similaridade zero.

4.3 Métodos de Agregação

Para poder gerar a similaridade do par de atributos $Sim(A_x, A_y)$, calculados através das instâncias V_x e V_y, é necessário utilizar um método de agregação, responsável por recuperar todos os resultados par a par dos valores das instâncias e agregá-los em um único resultado. Para realizar essa tarefa, utilizamos o método *Average Link* [Liu, 2006], selecionado através do um estudo que será descrito na Seção 4.4.

No Algoritmo 1 descrevemos como a nossa abordagem aplica a agregação. O algoritmo, como podemos observar nas Linhas 6 e 7, percorre cada item de cada conjunto para fazer a combinação dos seus valores. Na Linha 8, a função de similaridade é aplicada nesses pares e seu resultado é armazenado em incrementado à variável *simValores*. Para calcular a similaridade dos atributos, aplicamos o método de agregação (Linha 13), claculando a média dos resultado da função nos pares de valores. Para isso, na Linha 11 dividimos a variável *simValores* pelo total de pares de valores.

Algoritmo 1: Agregação utilizada nas funções de similaridade.

Entrada: A_x, A_y, V_x, V_y
Saída: Similaridade entre os atributos A_x, A_y

```
1  início
2  |  simValores ← 0
3  |  Seja V_x = {v_1^x, ..., v_n^x};
4  |  Seja V_y = {v_1^y, ..., v_m^y};
5  |  para cada v_i^x ∈ V_x faça
6  |  |  para cada v_j^y ∈ V_y faça
7  |  |  |  simValores ← simValores + funcSim(v_i^x, v_j^y)
8  |  |  fim
9  |  fim
10 |  Sim = simValores/(n × m)
11 |  retorna Sim
12 fim
```

A função Cosseno não utiliza esse algoritmo, uma vez que não compara os ele-

mentos par a par. Entretanto, como explicamos na Seção 4.2, o Cosseno é calculado em duas direções: $A_x \leftarrow A_y$ e $A_x \rightarrow A_y$. Esses cálculos obtêm resultados distintos, que precisam ser agregados para a similaridade receber um único valor. No Cosseno aplicamos duas funções de agregação e cada uma delas resultou em um valor de similaridade. Dessa forma, temos duas funções Cosseno: CossenoAVG, com o resultado aplicando *Average Link*; e CossenoSingleLink, com o resultado aplicando a agregação Single Link [Liu, 2006].

A agregação *Single Link* calcula a distância entre os dois pontos mais próximos em dois conjuntos. Para tanto, ele verifica quais pares de elementos possuem a menor distância, ou seja, maior similaridade. O par com maior similaridade será, então, escolhido.

Como podemos observar no Algoritmo 2, primeiro aplicamos o Cosseno nas duas direções das instâncias (Linhas 2 e 3). Então, aplicamos o método de agregação nesses resultados (Linha 4), e retornamos como a similaridade dos atributos.

Algoritmo 2: Agregação utilizada na função Cosseno.

Entrada: A_x, A_y, V_x, V_y
Saída: Similaridade entre os atributos A_x, A_y

```
1 início
2 |  funcSimXY ← Cosseno(V_x, V_y)
3 |  funcSimYX ← Cosseno(V_y, V_x)
4 |  Sim ← Agregacao(funcSimXY, funcSimYX)
5 |  retorna Sim
6 fim
```

Considerando o exemplo apresentado na Seção 4.2 para a função Cosseno, para os valores de similaridade $Cosseno^C(A_x, A_y) = 0.913$ e $Cosseno^C(A_y, A_x) = 0.89$, o cálculo será:

$$CossenoAVG = \frac{Cosseno^C(A_x, A_y) + Cosseno^C(A_y, A_x)}{2} = \frac{0.913 + 0.89}{2} = 0.902$$

$$CossenoSingleLink = max(Cosseno^C(A_x, A_y), Cosseno^C(A_y, A_x))$$

$$= max(0.913, 0.89) = 0.913$$

4.4 Seleção das Funções de Similaridade

Para selecionarmos as funções de similaridade, realizamos uma revisão da literatura para identificar quais as funções de similaridade poderiam ser aplicadas para as instâncias dos atributos. Na revisão, procuramos funções que foram utilizadas em métodos de casamento de esquema e funções que poderiam ser aplicadas em conjuntos.

Após a identificação das funções de similaridade, realizamos experimentos de validação para verificar quais funções de similaridade seriam utilizadas neste trabalho.

Nessa seção, descrevemos o processo de seleção da funções. Para tanto, detalharemos a revisão da literatura e explicaremos os experimentos de validação.

4.4.1 Revisão da Literatura

A revisão da literatura foi realizada analisando as principais referências sobre casamento de esquemas e os artigos citados por elas. Adicionalmente, pesquisamos trabalhos de casamento de esquemas que utilizam informação de instância nas máquinas de busca *Google Acadêmico*[1] e *Semantics Scholar*[2].

Selecionamos 132 artigos e 2 livros-texto para análise e, dentre esses trabalhos, encontramos 27 que utilizavam informação de instância. Após a análise, selecionamos 13 funções de similaridade para validarmos na nossa abordagem. A lista de funções de similaridade e os trabalhos que fazem referência a elas encontram-se na Tabela 4.7.

Funções de Similaridade	**Referências**
Camberra	[Liu, 2006]
Categorical (Similaridade para atributos Categóricos)	[Hoffmann et al., 2015] [Kagie et al., 2008]
Cosseno	[De Carvalho et al., 2013] [Mesquita et al., 2007a]
Euclideana	[Liu, 2006] [Hoffmann et al., 2015] [Kagie et al., 2008] [Kang & Naughton, 2003]
Fledex Content Similarity	[Mesquita et al., 2007a]
Fledex Value-based Similarity	[Mesquita et al., 2007a]
Jaccard	[Nguyen et al., 2011] [Hoffmann et al., 2015] [Kagie et al., 2008]
Jaro-Winkler	[De Carvalho et al., 2013]
Jensen Shannon	[Nguyen et al., 2011]
KLD	[Nguyen et al., 2011]
Manhattan	[Liu, 2006]
Numerical (Similaridade para atributos numéricos)	[Hoffmann et al., 2015] [Kagie et al., 2008]
TFIAF	[Mesquita et al., 2007b]

Tabela 4.7. Funções de Similaridade Avaliadas

[1]http://scholar.google.com.br
[2]https://www.semanticscholar.org

Algumas destas funções comparam os valores em pares e necessitam agregar os seus resultados para gerar um valor de similaridade para o atributo. Para realizar essa tarefa, selecionamos alguns métodos de agregação para *clusters* conhecidos para conjuntos. Na Tabela 4.8 apresentamos os métodos selecionados. Chamamos de AVG_E o método de agregação utilizado em uma das funções apresentada por De Carvalho et.al. [De Carvalho et al., 2013]. Essa função consiste em selecionar apenas as similaridades com resultado acima de um limiar e gerar média desses valores.

Métodos de Agregação	**Fontes**
Single Link	[Liu, 2006]
Complete Link	[Liu, 2006]
Average Link	[Liu, 2006]
Valor mais frequente	[Liu, 2006]
AVG_E	[De Carvalho et al., 2013]

Tabela 4.8. Métodos de Agregação Avaliados

4.4.2 Validação das Funções de Similaridade

Após a escolha das funções, realizamos a sua implementação e validação para verificar o comportamento de cada uma na nossa abordagem.

Para validarmos as funções de similaridade, aplicamos cada função individualmente na base dados *BDRI* e verificamos a quantidade de correspondências verdadeiras e falsas retornados, comparando com a quantidade de correspondências verdadeiras e falsas total para categoria.

Para comparar os resultados, contabilizamos a quantidade de correspondências verdadeiras e falsas retornadas pelas funções por par de loja e calculamos a média por categoria. Essa média foi comparada com a média de valores verdadeiros e falsos total. Nas figuras, as correspondências verdadeiras e falsas retornadas pelas funções são representadas como V e F, respectivamente.

Os resultados da validação das funções para as classes Categórico e Multicategórico são apresentados nas Figuras 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7 e 4.8.

Com exceção do KLD e do Jensen Shannon, as funções retornaram grande parte das correspondências falsas. Entretanto, a maioria das funções retornaram poucas correspondências verdadeiras. As exceções foram as funções a Jaro-Winkler, para categóricos, e a Cosseno, para categóricos e multicategóricos. Decidimos utilizar essas duas funções nessas classes de atributos.

Nas Figuras 4.9, 4.10, 4.11, 4.12 e 4.13 apresentamos os resultados da validação das funções para as classes Numéricos e Dimensionais. Para os numéricos, apenas a

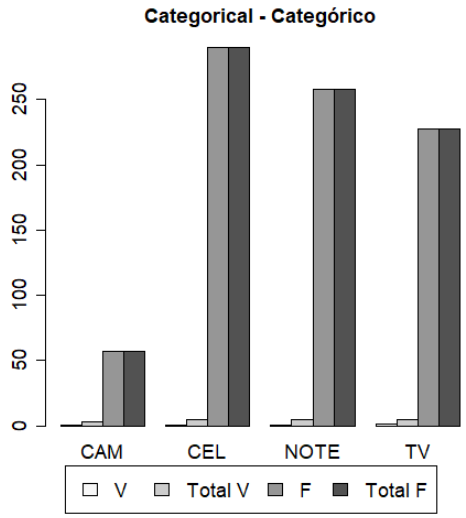

Figura 4.1. Experimentos com a Função de Similaridade Categorical.

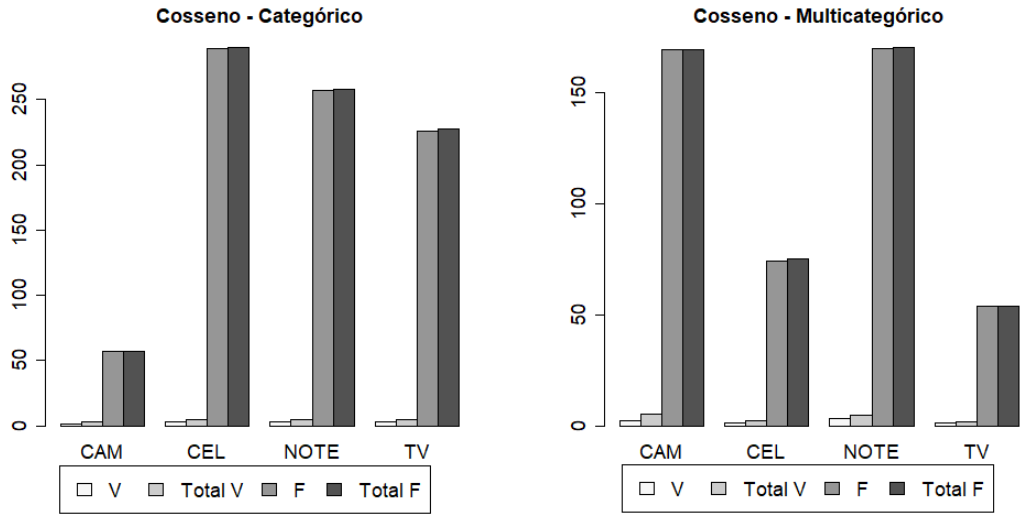

Figura 4.2. Experimentos com a Função de Similaridade Cosseno.

função *Numerical* obteve bons resultados para identificar os atributos correspondentes verdadeiros. Para os dimensionais, os melhores resultados foram obtidos pelas funções Camberra, Euclideana e Manhattan. Destacamos que os atributos dimensionais aparecem em menor quantidade nas bases de dados, sendo, muitas vezes, um único atributo por loja. Por isso, nessas funções os resultados das correspondências falsas foram baixas.

Considerando que algumas dessas funções precisam utilizar um método de agregação para gerar a similaridade, avaliamos qual método de agregação seria considerado na abordagem. Para tanto, selecionamos alguns pares de atributos da base de dados

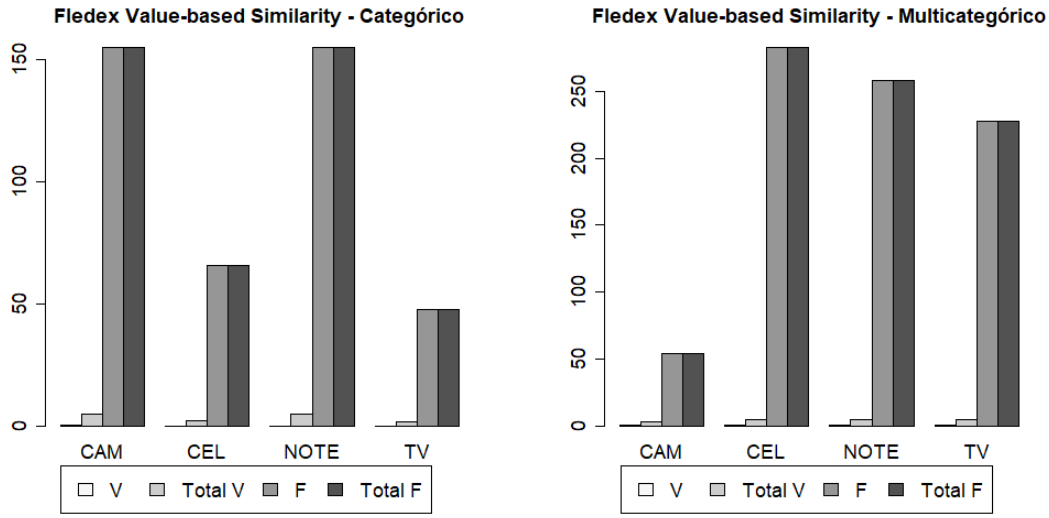

Figura 4.3. Experimentos com a Função de Similaridade Fledex Value-Based.

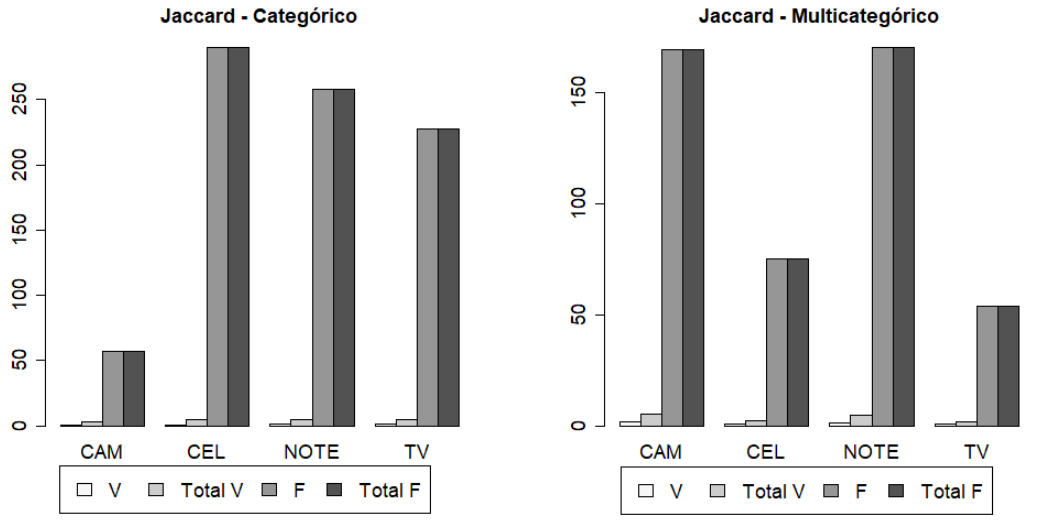

Figura 4.4. Experimentos com a Função de Similaridade Jaccard.

BDRI e verificamos o resultado retornado pelas funções para cada um dos métodos de agregação. Os pares de atributos utilizados nessa avaliação são apresentados na Tabela 4.9.

Nas Tabelas 4.10, 4.11, 4.12, 4.13, 4.14 apresentamos os resultados das funções para cada método de agregação. Podemos verificar que os métodos de agregação *Single Link*, *Complete Link* e *AVG_E* tendem a gerar valores muito altos para as funções, o que pode causar muitos falsos positivos e falsos negativos. Entretanto, o *Average Link* obteve resultados mais assertivos, tanto para as correspondências verdadeiras quanto para as falsas. Optamos por não utilizar o *Valor mais Frequente* uma vez que o valor

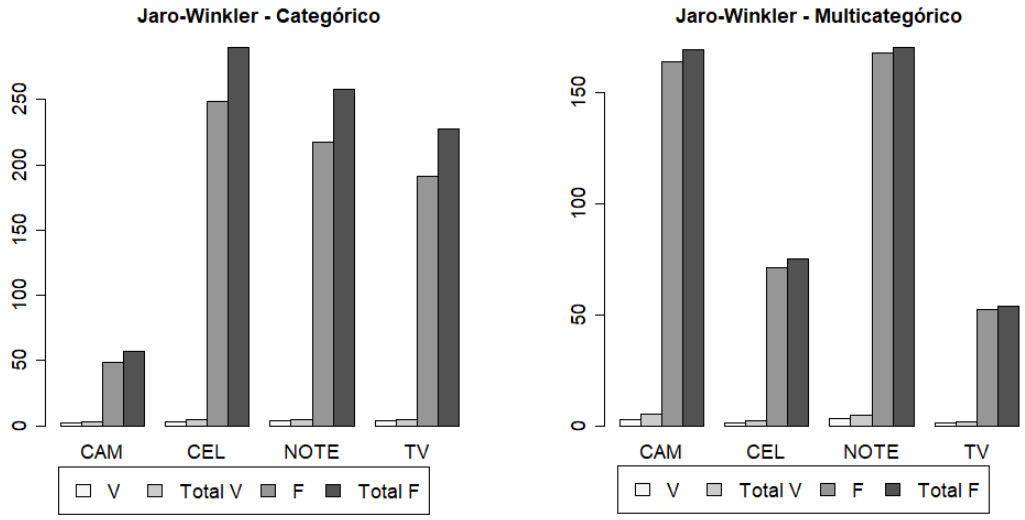

Figura 4.5. Experimentos com a Função de Similaridade Jaro-Winkler.

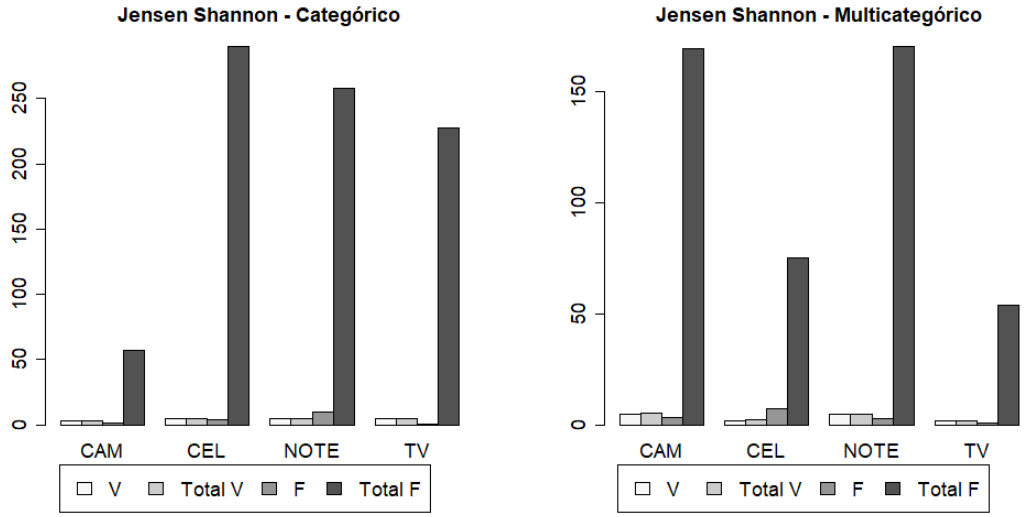

Figura 4.6. Experimentos com a Função de Similaridade Jensen Shannon.

Referência	Classe de Atributo	Ax	Ay	Correspondência
ex1	Categórico	cor do visor	cor do visor	Verdadeira
ex2	Categórico	marca	cor do visor	Falsa
ex3	Multicategórico	idiomas do menu	idiomas do menu	Verdadeira
ex4	Numérico	tamanho da tela	resolucao	Falsa
ex5	Numérico	peso aproximado da embalagem c produto (kg)	peso aproximado da embalagem c produto (kg)	Verdadeira
ex6	Dimensional	dimensões aproximadas da embalagem (cm) - axlxp	dimensões aproximadas da embalagem (cm) - axlxp	Verdadeira

Tabela 4.9. Amostras Utilizadas para Validar os Métodos de Agregação.

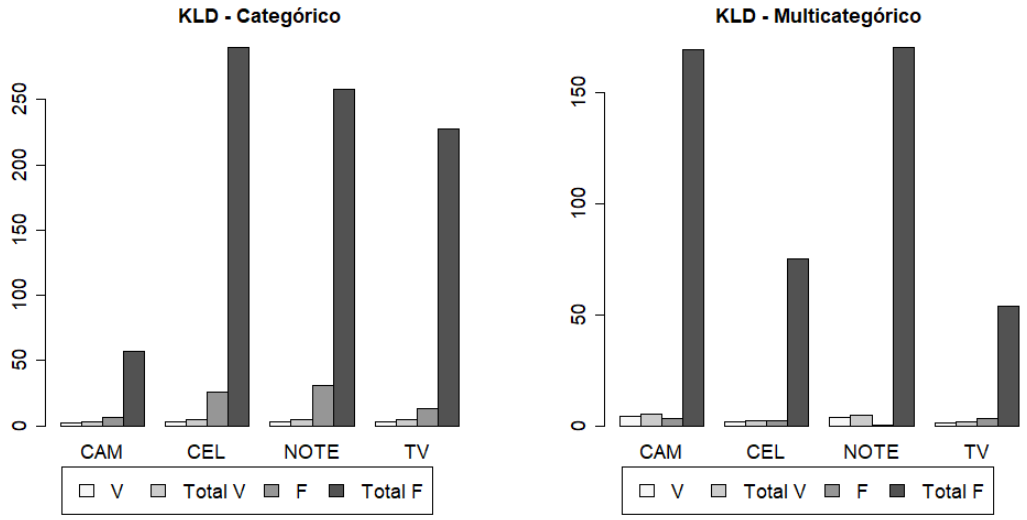

Figura 4.7. Experimentos com a Função de Similaridade KLD.

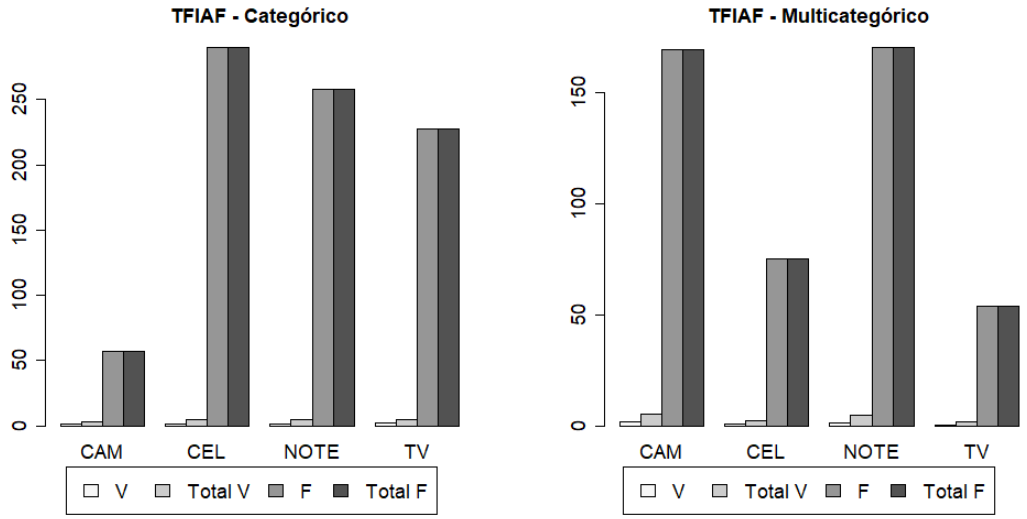

Figura 4.8. Experimentos com a Função de Similaridade TFIAF.

mais frequente de dois atributos podem ser muito diferentes, apesar dos atributos serem correspondentes. Isso pode ser observado na Tabela 4.13 no ex6 da função Manhattan.

Dessa forma, decidimos utilizar o *Average Link* como método de agregação. Para a função Cosseno, além do *Average Link* decidimos utilizar o método *Single Link*, uma vez que o cálculo do Cosseno ocorre conforme explicado na Seção 4.3.

Nas Tabelas 4.15 e 4.16 apresentamos o resultado final da avaliação das funções de similaridade e dos métodos de agregação.

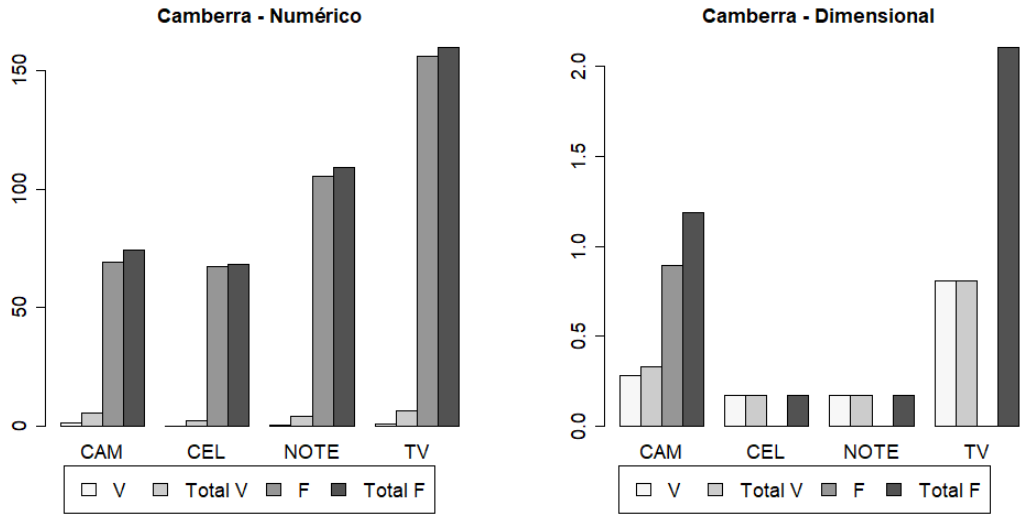

Figura 4.9. Experimentos com a Função de Similaridade Camberra.

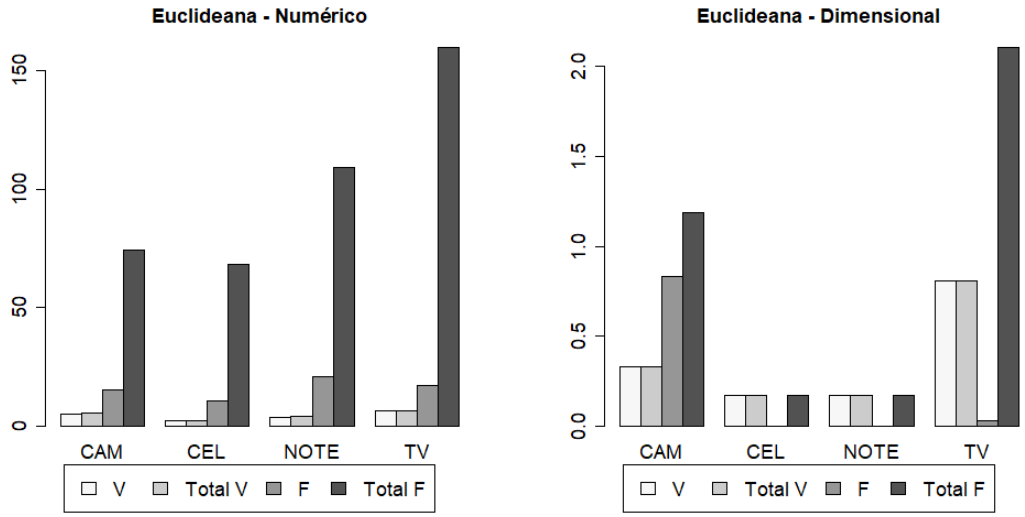

Figura 4.10. Experimentos com a Distância Euclideana.

Single Link						
	ex1	**ex2**	**ex3**	**ex4**	**ex5**	**ex6**
Jaro-Winkler	1	0.778	-	-	-	-
Cosseno	0.984	0	0.991	-	-	-
Numerical	-	-	-	0.85	0.016	-
Camberra	-	-	-	1	0	1
Euclideana	-	-	-	1	1	1
Manhattan	-	-	-	1	1	1

Tabela 4.10. Resultados da Validação do Single Link.

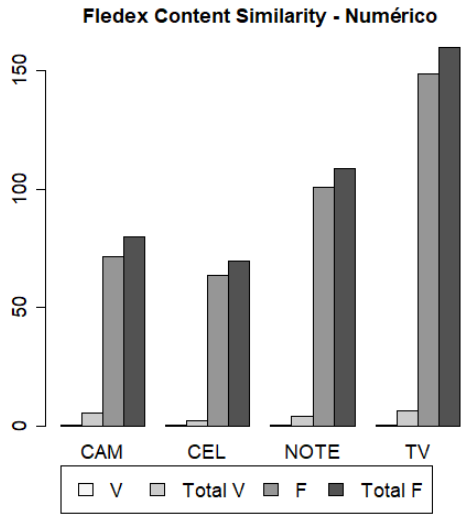

Figura 4.11. Experimentos com a Função de Similaridade Fledex Content.

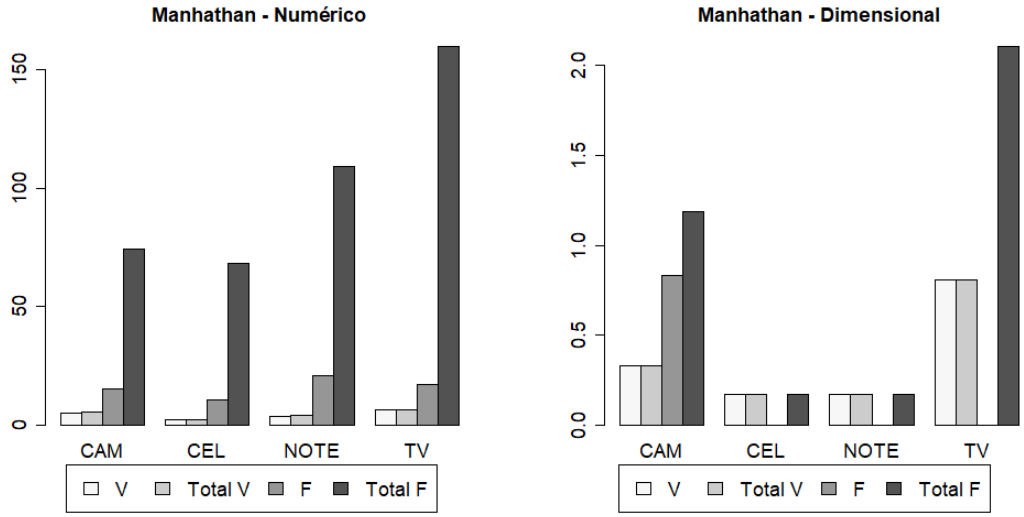

Figura 4.12. Experimentos com a Função de Similaridade Manhattan.

Complete Link						
	ex1	**ex2**	**ex3**	**ex4**	**ex5**	**ex6**
Jaro-Winkler	1	0.778	-	-	-	-
Cosseno	1	0	0	-	-	-
Numerical	-	-	-	0.85	0.016	-
Camberra	-	-	-	1	0	1
Euclideana	-	-	-	1	1	1
Manhattan	-	-	-	1	1	1

Tabela 4.11. Resultados da Validação do Complete Link.

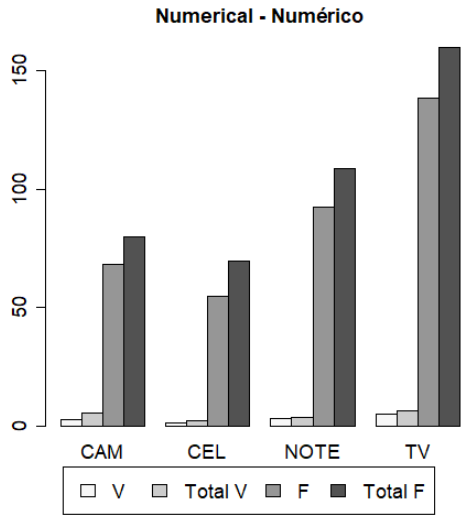

Figura 4.13. Experimentos com a Função de Similaridade Numerical.

Average Link						
	ex1	**ex2**	**ex3**	**ex4**	**ex5**	**ex6**
Jaro-Winkler	0.678	0.292	-	-	-	-
Cosseno	0.982	0	0.991	-	-	-
Numerical	-	-	-	0.547	0.016	-
Camberra	-	-	-	1	0	1
Euclideana	-	-	-	1	0.512	1
Manhattan	-	-	-	1	0.64	1

Tabela 4.12. Resultados da Validação do Avarage Link.

Valor mais Frequente						
	ex1	**ex2**	**ex3**	**ex4**	**ex5**	**ex6**
Jaro-Winkler	1	0	-	-	-	-
Cosseno	-	0	-	-	-	-
Numerical	-	-	-	0.97	0.016	-
Camberra	-	-	-	1	0	1
Euclideana	-	-	-	1	0.506	1
Manhattan	-	-	-	1	0.506	0.046

Tabela 4.13. Resultados da Validação do Valor mais Frequente.

4.5 Considerações Finais

Nesse capítulo apresentamos a nossa abordagem para utilizar a informação de instâncias em métodos de casamento de esquemas no domínio de comércio eletrônico. Decidimos considerar que cada atributo se enquadra em apenas uma classe de atri-

AVG_E						
	ex1	**ex2**	**ex3**	**ex4**	**ex5**	**ex6**
Jaro-Winkler	0.775	0.591	-	-	-	-
Cosseno	0	0	0	-	-	-
Numerical	-	-	-	0.724	0	-
Camberra	-	-	-	1	0.751	1
Euclideana	-	-	-	1	0.776	1
Manhattan	-	-	-	1	0.776	1

Tabela 4.14. Resultados da Validação do AVG_E.

Matcher	Categórico	Multicategórico	Numérico	Dimensional
Camberra	-	-	x	OK
Categorical	x	-	-	-
Cosseno	OK	OK	-	-
Euclideana	-	-	x	OK
Fledex Content Similarity	-	-	x	-
Fledex Value-based Similarity	x	x	-	-
Jaccard	x	x	-	-
Jaro-Winkler	OK	x	-	-
Jensen Shannon	x	x	-	-
KLD	x	x	-	-
Manhattan	-	-	x	OK
Numerical	-	-	OK	-
TFIAF	x	x	-	-

Tabela 4.15. Resultado da Validação das Funções de Similaridade

Métodos de Agregação	Situação
Single Link	x
Complete Link	x
Average Link	OK
Valor mais frequente	x
AVG_E	x

Tabela 4.16. Resultado da Validação das Funções de Similaridade

butos (Capítulo 3), sendo elas Categóricos, Multicategóricos, Booleanos, Numéricos e Dimensionais. Também apresentamos um estudo sobre quais funções de similaridade se enquadrariam melhor para cada classe de atributos, bem como quais métodos de agregação deveriam ser utilizados em cada função. De forma que a informação de instância pudesse contribuir da melhor maneira possível para os métodos de casamento de esquemas. Como resultado desse estudo, selecionamos como função de similaridade dos atributos categóricos as funções Cosseno e Jaro-winkler, para os multicategóricos a função Cosseno, para os atributos numéricos a função Numerical e para os atributos dimensionais as funções Camberra, Euclidiana e Manhathan. Também escolhemos como método de agregação o Average Link, que calcula a média dos resultados das funções

em cada instância. Adicionalmente, para a função Cosseno também utilizaremos o método Single Link, uma vez que calculamos cosseno duas vezes para cada instância, conforme explicado na Seção 4.3.

Capítulo 5

Resultados Experimentais

Esse capítulo apresenta uma avaliação experimental sobre o uso de informação de instância para o casamento de esquemas no domínio de comércio eletrônico. O objetivo principal desse capítulo é verificar experimentalmente se matchers baseados em instância influenciam de forma positiva os resultados retornados pelos métodos de casamento de esquema.

Para tanto, os experimentos foram executados com três métodos: COMA [Do & Rahm, 2002], ALMa [Rodrigues et al., 2015] e RFSM [Rodrigues, 2017]. Procurou-se analisar o comportamento de cada um destes métodos nas bases de dados de comércio eletrônico descritas no Capítulo 3, bem como se os resultados dos experimentos utilizando *matchers* baseados em instância se sobressaíram aos resultados utilizando apenas *matchers* baseados em esquema.

Nas seções seguintes, são apresentadas as métricas de avaliação utilizadas para comparar os experimentos, as configurações utilizadas para cada método, os resultados dos experimentos, uma análise sobre como os *matchers* foram utilizados nos métodos, um estudo sobre as configurações dos métodos e, por fim, as considerações finais sobre os resultados obtidos.

5.1 Métricas de Avaliação

Para avaliar os resultados dos experimentos, foram escolhidas as mesmas métricas de avaliação utilizadas originalmente para avaliar o COMA e o ALMa: precisão, revocação e medida-F. Nos resultados, foram considerados apenas os pares positivos, em que o par equivale a uma correspondência correta, uma vez que encontrar os casos positivos é o objetivo real do problema de casamento de esquemas. Além disso, a quantidade de pares verdadeiros e falsos é desbalanceada, o que prejudica a avaliação.

Na avaliação, foram considerados os resultados das médias de cada métrica entre todas as tarefas para cada categoria. Considerando C como o conjunto de pares verdadeiros de uma tarefa e R como o conjunto de pares retornados como verdadeiros pelo método, as fórmulas utilizadas foram:

- Precisão: Calcula o percentual de pares retornados pelo método que equivalem aos pares verdadeiros.
$$precisão = \frac{|C \cap R|}{|R|}$$
- Revocação: Verifica dentre todos os pares verdadeiros, quantos foram retornados pelo método.
$$revocacão = \frac{|C \cap R|}{|C|}$$
- Medida-F: Média ponderada da precisão e da revocação. Verifica a concordância entre as duas métricas.
$$medida - F = \frac{2 \times precisão \times revocacão}{precisão + revocacão}$$

5.2 Configuração dos Experimentos

5.2.1 Métodos Utilizados nos Experimentos

Os experimentos foram realizados usando três métodos distintos, COMA [Do & Rahm, 2002], que utiliza uma abordagem heurística; ALMa [Rodrigues et al., 2015], que utiliza aprendizado ativo, e RFSM [Rodrigues, 2017], que utiliza aprendizado de máquina supervisionado. As implementações dos métodos COMA, ALMa e RFSM utilizados foram baseadas em implementações usadas em outros trabalhos do nosso grupo de pesquisa [Rodrigues et al., 2015] e nos foram gentilmente fornecidas.

5.2.2 Matchers Utilizados

Os experimentos foram realizados utilizando duas configurações distintas de *matchers*: utilizando apenas *matchers* baseados em esquema e utilizando *matchers* baseados em esquema com a adição de *matchers* baseados em instância. Para os *matchers* baseados em esquema, foi utilizada a mesma configuração de *matchers* apresentada por Rodrigues et al. [Rodrigues et al., 2015], constituída pelos *matchers* disponibilizados pela

biblioteca do COMA com a adição de outras medidas de similaridade disponibilizadas pelo *Second String Project*[1].

Como o RFSM tem como princípio selecionar instâncias aleatoriamente para gerar as árvores de decisão, não seria possível garantir que os *matchers* de instância serão selecionados. Isso impediria que fosse avaliado se os *matchers* de instância trariam um efeito positivo no método. Então, para realizar a avaliação, foi necessário executar os experimentos 300 vezes e recuperar os 30 resultados em que o método gerou a maior quantidade de árvores de decisão com pelo menos um *matcher* de instância.

5.2.3 Configurações Utilizadas nos Experimentos

A seguir são apresentadas as configurações individuais de cada método utilizadas nos experimentos. Essas configurações foram definidas considerando as configurações reportadas como as que obtiveram melhores resultados em seus artigos de origem e considerando os resultados de um estudo detalhado dos parâmetros, que apresentamos posteriormente na Seção 5.5.

COMA: Foram utilizados *AverageLink* como método de agregação, MaxDelta como método de seleção com valor $\delta = 0{,}02$ e *Threshold* também como método de seleção com valor $th = 0.5$, para base *BDRI* e $th = 0{,}6$ para a base *Dexter*.

ALMa: O método de aprendizagem ativo utiliza um conjunto de 30 árvores de decisão, onde cada árvore é treinada com um subconjunto aleatório e diferente de *matchers*. O primeiro conjunto de treino usado pelo método é composto de pares selecionados aleatoriamente de cinco *clusters* obtidos por um algoritmo K-Means. A cada rodada, o comitê de decisões escolhe pares para serem rotulados pelo usuário. Para a *BDRI* foram escolhidos 25 pares por rodada e na *Dexter* 20 pares.Para rotular um par como verdadeiro o comitê precisa ter menos de 0.75 de confiança. Para garantir a validade do experimento, cada tarefa de casamento foi executada pelo menos 30 vezes, mudando a semente aleatória em todas elas.

RFSM: Foram utilizadas as configurações padrão para *Random Forest* indicadas em [Breiman, 2001], com 10 árvores de decisão; quantidade de variáveis utilizadas para a seleção randômica igual a $\sqrt{p}$, onde p é o número total de *matchers* disponíveis; e tamanho da base de treino de 50% para a *BDRI* e 40% para a *Dexter*. Também optamos por validar os experimentos executando 30 vezes cada tarefa de casamento.

[1] http://secondstring.sourceforge.net/

5.3 Resultados Experimentais

Para analisar a eficácia dos métodos com a inclusão dos *matchers* baseados em instância, foram executados experimentos nas bases *BDRI* e *Dexter* nos métodos COMA, ALMa e RFSM. Os resultados foram analisados em nível de precisão, revocação e medida-F e foram comparados com a execução dos métodos apenas com matchers de esquema.

Destacamos que foram realizadas operações de limpeza e padronização nos valores dos atributos presentes nas bases de dados. Também realizamos as devidas conversões de valores nos atributos numéricos e dimensionais para que todos permanecessem na mesma unidade de medida. Todas as comparações realizadas nesses experimentos foram confirmadas realizando o teste t com 95% de confiança.

As Tabelas 5.1 apresentam os resultados dos experimentos nos métodos COMA, ALMa e RFSM, respectivamente, na base de dados *BDRI*.

Nos resultados do COMA e do ALMa observa-se aumento na precisão e na medida-F nos experimentos com *matchers* de instância em relação aos apenas com *matchers* de esquema. Porém, a revocação foi ligeiramente menor. Nos resultados do RFSM as três métricas obtiveram aumento nos experimentos usando *matchers* baseados em instância.

A diminuição da revocação ocorre porque os *matchers* de instância são mais rigorosos nos seus resultados. Quando não existem correspondências entre as instâncias dos atributos os valores são muito baixos e quando os atributos são de classes diferentes o matcher recebe valor 0.

Na Figura 5.1, podemos observar uma análise detalhada dos resultados da medida-F. Observa-se que na maioria dos casos, os melhores resultados foram os dos experimentos utilizando *matchers* de instância. As exceções ocorreram no COMA na categoria CAM e no RFSM na categoria TV, em que o resultado ficou equivalente. Algumas tarefas de casamento do COMA possuem poucas correspondências, o que resultou em valores baixos para os *matchers*. Para gerar os resultados o método COMA calcula a média de todos os valores e, como os *matchers* instância estavam com valores baixos, resultou em uma média baixa.

Nos experimentos realizados com a base de dados *Dexter*, Tabelas 5.2, podemos verificar que os resultados apresentaram o mesmo padrão dos resultados com a base *BDRI*.

Observando a medida-F na Figura 5.2 podemos perceber que em todos os casos os resultados dos experimentos utilizando *matchers* de instância foram melhores que os resultados dos experimentos utilizando apenas *matchers* de esquema. Obtivemos um

COMA - BDRI

	precisão		revocação		medida-f	
	c/ instância	**s/ instância**	**c/ instância**	**s/ instância**	**c/ instância**	**s/ instância**
CAM	**0.840**	0.822	0.860	**0.966**	0.841	**0.878**
CEL	**0.885**	0.812	0.896	**0.967**	**0.887**	0.877
NOTE	**0.602**	0.546	0.929	**0.941**	**0.714**	0.672
TV	**0.629**	0.566	0.655	**0.666**	**0.633**	0.598

ALMa - BDRI

	precisão		revocação		medida-f	
	c/ instância	**s/ instância**	**c/ instância**	**s/ instância**	**c/ instância**	**s/ instância**
CAM	**0.733**	0.724	0.951	**0.954**	**0.806**	0.802
CEL	**0.813**	0.796	**0.976**	**0.976**	**0.872**	0.858
NOTE	**0.853**	0.840	0.971	**0.972**	**0.903**	0.887
TV	**0.687**	0.665	0.884	**0.885**	**0.752**	0.737

RFSM - BDRI

	precisão		revocação		medida-f	
	c/ instância	**s/ instância**	**c/ instância**	**s/ instância**	**c/ instância**	**s/ instância**
CAM	**0.787**	0.780	**0.711**	0.677	**0.722**	0.696
CEL	**0.852**	0.843	**0.795**	0.766	**0.807**	0.786
NOTE	**0.950**	0.946	**0.949**	0.905	**0.944**	0.917
TV	0.773	**0.777**	**0.613**	0.581	**0.659**	0.637

Tabela 5.1. Resultados dos experimentos na base de dados *BDRI*

ganho médio de 2,83% em precisão e 2.1% em medida-f nos experimentos feitos com a base de dados *BDRI* e tivemos um aumento de 14.64% de precisão e de 8.05% de medida-f nos experimentos feitos na base de dados *Dexter*.

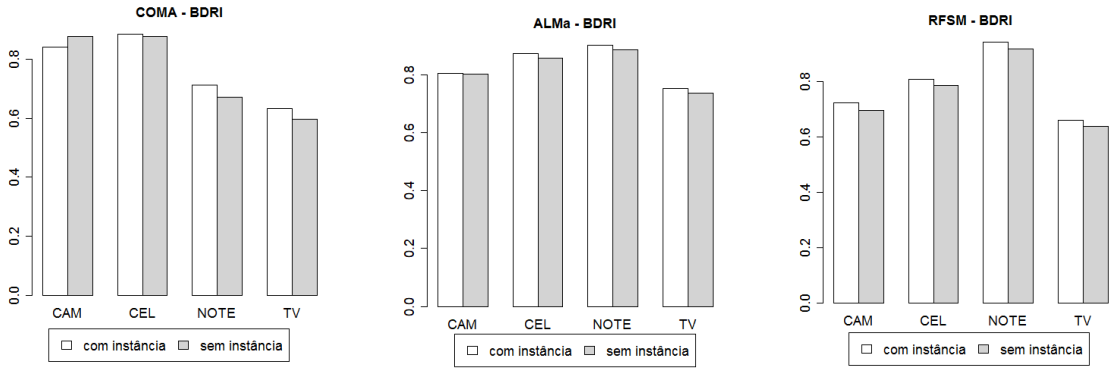

Figura 5.1. Resutados do F-Measure na base de dados *BDRI*

5.4 Utilização dos Matchers

O objetivo desse experimento é identificar que tipo de *matchers*, os baseados em esquema ou os baseados em instância, foram mais utilizada pelos métodos. Destacamos

COMA - Dexter

	precisão		revocação		medida-f	
	c/ instância	**s/ instância**	**c/ instância**	**s/ instância**	**c/ instância**	**s/ instância**
CAMDX	**0.684**	0.474	0.513	**0.607**	**0.562**	0.519
MONDX	**0.929**	0.764	**0.609**	**0.609**	**0.728**	0.662
TVDX	**0.582**	0.363	0.741	**0.779**	**0.639**	0.483

ALMa - Dexter

	precisão		revocação		medida-f	
	c/ instância	**s/ instância**	**c/ instância**	**s/ instância**	**c/ instância**	**s/ instância**
CAMDX	**0.671**	0.617	**0.751**	0.739	**0.726**	0.680
MONDX	**0.754**	0.727	**0.966**	0.952	**0.812**	0.790
TVDX	**0.675**	0.663	**0.858**	0.767	**0.730**	0.708

RFSM - Dexter

	precisão		revocação		medida-f	
	c/ instância	**s/ instância**	**c/ instância**	**s/ instância**	**c/ instância**	**s/ instância**
CAMDX	**0.708**	0.683	0.503	**0.508**	**0.573**	0.566
MONDX	**0.836**	0.792	**0.695**	0.607	**0.734**	0.665
TVDX	0.790	**0.809**	**0.760**	0.726	**0.754**	0.749

Tabela 5.2. Resultados dos experimentos na base de dados *Dexter*

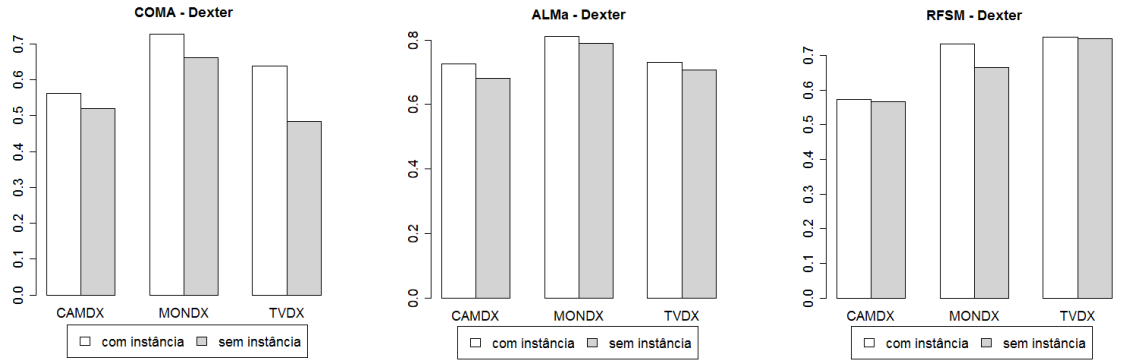

Figura 5.2. Resutados do F-Measure na base de dados *Dexter*

que esses experimentos foram realizados apenas nos métodos ALMa e RFSM, pois estes são métodos de aprendizagem de máquina que selecionam os *matchers* utilizados. O COMA, ao contrário, usa sempre todos os matchers disponíveis.

Para realizar esse experimento, foram contabilizados os *matchers* usados em todas as tarefas de casamento no ALMa e no RFSM. Foram analisadas mais de mil árvores no ALMa e mais de 20 mil no RFSM. No ALMa, foram observadas apenas as árvores usadas como membros dos comitê de decisão e no RFSM, as árvores das 30 rodadas em que foram utilizadas a maior quantidade de *matchers* de instância. Essas árvores equivalem àquelas utilizadas para realizar os experimentos da Seção 5.3. Para contabilizar a frequência de *matchers* utilizados, calculamos o percentual do seu uso em cada árvore utilizada nos experimentos e, ao final, computamos a média por tarefa

de casamento. Nas Figuras 5.3 e 5.4 apresentamos a variação da frequência do uso dos *matchers* de instância nas bases de dados *BDRI* e *Dexter* nos métodos ALMa e RFSM.

Observando os gráficos, podemos perceber que a frequência de *matchers* usando pelo RFSM é maior em comparação à frequência apresentada pelo ALMa, em ambas bases de dados. Na base de dados *BDRI*, podemos observar que a amplitude das categorias foram menores nos experimentos realizados com o método ALMa. No ALMa e no RFSM, a maior amplitude, ou seja, a maior variabilidade de árvores com *matchers* de instância, foi da categoria NOTE e a menor foi a CEL. Na base de dados *Dexter*, a categoria MONDX apresentou amplitude semelhante nos dois experimentos. Entretanto, as categorias CAMDX e TVDX foi bem menor nos experimentos com o método ALMa. Nos dois métodos, a categoria com menor amplitude foi a CAMDX e a maior MONDX.

Em relação a simetria dos dados, na base *BDRI*, os experimentos com o ALMa nas categorias CEL, NOTE e TV apresentaram assimetria positiva, enquanto que a CAM apresentou simetria. No método RFSM, as categorias CAM, NOTE e TV apresentaram a assimetria positiva e a CEL assimetria negativa. Na *Dexter*, nos dois experimentos, a maioria das categorias apresentou assimetria positiva, com exceção da TVDX, que apresentou simetria no método ALMa e assimetria negativa na RFSM.

Nos experimentos realizados com a base de dados *BDRI*, as categorias CAM, CEL e TV apresentaram valores discrepantes, representados na figura como um ponto externo ao gráfico. Nos experimentos com RFSM, os valores discrepantes apareceram nas categorias CAM, CEL e TV. Nesses casos, ocorreram tarefas de casamento que utilizaram uma quantidade maior de *matchers* de instância. Entretanto, isso não ocorreu nos experimentos com a base de dados *Dexter*.

Os métodos apresentam um percentual baixo da frequência do uso de *matchers* de instância, uma vez que são mais precisos nos seus resultados. Se um par de atributos for diferente, então suas instâncias serão diferentes, o que faz os resultados dos *matchers* serem baixos. Como a quantidade de pares falsos é muito maior que a de pares verdadeiros, nem sempre esses valores trazem vantagem para o resultado final. No entanto, como podemos observar nos resultados da precisão, revocação e medida-F, mesmo com a pouca utilização de *matchers* de instância, eles auxiliaram de forma positiva nos resultados.

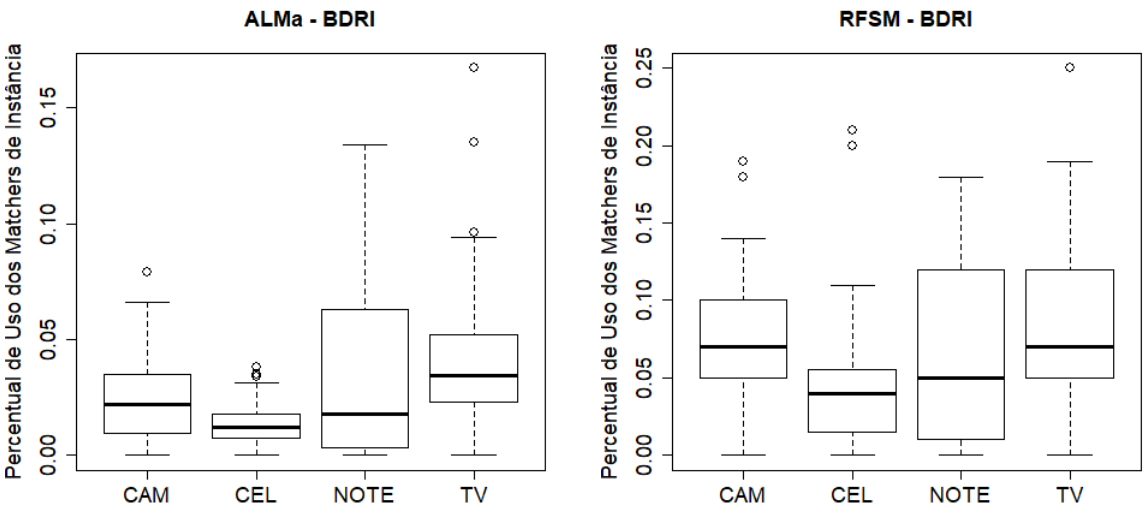

Figura 5.3. Frequência de *Matchers* na base de dados *BDRI*.

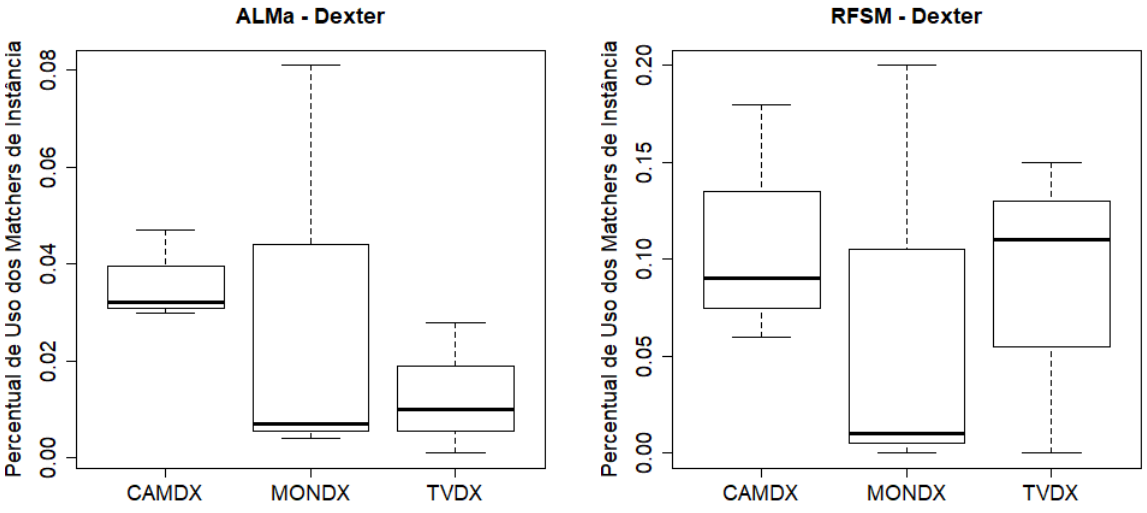

Figura 5.4. Frequência de *Matchers* base de dados *Dexter*.

5.5 Estudo de Parâmetros utilizados nos Métodos de Casamento de Esquema

Antes de executar os experimentos apresentados na Seção 5.3, realizamos um estudo para identificar quais configurações dos métodos COMA, ALMa e RFSM poderiam contribuir para melhorar os resultados dos métodos em termos de precisão, revocação e medida-f. Para decidir quais resultados seriam usados, demos prioridade para as con-

figurações com maiores valores da medida-f nos experimentos que utilizaram instância e nos que não utilizaram instância.

O método COMA tem suporte a uma série de configurações, entre elas estão a agregação dos valores dos *matchers* e a forma de seleção dos pares correspondentes. Os autores recomendam como medida de seleção uma combinação dos resultados do MaxDelta e do *Threshold*. Decidimos permanecer com as configurações recomendadas para os valores do MaxDelta δ, como 0.02, e com a média como método de agregação. Entretanto, variamos os valores do *Threshold*, aqui representado pelo parâmetro th, com valores entre 0.5 e 0.8.

Nas Figuras 5.5 e **??** apresentamos os resultados do COMA na *BDRI*. Observamos que a revocação obtém queda, conforme aumentamos o valor do th. Entretando, essa queda é mais discreta com o uso de instâncias. Os resultados da medida-f são crescentes nos experimentos utilizando instâncias, porém decrescem nos experimentos sem instâncias. Escolhemos utilizar os resutados do $th = 0.5$, pois obtém os melhores resultados nos dois experimentos.

Na *Dexter*, observamos nas Figuras 5.6 e **??**, os resultados da revocação também foram melhores nos experimentos que utilizaram instância. A medida-f apresentou queda nos experimentos sem instância com o aumento do th. Os melhores resultados para os dois experimentos foram com $th = 0.6$.

No ALMa, mantivemos o número de árvores de decisões membros do comitê como 30 e o valor mínimo de confiança da decisão de uma árvore como 0.75. Realizamos o estudo modificando a quantidade de pares rotulados pelo usuário, uma vez que a cada rodada o método solicita que o usuário rotule alguns pares que geraram dúvida no comitê. Variamos esse parâmetro entre 5 e 25.

Nas Figuras 5.7 e **??** observamos os resultados do ALMa na *BDRI* e nas figuras 5.8, **??** apresentamos os resultados dos experimentos na *Dexter*. Observamos que os resultados melhoram conforme aumentamos o número de pares rotulados pelo usuário. Na *BDRI*, os resultados da medida-f quando solicitamos para o usuário rotular 25 pares foram os melhores. Para a *Dexter*, observamos que os resultados melhoram até 20 pares rotulados e apresentam uma leve queda com 25 pares. Para essa base de dados utilizamos os resultados com 20 pares rotulados.

O método RFSM permite a configuração da quantidade de árvores que irão compor a floresta, da quantidade de variáveis utilizadas pelo método para gerar uma árvore e o tamanho da base de treino. Mantivemos o tamanho padrão de 10 árvores compondo a floresta, como indicado por Breiman [Breiman, 2001]. Para a quantidade de variáveis, testamos o valor padrão $\sqrt{p}$, a metade de p e com o total de elementos em p. Alteramos o tamanho da base de treino entre 10% e 50%.

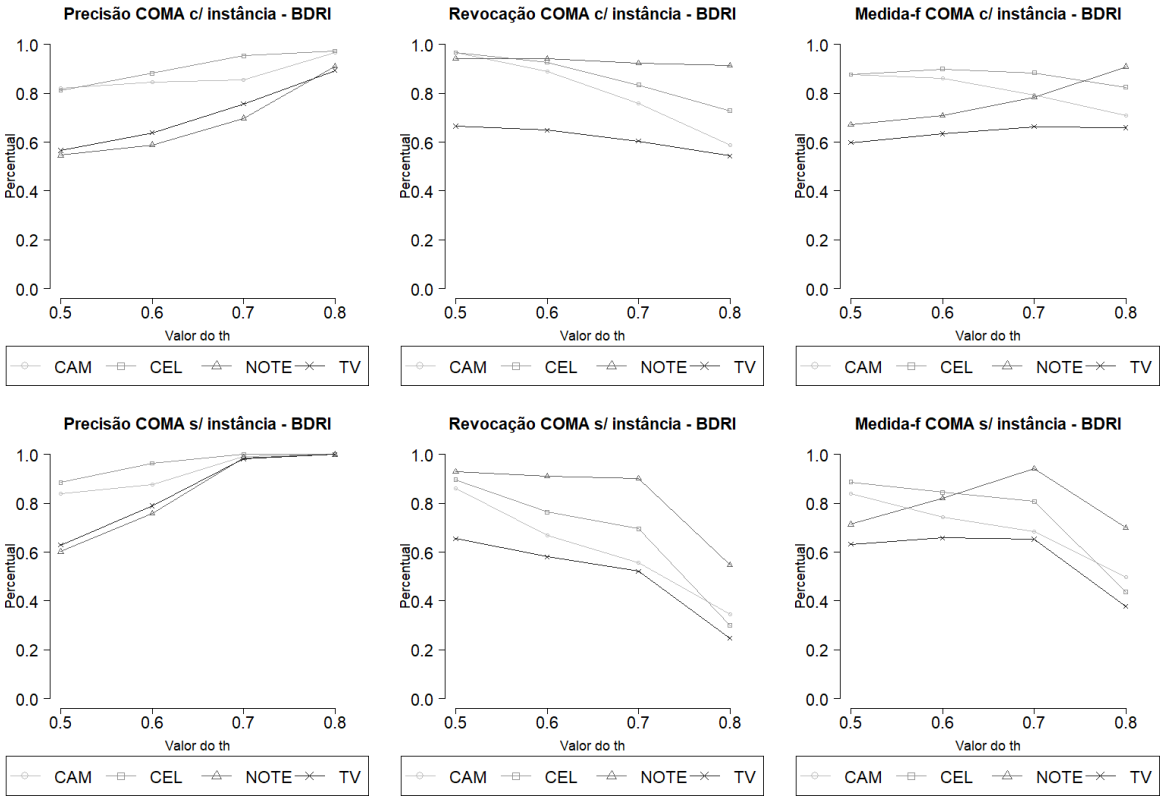

Figura 5.5. Resultados do método COMA na BDRI

Testamos a quantidade de variáveis apenas na base de dados *BDRI*, uma vez que queríamos padronizar essa configuração nas duas bases de dados. Os resultados podem ser observados na Figura 5.9 e na Figura **??**. Nesse experimento analisamos apenas os resultados da medida-f. Nas figuras podemos observar que os resultados decaem conforme aumentamos o valor de p. A configuração padrão $\sqrt{p}$ obteve os melhores resultados.

Após decidirmos o valor de p, realizamos novos experimentos avaliando o tamanho da base de treino. Nas Figuras 5.10 e **??** observamos os resultados do RFSM na *BDRI*. Percebemos que os dois experimentos apresentam crescimentos semelhantes nos resultados da precisão, revocação e medida-f. Na medida-f observamos um aumento gradual que obteve melhores resultados com 50% como tamanho da base de treino nos dois experimentos.

Nas figuras 5.11 e **??** apresentamos os resultados dos experimentos na *Dexter*. Os melhores resultados da medida-f foram apresentados com 40% como tamanho da base de treino. Apesar dos resultados do TVDX terem sido melhores com 50% de treino, mas utilizamos os resultados que foram melhores na maioria das categorias.

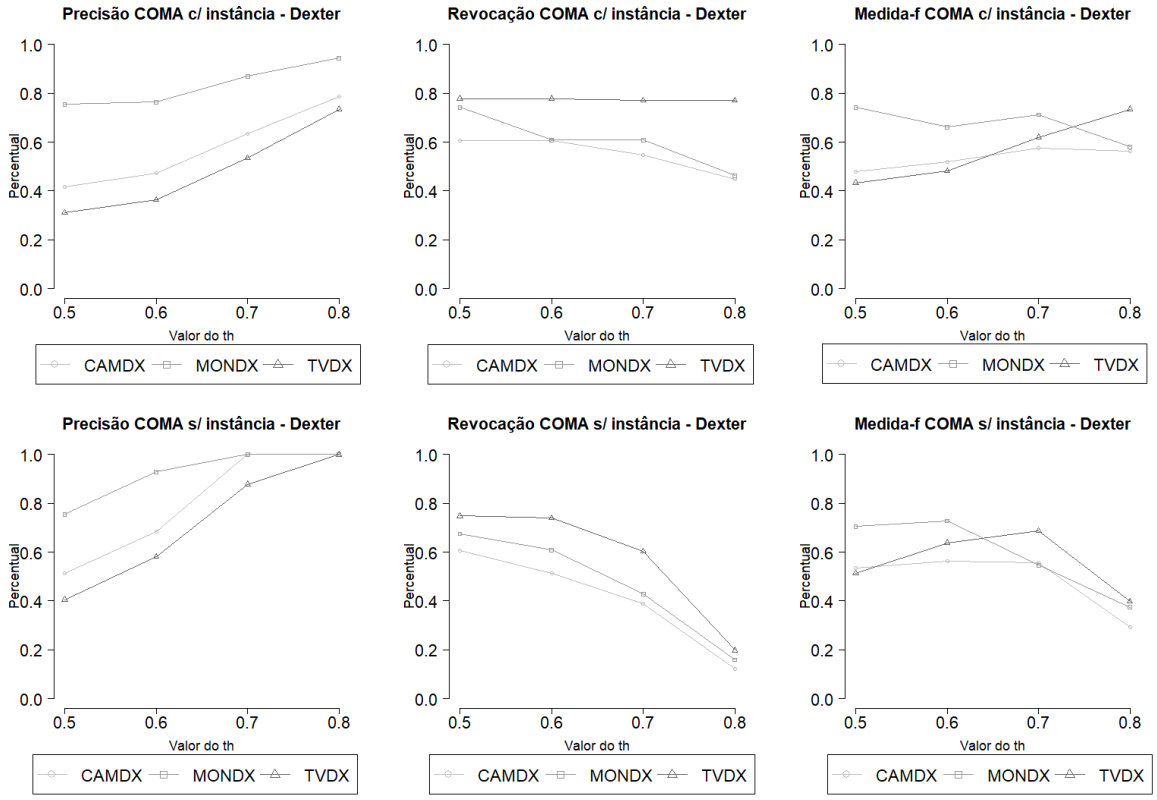

Figura 5.6. Resultados do método COMA na Dexter

5.6 Considerações Finais

Neste capítulo apresentamos uma avaliação experimental dos métodos COMA, ALMa e RFSM utilizando a informação de instância. Nos experimentos realizados, pudemos observar que utilizar *matchers* de instância influencia positivamente os resultados os métodos. Nos experimentos pode-se verificar melhoras nos resultados da precisão e medida-F, em relação as experimentos utilizando apenas informação de esquema. Esse comportamento foi observado nas duas bases de dados utilizadas, a base de dados *BDRI*, com informações em português e na *Dexter*, com informações em inglês.

Avaliamos também a frequência de *matchers* de instância utilizada pelos métodos nos experimentos. Podemos observar que o método RFSM utiliza uma quantidade maior de *matchers* do que o ALMa. Em ambos, porém, a frequência foi baixa. Entretanto, isso não foi um fator determinante para gerar piores resultados. Com isso, percebemos que os *matchers* de instância não precisam ser muito utilizados para gerarem melhores resultados, mas suas informações trazem de fato, vantagens para os métodos.

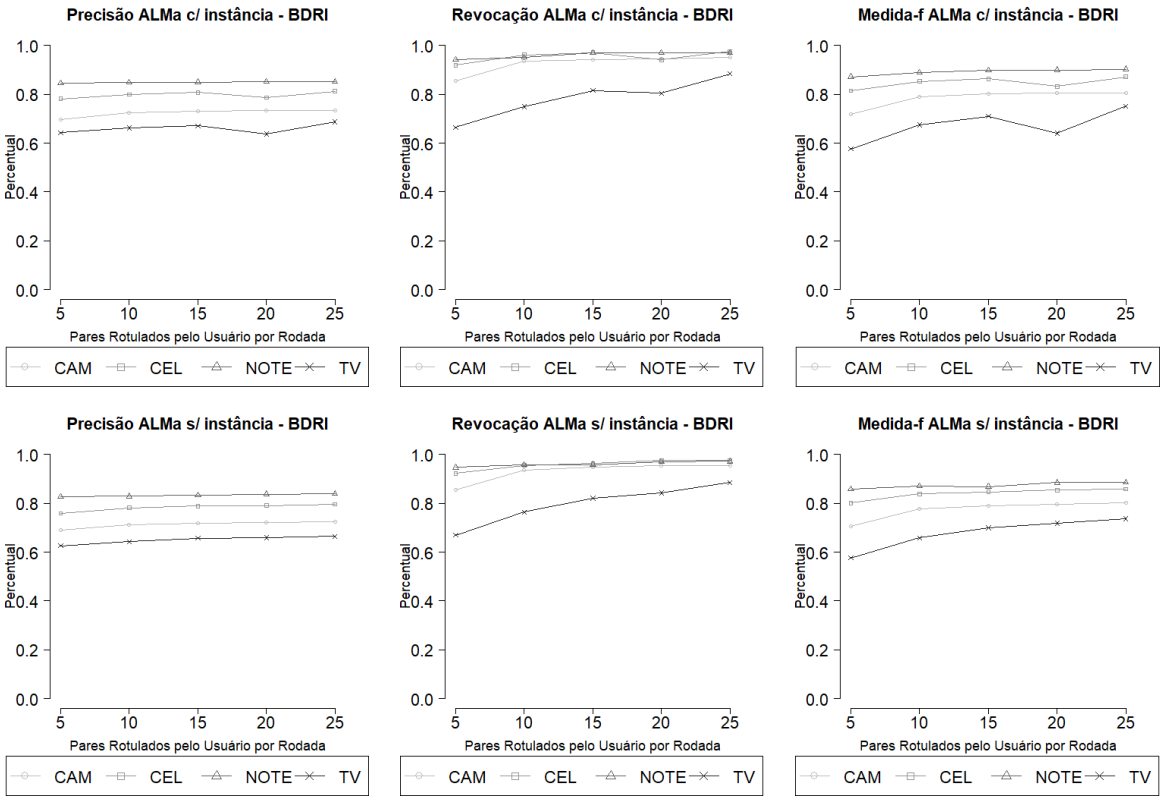

Figura 5.7. Resultados do método ALMa na BDRI

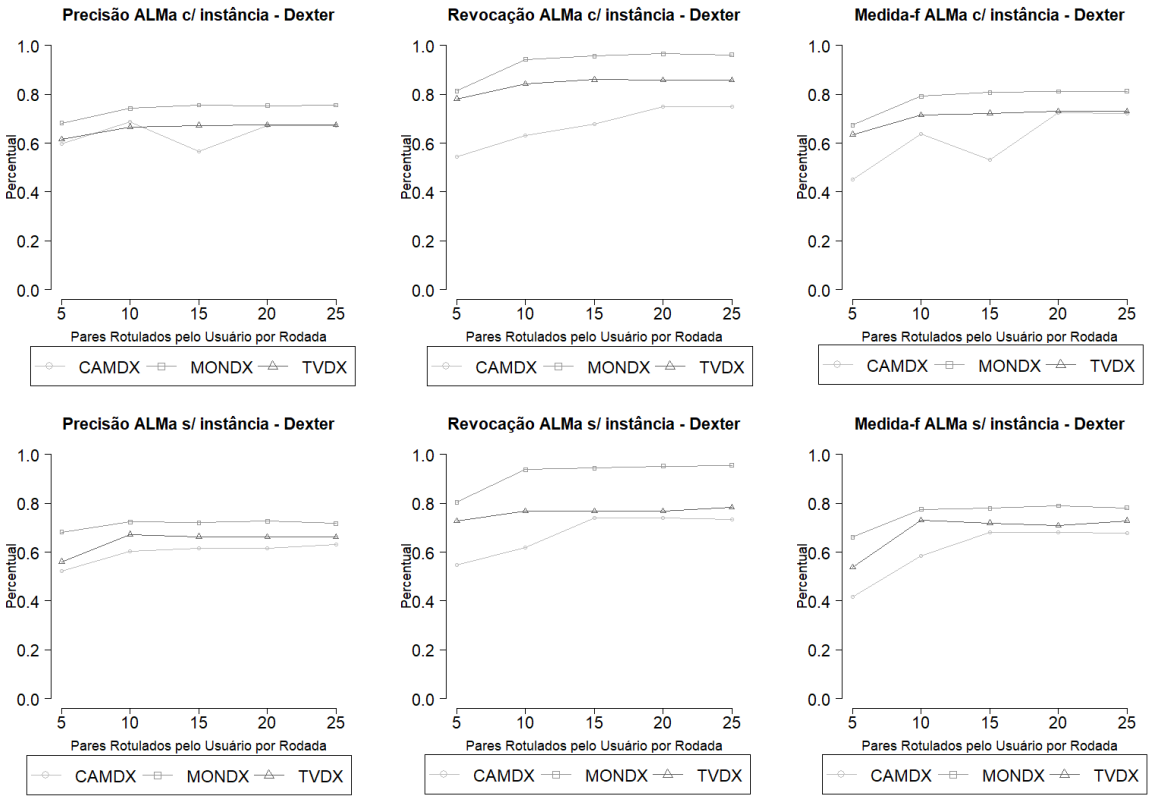

Figura 5.8. Resultados do método ALMa na Dexter

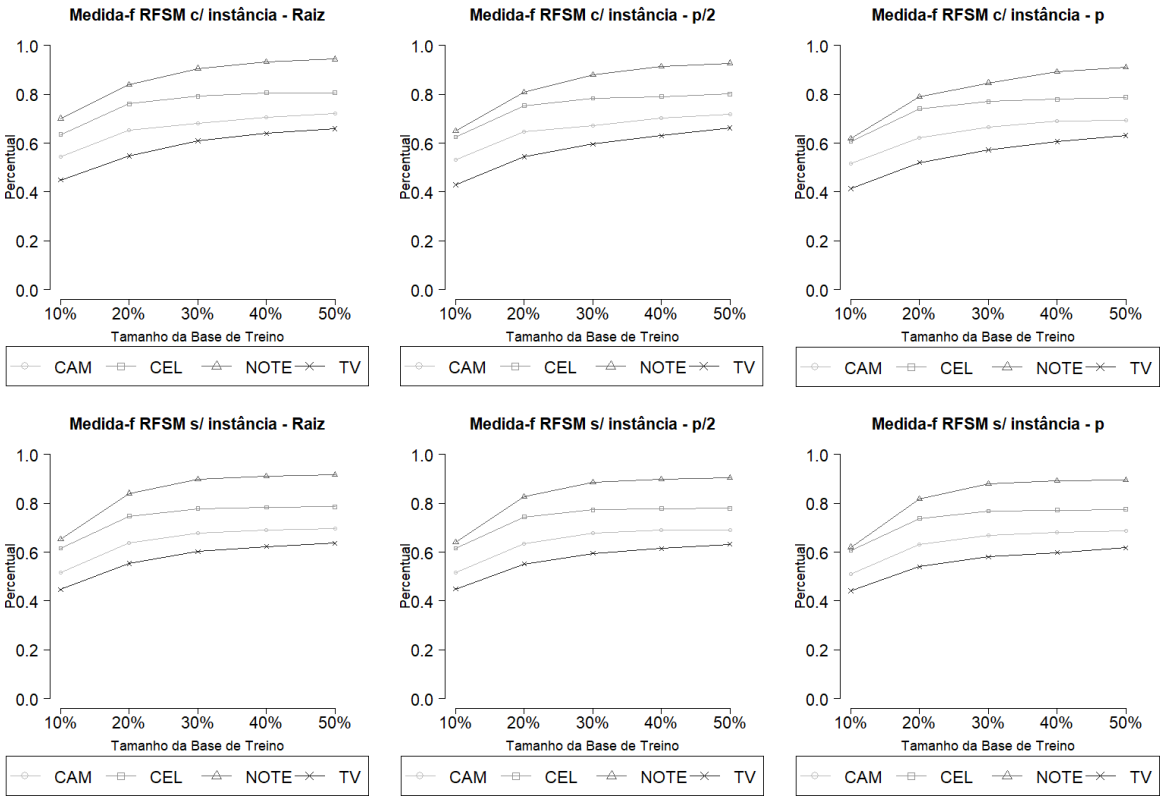

Figura 5.9. Resultados do método RFSM para quantidade da variáveis

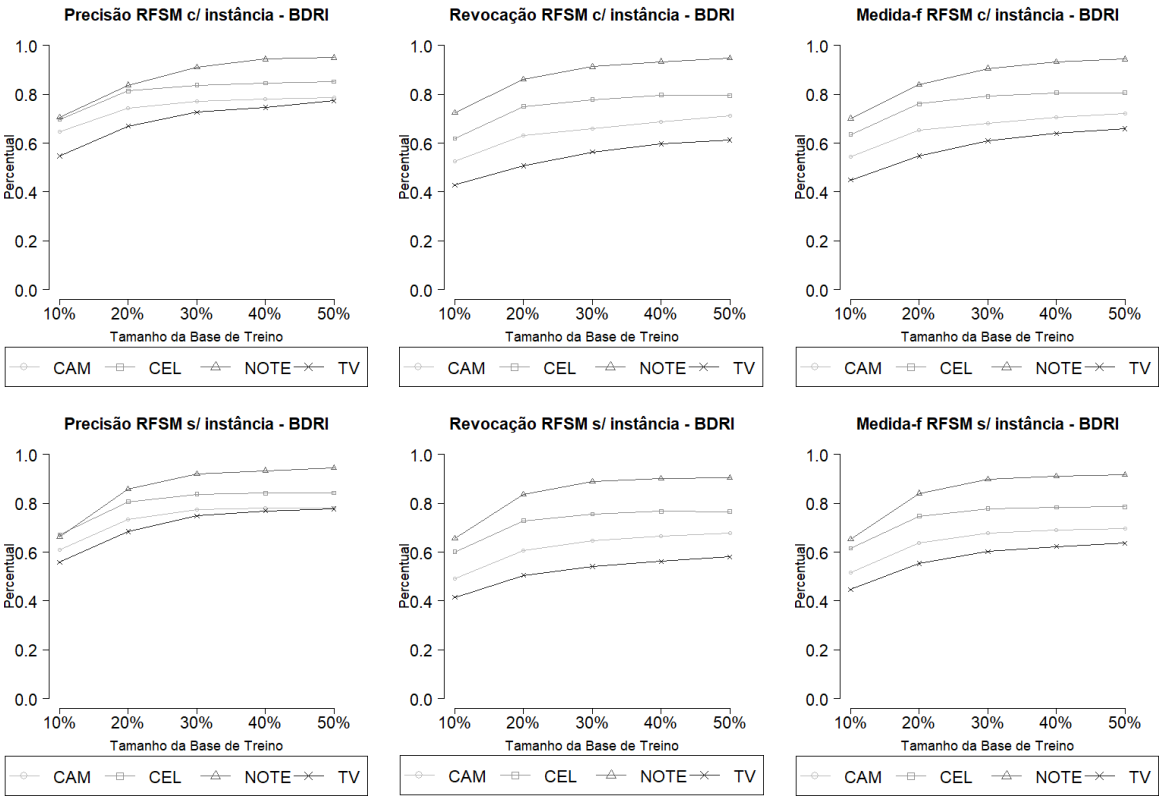

Figura 5.10. Resultados do método RFSM na BDRI

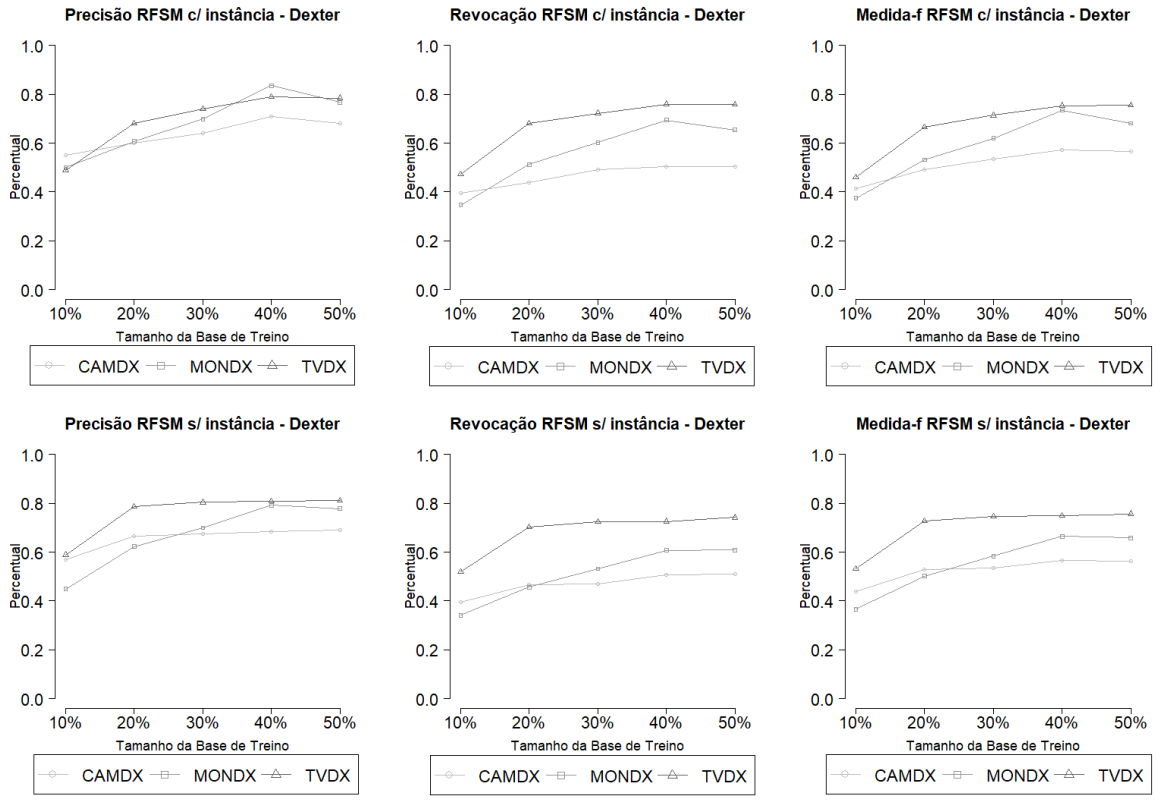

Figura 5.11. Resultados do método RFSM na Dexter

Capítulo 6

Conclusões e Trabalhos Futuros

Neste trabalho, apresentamos um estudo sobre o uso de informação de instância em métodos de casamento de esquemas no domínio de comércio eletrônico. Apesar do grande número de trabalhos sobre casamento de esquemas apresentado na literatura ao longo dos anos, este domínio foi abordado por poucos trabalhos. Assim, sendo este um domínio de grande importância prática e onde os esquemas apresentam grande heterogeneidade, são necessários estudos específicos com este enfoque. Especificamente, neste trabalho procuramos identificar se no domínio de comércio eletrônico o uso de *matchers* de instância pode ser mais informativo para os métodos de casamento de esquema.

Para tanto, primeiramente construímos duas bases de dados voltadas para casamento de esquemas, com informações de comércio eletrônico de diversas lojas e em diversas categorias. A primeira base de dados, *BDRI*, contém dados de nove lojas de brasileiras em português em quatro categorias. A segunda base de dados, *Dexter*, contêm informações de três lojas internacionais em inglês em três categorias. Apresentamos um estudo detalhado sobre elas no Capítulo 3. Identificamos que os atributos locais, agrupados por categorias, podem ser separados em grupos, diminuindo a quantidade de atributos locais em ambas bases de dados. Com isso, verificamos que os atributos possuem mais de uma representação em uma mesma categoria. Também foi possível dividir os atributos em cinco classes: Categóricos, Multicategóricos, Booleanos, Numéricos e Dimensionais. Essas classes são bem distribuídas nas bases de dados, entretanto, a classe dimensional é a que possui menor frequência.

Levando em consideração a classificação dos atributos, revisamos a literatura e selecionamos funções de similaridade que utilizassem instâncias e que pudessem ser aplicadas a cada das classes. Nos experimentos de validação, apresentados no Capítulo 4, selecionamos cinco funções de similaridade que consideramos como sendo as

melhores no nosso contexto.

Realizamos experimentos usando as bases de dados em três métodos de casamento de esquemas: COMA [Do & Rahm, 2002], que usa abordagem heurística; ALMa [Rodrigues et al., 2015], que usa aprendizado ativo; e RFSM [Rodrigues, 2017], que usa aprendizado de máquina supervisionado. Aplicamos os *matchers* selecionados nos métodos e comparamos os seus resultados sem a utilização desses *matchers*, ou seja, utilizando apenas *matchers* baseados em esquema. Os resultados indicaram que o uso de informação de instância traz melhoria para os métodos de casamento de esquemas. Nos experimentos, na maioria das categorias observamos que os resultados da precisão e medida-f com o uso de instâncias foram melhores que os sem o uso de instância, em ambas bases de dados. Tivemos um ganho médio de 2,83% em precisão e 2.1% em medida-f nos experimentos feitos com a base de dados *BDRI*. Já na base de dados *Dexter*, tivemos um aumento de 14.64% de precisão e de 8.05% de medida-f.

Também avaliamos a frequência de uso de *matchers* de instância e percebemos que, apesar do uso dos *matchers* de instância ser baixo, ainda gera melhorias nos resultados. Portanto, a frequência do uso não foi um fator determinante para a melhoria dos resultados. Mas, ainda assim, a utilização dos *matchers* de instância influenciou nos resultados.

Com esses resultados, podemos concluir que o uso da informação de instância auxilia os métodos de casamento de esquemas a obterem melhores resultados no domínio de comércio eletrônico, independente da abordagem que eles utilizam.

Como trabalhos futuros, nós pretendemos evoluir a pesquisa o problema de redes de casamentos de esquema (*matching network*). Nesse contexto, o casamento de esquemas não se limita a apenas duas lojas e as correspondências entre um par de lojas podem ser reutilizadas para outros pares de lojas. Além disso, considerando que as bases de dados construídas estão presentes em dois idiomas, pretendemos também explorar o casamento de esquemas multilinguístico. Isso para que os métodos existentes não se limitem a apenas um idioma e possam ser utilizados em uma escala global.

Referências Bibliográficas

[Baeza-Yates & Ribeiro, 2012] Baeza-Yates, R. & Ribeiro, B. d. A. N. (2012). *Modern information retrieval*. ACM Press Harlow, New York. ISBN 0-201-39829-X.

[Bellahsene et al., 2011] Bellahsene, Z.; Bonifati, A. & Rahm, E. (2011). *Schema Matching and Mapping*. Springer Publishing Company, Incorporated, 1st edição. ISBN 9783642165177.

[Bernstein et al., 2011] Bernstein, P. A.; Madhavan, J. & Rahm, E. (2011). Generic schema matching, ten years later. *PVLDB*, 4(11):695–701.

[Bilke & Naumann, 2005] Bilke, A. & Naumann, F. (2005). Schema matching using duplicates. Em *Proceedings of the 21st International Conference on Data Engineering*, ICDE '05, pp. 69--80, Washington DC USA. IEEE Computer Society.

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5--32. ISSN 1573-0565.

[Cohn et al., 1994] Cohn, D.; Atlas, L. & Ladner, R. (1994). Improving generalization with active learning. *Mach. Learn.*, 15(2):201--221. ISSN 0885-6125.

[De Carvalho et al., 2013] De Carvalho, M. G.; Laender, A. H. F.; GonçAlves, M. A. & Da Silva, A. S. (2013). An evolutionary approach to complex schema matching. *Inf. Syst.*, 38(3):302--316. ISSN 0306-4379.

[Do & Rahm, 2002] Do, H.-H. & Rahm, E. (2002). Coma: A system for flexible combination of schema matching approaches. Em *Proceedings of the 28th International Conference on Very Large Data Bases*, VLDB '02, pp. 610--621. VLDB Endowment.

[Doan et al., 2001] Doan, A.; Domingos, P. & Halevy, A. Y. (2001). Reconciling schemas of disparate data sources: A machine-learning approach. *SIGMOD Rec.*, 30(2):509--520. ISSN 0163-5808.

[Doan et al., 2012] Doan, A.; Halevy, A. & Ives, Z. (2012). *Principles of Data Integration*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edição. ISBN 0124160441, 9780124160446.

[Duchateau et al., 2009] Duchateau, F.; Coletta, R.; Bellahsene, Z. & Miller, R. J. (2009). (not) yet another matcher. Em *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pp. 1537--1540, New York NY USA. ACM.

[Gal, 2006] Gal, A. (2006). Why is schema matching tough and what can we do about it? *SIGMOD Rec.*, 35(4):2--5. ISSN 0163-5808.

[Hoffmann, 2016] Hoffmann, U. (2016). Learning to recommend equivalent products in e-commerce catalogs.

[Hoffmann et al., 2015] Hoffmann, U.; da Silva, A. S. & de Carvalho, M. G. (2015). Finding similar products in e-commerce sites based on attributes. Em *Proceedings of the 9th Alberto Mendelzon International Workshop on Foundations of Data Management, Lima, Peru, May 6 – 8, 2015.*

[Jaccard, 1912] Jaccard, P. (1912). The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37--50.

[Kagie et al., 2008] Kagie, M.; van Wezel, M. & Groenen, P. J. (2008). Choosing attribute weights for item dissimilarity using clikstream data with an application to a product catalog map. Em *Proceedings of the 2008 ACM Conference on Recommender Systems*, RecSys '08, pp. 195--202, New York, NY, USA. ACM.

[Kang & Naughton, 2003] Kang, J. & Naughton, J. F. (2003). On schema matching with opaque column names and data values. Em *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, SIGMOD '03, pp. 205--216, New York, NY, USA. ACM.

[Köpcke et al., 2012] Köpcke, H.; Thor, A.; Thomas, S. & Rahm, E. (2012). Tailoring entity resolution for matching product offers. Em *Proceedings of the 15th International Conference on Extending Database Technology*, EDBT '12, pp. 545--550, New York, NY, USA. ACM.

[Li et al., 2012] Li, X.; Dong, X. L.; Lyons, K.; Meng, W. & Srivastava, D. (2012). Truth finding on the deep web: Is the problem solved? *Proc. VLDB Endow.*, 6(2):97--108. ISSN 2150-8097.

[Liu, 2006] Liu, B. (2006). *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA. ISBN 3540378812.

[Madhavan et al., 2001] Madhavan, J.; Bernstein, P. A. & Rahm, E. (2001). Generic schema matching with cupid. Em *Proceedings of the 27th International Conference on Very Large Data Bases*, VLDB '01, pp. 49--58, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Manning & Schütze, 1999] Manning, C. D. & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA. ISBN 0-262-13360-1.

[Melnik et al., 2002] Melnik, S.; Garcia-Molina, H. & Rahm, E. (2002). Similarity flooding: A versatile graph matching algorithm and its application to schema matching. Em *Proceedings of the 18th International Conference on Data Engineering*, ICDE '02, pp. 117--, Washington, DC, USA. IEEE Computer Society.

[Mesquita et al., 2007a] Mesquita, F.; Barbosa, D.; Cortez, E. & da Silva, A. S. (2007a). Fledex: Flexible data exchange. Em *Proceedings of the 9th Annual ACM International Workshop on Web Information and Data Management*, WIDM '07, pp. 25--32, New York, NY, USA. ACM.

[Mesquita et al., 2007b] Mesquita, F.; da Silva, A. S.; de Moura, E. S.; Calado, P. & Laender, A. H. F. (2007b). Labrador: Efficiently publishing relational databases on the web by using keyword-based query interfaces. *Inf. Process. Manage.*, 43(4):983--1004. ISSN 0306-4573.

[Nguyen et al., 2011] Nguyen, H.; Fuxman, A.; Paparizos, S.; Freire, J. & Agrawal, R. (2011). Synthesizing products for online catalogs. *Proc. VLDB Endow.*, 4(7):409--418. ISSN 2150-8097.

[Qiu et al., 2015] Qiu, D.; Barbosa, L.; Dong, X. L.; Shen, Y. & Srivastava, D. (2015). Dexter: Large-scale discovery and extraction of product specifications on the web. *Proc. VLDB Endow.*, 8(13):2194--2205. ISSN 2150-8097.

[Rahm & Bernstein, 2001] Rahm, E. & Bernstein, P. A. (2001). A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334350. ISSN 10668888.

[Rijsbergen, 1979] Rijsbergen, C. J. V. (1979). *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, 2nd edição. ISBN 0408709294.

[Rodrigues, 2013] Rodrigues, D. (2013). Casamento de esquemas de banco de dados aplicando aprendizado ativo.

[Rodrigues, 2017] Rodrigues, D. (2017). *RFSM - Random Forest Schema Matching.* Tese de doutorado, Universidade Federal do Amazonas.

[Rodrigues et al., 2015] Rodrigues, D.; da Silva, A.; Rodrigues, R. & dos Santos, E. (2015). Using active learning techniques for improving database schema matching methods. Em *Neural Networks (IJCNN), 2015 International Joint Conference on*, pp. 1–8.

[Settles, 2012] Settles, B. (2012). *Active Learning.* Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool.

[Wong et al., 2011] Wong, T.-L.; Bing, L. & Lam, W. (2011). Normalizing web product attributes and discovering domain ontology with minimal effort. Em *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pp. 805--814, New York, NY, USA. ACM.