3DR-INDEXING: UM MÉTODO PARA IDENTIFICAÇÃO AUTOMÁTICA DOS MELHORES ATRIBUTOS DE INDEXAÇÃO EM DEDUPLICAÇÃO DE DADOS

LEVY DE SOUZA SILVA

3DR-INDEXING: UM MÉTODO PARA IDENTIFICAÇÃO AUTOMÁTICA DOS MELHORES ATRIBUTOS DE INDEXAÇÃO EM DEDUPLICAÇÃO DE DADOS

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

ORIENTADOR: MIRELLA MOURA MORO

Belo Horizonte

Junho de 2018

© 2018, Levy de Souza Silva.
Todos os direitos reservados.

Silva, Levy de Souza

S586t 3DR-Indexing: um Método para Identificação Automática dos Melhores Atributos de Indexação em Deduplicação de Dados / Levy de Souza Silva. — Belo Horizonte, 2018
xxiii, 78 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de Minas Gerais
Orientador: Mirella Moura Moro

1. Computação - Teses. 2. Deduplicação de Dados. 3. Resolução de Entidades. 4. Seleção de Atributos. 5. Indexação. I. Orientador. II. Título.

CDU 519.6*73(043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

3DR-INDEXING: UM MÉTODO PARA IDENTIFICAÇÃO AUTOMÁTICA DOS MELHORES ATRIBUTOS DE INDEXAÇÃO EM DEDUPLICAÇÃO DE DADOS

LEVY DE SOUZA SILVA

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROFA. MIRELLA MOURA MORO - Orientadora
Departamento de Ciência da Computação - UFMG

PROFA. ANA CAROLINA BRANDÃO SALGADO
Centro de Informática - UFPE

PROF. ANA PAULA COUTO DA SILVA
Departamento de Ciência da Computação - UFMG

PROF. CARLOS EDUARDO SANTOS PIRES
Departamento de Sistemas e Computação - UFCG

DRA. MICHELE AMARAL BRANDÃO
Pós Doc/Departamento de Ciência da Computação - UFMG

Belo Horizonte, 15 de junho de 2018.

Aos aos meus pais Jonas Trajano (in memoriam) e Eneida Jordão.

Agradecimentos

Em primeiro lugar, agradeço a Deus por ter me dado forças para continuar essa difícil caminhada acadêmica. Muitas vezes pensei em desistir devido a inúmeros fatores como distância da família e bastante solidão. No entanto, depois que mudei de orientador, em meados do segundo semestre, o mestrado começou a tomar rumo para mim. Troquei de laboratório, conheci novos amigos, e tudo foi se encaixando como tinha que ser. Além disso, agraço a Deus pela oportunidade de estudar em um dos melhores departamentos de computação do país, o famoso DCC-UFMG. Sem sombra de dúvida, qualquer estudante de computação do país queria estar no meu lugar. O DCC-UFMG é sensacional e indescritível. Por mais que eu te conte, você nunca saberá. Você precisa passar por aqui para sentir isso na pele. Em seguida, gostaria de agradecer aos meus familiares pelo incentivo e apoio em todas as minhas decisões. Em especial a minha mãe, que não segurava o choro quase todas às vezes que falava comigo pelo telefone. Também gostaria de agradecer especialmente a minha orientadora Mirella Moro pelos ensinamentos transmitidos de forma clara e objetiva. Confesso que aprendi mais com ela do que com as disciplinas que cursei nesse departamento de computação e eu teria desistido de tudo isso se não tivesse conhecido a Mirella. Ela sempre dizia: você consegue! vá em frente! vai firme! Um bom orientador faz toda a diferença! Ademais, não poderia esquecer do meu braço direito em BH, Laís Mota, sempre disposta a me ajudar em tudo que precisava e da Dra. Michele A. Brandão que diariamente me auxiliava nos quesitos acadêmicos e quase todas as tardes me acompanhava no café. Por fim, gostaria de agradecer aos demais colegas do laboratório CS-X; aos amigos da Escola Jornalista Manuel Amaral que, mesmo distantes, viveram esses dias comigo; a professora Áurea que não mediu esforços para liberar meu afastamento; e ao Governo do Estado de Pernambuco por ter financiado integralmente meus estudos. A todos vocês, Muito Obrigado!

"A vida é uma corrida,
que não se corre sozinho.
Que vencer não é chegar,
é aproveitar o caminho."
(Bráulio Bessa)

Resumo

Deduplicação de dados é a tarefa de encontrar registros duplicados em bancos de dados. Registros duplicados são instâncias de dados que representam o mesmo objeto no mundo real. Via de regra, a deduplicação é composta por três etapas: indexação (que atribui uma chave de bloco a cada registro), clusterização (que agrupa os registros de acordo com as chaves) e classificação (que compara e classifica os registros dentro de cada grupo). Nosso estudo foca na etapa de indexação, a qual cria estruturas de chaves de bloco para agrupar registros similares. A indexação deve ser eficaz para melhor distinguir os registros e eficiente para permitir um rápido processo de deduplicação. Assim, nesta etapa, um atributo é escolhido e seu valor é codificado por uma função para produzir o valor da chave de bloco. Nos trabalhos atuais, os atributos de indexação são escolhidos manualmente por usuários especialistas sobre o domínio dos dados, o que requer tempo, além de aumentar o custo total da tarefa. Por este motivo, nós apresentamos o método *3DR-Indexing*, que automaticamente seleciona os melhores atributos para a etapa de indexação. Além disso, nós analisamos o impacto do atributo de indexação sobre etapas da deduplicação. Por fim, nós avaliamos o atributo de indexação e o método proposto em um conjunto de 13 *datasets* distintos, ou seja, com diferentes domínios, quantidades de registros duplicados e total de instâncias. Os resultados indicam que o atributo de indexação possui maior efeito sobre as etapas da deduplicação. Por exemplo, o melhor atributo de indexação difere do pior em uma média de 44% em termos de *F-Measure* considerando todos os conjuntos de dados. Ademais, o método *3DR-Indexing* tem resultados significantes, pois ele identifica o melhor atributo de indexação em 10 dos 13 conjuntos de dados utilizados.

Palavras-chave: Deduplicação de Dados, Indexação, Resolução de Entidades, Seleção de Atributos.

Abstract

Data deduplication aims to find duplicate records in databases. Duplicate records are data instances that represent the same object in the real world. Usually, the deduplication has three steps: indexing (which assigns a block key value for each record), clustering (which groups the records with similar block key) and classification (which compares and classifies the records within the same group). Our study focuses on the indexing step, which creates block key structures to group similar records. Indexing must be effective as to better distinguish the values and efficient as to allow faster deduplication runtime. Thus, in this step, an attribute is chosen and its value is encoded by a function to produce the block key value. Currently, the indexing attributes are chosen manually by expert users, which takes time and increases the task total cost. Therefore, we present the method 3DR-Indexing, which automatically selects the best attributes for the indexing step. Furthermore, we analyze the impact of the indexing attribute over data deduplication steps. Finally, we evaluate the indexing attribute and the proposed method over 13 distinct datasets, that is, with different domains, number of duplicate records and the total of instances. The results indicate the indexing attribute has highest impact over deduplication process. For instance, the best indexing attribute differs from the worst one by an average of 44% in terms of F-Measure considering all datasets. Moreover, the 3DR-Indexing has significant results, because it identifies the best indexing attribute in 10 out of 13 datasets.

Keywords: Data Deduplication, Indexing, Record Linkage, Entity Resolution, Attribute Selection.

Lista de Figuras

2.1 Processo de deduplicação de dados (adaptado de [Christen, 2012a]) 8
2.2 Exemplo do algoritmo *Sorted Neighborhood* (adaptado de [Yan et al., 2007]) 11

4.1 Aplicação do método *3DR-Indexing* no processo de deduplicação de dados 29

5.1 Projeto experimental com as dimensões avaliadas e variáveis consideradas em cada dimensão 32

6.1 Efeito explicado por cada fator e suas iterações no projeto fatorial 2^3r . . . 42
6.2 Avaliação do atributo de indexação nos métodos *Schema-Agnostic* e *Schema-Based Configurations* no conjunto de dados *DatasetSynthetic* . . . 44
6.3 Avaliação do atributo de indexação nos métodos *Schema-Agnostic* e *Schema-Based Configurations* em dados reais 45
6.4 Tempo de execução do atributo de indexação nos métodos *Schema-Agnostic* e *Schema-Based Configurations* 46
6.5 Avaliação do atributo de indexação nos conjuntos de dados sintéticos . . . 48
6.6 Avaliação do atributo de indexação nos conjuntos de dados reais 49
6.7 Avaliação do *3DR-Indexing* de acordo com o total de registros duplicados . 51
6.8 Avaliação do *3DR-Indexing* de acordo com o total de instâncias 53
6.9 Avaliação do *3DR-Indexing* em conjuntos de dados reais 55
6.10 Revocação e precisão por atributo de indexação em dados sintéticos 57
6.11 Classificação geral dos atributos de indexação 66

Lista de Tabelas

1.1 Identificando registros duplicados em duas fontes de dados 3

2.1 Codificação *Soundex* (adaptado de [Christen, 2012a]) 9
2.2 Exemplo de codificação *Soundex* e *Suffix* 9

3.1 Visão geral dos trabalhos relacionados por etapa da deduplicação 22
3.2 Visão geral dos trabalhos relacionados de acordo com os experimentos . . . 22
3.3 Visão geral dos trabalhos relacionados considerando o domínio dos dados . 23

4.1 Registros duplicados em duas fontes de dados (Tabela 1.1 novamente) . . . 28
4.2 Seleção dos melhores atributos para a etapa de indexação 28

5.1 Configurações do projeto fatorial 33
5.2 Configurações para geração dos conjuntos de dados sintéticos 37
5.3 Conjuntos de dados experimentais 38

6.1 Resultados do projeto fatorial em termos de *F-Measure* 43
6.2 *Ranking* de atributos produzidos pelo método *3RD-Indexing* considerando o total de registros duplicados 51
6.3 *Ranking* de atributos produzidos pelo método *3RD-Indexing* considerando o total de instâncias 53
6.4 *Ranking* de atributos produzidos pelo *3RD-Indexing* em dados reais 55
6.5 Revocação, Precisão, *F-Measure* e Tempo Médio de execução de acordo com o atributo de indexação em dados reais 58
6.6 Coeficiente de correlação de *Spearman* entre o *ranking* dos resultados e o *ranking* produzido pela combinação das métricas em cada conjunto de dados 60
6.7 Valor máximo de *F-Measure* em dados reais: *3DR-Indexing* vs. *Baselines* . 60
6.8 Avaliação da eficácia da combinação de atributos no *dataset Restaurant* . . 63
6.9 Avaliação da eficácia da combinação dos atributos mais relevantes segundo o método *3DR-Indexing* nos *datasets CDs* e *DatasetSynthetic* 63

6.10 Avaliação do tempo médio de execução da combinação de atributos mais relevantes na etapa de indexação 64
6.11 Análise do melhor e pior valor de *F-Measure* nos conjuntos de dados . . . 65
6.12 Avaliação dos resultados do processo de deduplicação de dados considerando os tipos de atributos classificados pelas métricas apresentadas no Capítulo 4 67

Sumário

Agradecimentos **ix**

Resumo **xiii**

Abstract **xv**

Lista de Figuras **xvii**

Lista de Tabelas **xix**

Lista de Abreviaturas **xxv**

1 Introdução **1**

1.1 Exemplo de Aplicação do Processo de Deduplicação de Dados 3

1.2 Soluções Atuais 4

1.3 Objetivos 5

1.4 Contribuições 5

1.5 Organização do Trabalho 6

2 Processo de Deduplicação de Dados **7**

2.1 Visão Geral 7

2.2 Etapa de Indexação 8

2.3 Etapa de Clusterização 10

2.4 Etapa de Classificação 11

2.5 Considerações Finais 12

3 Trabalhos Relacionados **13**

3.1 Uma Visão Geral sobre Deduplicação de Dados 14

3.2 Indexação e Clusterização dos Registros 15

3.3 Classificação dos Registros 18

3.4 Seleção de Atributos Relevantes para a Deduplicação de Dados 18
3.5 Deduplicação de dados Orientada ao Domínio 19
3.6 Discussões 21

4 Método para Seleção de Atributos Relevantes 25
4.1 Métricas para Identificação da Relevância de um Atributo no Conjunto de Dados 26
4.2 O Método 3DR-Indexing 27
4.3 Exemplo 28
4.4 Deduplicação de Dados Utilizando o Método *3DR-Indexing* 29
4.5 Considerações Finais 30

5 Projeto Experimental 31
5.1 Metodologia dos Experimentos 31
5.1.1 Análise do Atributo de Indexação sobre as Etapas do Processo de Deduplicação de Dados 32
5.1.2 Análise do Atributo de Indexação nos Métodos *Schema-Agnostic* e *Schema-Based Configurations* 33
5.1.3 Análise do Atributo de Indexação nos Algoritmos *Standard Blocking* e *Sorted Neighborhood* 34
5.1.4 Avaliação do Método *3DR-Indexing* 35
5.1.5 Análise da Combinação de Atributos na Etapa de Indexação do Processo de Deduplicação 35
5.2 Conjuntos de Dados Experimentais 36
5.3 Configurações e Métricas de Avaliação 38
5.4 Considerações Finais 40

6 Resultados e Discussões 41
6.1 Projeto Fatorial 41
6.2 Análise do Atributo de Indexação na Deduplicação de Dados 43
6.2.1 Análise do Atributo de Indexação nas Abordagens *Schema-Agnostic* e *Schema-Based Configurations* 43
6.2.2 Avaliação do Atributo de Indexação nos Algoritmos *Standard Blocking* e *Sorted Neighborhood* 47
6.3 Avaliação do Método *3DR-Indexing* 49
6.3.1 *Trade-off* entre Eficiência e Eficácia 50
6.3.2 Avaliação do *3DR-Indexing* em Dados Sintéticos 50
6.3.3 Avaliação *3DR-Indexing* em Dados Reais 54

6.3.4 Avaliação do *3DR-Idexing* Considerando outra Perspectiva de Qualidade 56
6.3.5 Avaliação da Correlação entre *Rankings* 58
6.3.6 Comparação do Método *3RD-Indexing* com *Baselines* 60
6.4 Análise da Combinação de Atributos na Etapa de Indexação 62
6.5 Discussões 65

7 Conclusões e Trabalhos Futuros **69**

Referências Bibliográficas **73**

Lista de Abreviaturas

Sigla	**Descrição**
BK	Chave de Bloco (do inglês *Block Key*)
BKV	Valor de Chave de Bloco (do inglês *Block Key Value*)
VP	Verdadeiros Positivos
FP	Falsos Positivos

Capítulo 1

Introdução

Deduplicação de dados é o problema de identificar registros duplicados em fontes de dados (i.e., bancos de dados e conjuntos de dados). Registros duplicados/duplicatas são instâncias de dados que representam o mesmo objeto no mundo real. O problema da deduplicação vem sendo estudado em ambas academia e indústria. Por exemplo, na biblioteca digital de Ciência da Computação, DBLP, existem pelo menos 799 artigos publicados sobre o assunto entre os anos de 2012 e 2018 (i.e., uma média de 114 estudos por ano). Considerando os anos anteriores de 2005 a 2011, há aproximadamente 258 estudos, ou seja, um aumento de mais de 200% em sete anos.[1]

Deduplicação de dados possui várias aplicações. Por exemplo, ela ajuda a identificar produtos duplicados em lojas onlines [Su et al., 2010], profissionais duplicados em sistemas de saúde [Carvalho et al., 2015], contatos duplicados em dispositivos móveis [Borges et al., 2017] e duplicatas em sistemas de registro de homicídios [Sadinle & Fienberg, 2013]. Em um contexto mais amplo, em recuperação de informação, ela é importante para remover documentos duplicados dos resultados retornados por motores de busca, bibliotecas digitais e sistemas de indexação automática de texto (e.g., páginas web e citações bibliográficas) [Canuto et al., 2013; Carvalho et al., 2011; Davis et al., 2003; Hajishirzi et al., 2010].

Igualmente, em programas de governo, a deduplicação evita fraudes identificando cidadãos duplicados em programas sociais [Christen, 2012a]. Além disso, depois da expansão do comércio eletrônico e do desenvolvimento da computação em nuvem, a tarefa de deduplicação de dados não acontece mais apenas *offline* (i.e., re-processando toda a base de dados), porque ela pode ser executada em tempo real sobre uma consulta

[1] Encontrados na DBLP Computer Science Bibliography em 02/2018 - keywords: Deduplication, Record Linkage e Entity Resolution. URLs: http://dblp.uni-trier.de/search?q=Deduplication , http://dblp.uni-trier.de/search?q=Record%20Linkage , http://dblp.uni-trier.de/search?q=Entity%20Resolution

em uma biblioteca digital [Canalle et al., 2017; Vieira et al., 2016]. Outra abordagem mais atual é ter os registros duplicados identificados de forma incremental, ou seja, à medida em que são inseridos no banco de dados [Nascimento et al., 2018].

É importante notar que a tarefa de identificar duplicatas possui várias nomenclaturas, incluindo *Record Linkage*, *Entity Resolution*, *Object Reference*, *Duplicate Detection* e *Data Deduplication* [Christen, 2012a]. Quando considera apenas uma única fonte de dados é chamada de *Data Deduplication* ou *Duplicate Detection*. Entretanto, ela pode considerar várias fontes de dados, sendo referenciada por *Entity Resolution* ou *Record Linkage*. Nesse trabalho, nós adotamos o termo deduplicação de dados.

Independente da nomenclatura, o processo para identificar e eliminar registros duplicados é composto de três principais etapas: **indexação** que atribui uma chave para cada registro, **clusterização** que agrupa as instâncias de acordo com o valor da chave e **classificação** que classifica os registros pertencentes a cada grupo. Nosso foco é na etapa de *indexação*, que cria estruturas de Chave de Bloco (do termo em inglês *Block Key - BK*) para agrupar registros similares [Christen, 2012a]. A indexação é muito importante porque sem ela cada instância seria comparada com todas as outras, resultando em $(n * (n-1))/2$ comparações de pares de registros para um conjunto de dados com n instâncias. Em grandes conjuntos de dados, tal abordagem não é viável devido à sua complexidade quadrática [Evangelista et al., 2010].

Abordagens de deduplicação existentes usam BKs com diferentes propósitos, como ordenar e clusterizar os registros. Exemplos incluem *Standard Blocking* [Fellegi & Sunter, 1969], *Sorted Neighborhood* [Hernández & Stolfo, 1995], *Canopy* [McCallum et al., 2000] e *Adaptive Sorted Neighborhood* [Yan et al., 2007]. Porém, em todos os casos, o Valor de Chave de Bloco (do inglês *Block Key Value - BKV*) é definido de acordo com os atributos disponíveis nos conjuntos de dados. Idealmente, tais atributos de indexação devem ser eficazes (i.e., para melhor distinguir os registros) e eficientes (i.e., para permitir um rápido processo de deduplicação). Por exemplo, em uma avaliação com um conjunto de dados sintético contendo um milhão de registros, o melhor atributo de indexação executa o processo de deduplicação de dados em 15 horas, enquanto que o pior atributo requer mais de um dia de processamento (conforme discutido posteriormente no Capítulo 6).

Além da eficiência e eficácia, existem pelo menos dois requisitos extras para escolher um bom atributo de indexação [Christen, 2012a]: *qualidade*, para evitar valores em falta ou vazios (porque eles potencialmente criarão um enorme e inútil bloco com valores vazios); e *frequência*, para prover valores com boa distribuição (porque a distribuição influencia o número de pares candidatos que são gerados). Um par candidato são duas instâncias que têm atributos similares e podem corresponder à mesma enti-

Tabela 1.1: Identificando registros duplicados em duas fontes de dados

CD	ID	Fonte	Título	Artista	Categoria	Gênero	Extra	Ano
A	1	Atlantic Records	The Moment	Kenny G	Jazz	Jazz	ID3G:8	1996
	2	J Records	Moment	Kenny G	Null	Classical	Null	1996
B	3	Atlantic Records	Donde Hay Musica	Ramazzotti	Jazz	Ballad	Null	Null
C	4	Atlantic Records	Alien Ant Farm	Anthology	Rock	Rock	Null	Null
	5	J Records	Alien Ant Farm	Anthology	NewAge	AlternRock	ID3G:40	2001

dade no mundo real. A seguir, apresentamos um exemplo do processo de deduplicação de dados com ênfase na etapa de indexação para melhor justificar este trabalho, seguido por um sumário de soluções atuais, objetivos e principais contribuições.

1.1 Exemplo de Aplicação do Processo de Deduplicação de Dados

Considere a Tabela 1.1 que apresenta um exemplo simples de um *dataset* com cinco instâncias que são identificadas pela coluna *ID*. Tais instâncias se referem a três CDs de músicas identificados pela coluna *CD*. O problema é identificar os registros duplicados. Na etapa de indexação, as instâncias são indexadas por um atributo para evitar comparar todos os pares de registros entre si, 10 comparações neste caso. Considere também que a chave de bloco é formada pelo valor dos atributos. Então, dependendo do atributo de indexação selecionado, as seguintes situações podem ocorrer.

- Indexar por meio da coluna *Fonte*, são definidos dois blocos: um para os registros da fonte *Atlantic Records* e outro para *J Records*. No final, CDs 1 e 2, e 4 e 5 não são comparados porque estão em blocos distintos. Tal escolha prejudicaria a eficácia, pois nenhuma das duplicatas poderia ser encontrada.

- Indexar pela coluna *Título*, são definidos quatro blocos com os seguintes CDs: {1}, {2}, {3} e {4,5}. Essa escolha falha porque os registros 1 e 2 não são comparados, devido às mesmas razões da coluna *Fonte*.

- Indexar por meio da coluna *Ano*, são definidos dois blocos com os seguintes CDs: {1,2} e {5}. O processo então compara os CDs 1 e 2, mas falha porque não compara 4 e 5, visto que existem valores nulos.

- Indexar individualmente pelas colunas *Categoria*, *Gênero* e *Extra*, é igualmente ineficaz devido às mesmas razões. Além disso, diferentes fontes de dados podem ter distintas interpretações para categoria e gênero, como ilustrado no exemplo.

- Finalmente, indexar por meio da coluna *Artista*, o processo de deduplicação é mais eficaz, porque seriam comparados os CDs 1 e 2, bem como 4 e 5. O atributo *Artista* é o melhor atributo de indexação neste exemplo, pois 100% das duplicatas poderiam ser encontradas na classificação considerando um classificador perfeito.

Diante da problemática apresentada, observa-se a necessidade da criação de estratégias onde apenas os atributos mais relevantes sejam identificados e consequentemente utilizados na etapa de indexação. Um atributo é considerado relevante quando contribui de forma eficaz e eficiente para identificar o maior número de duplicatas verdadeiras, isto é, Verdadeiros Positivos (VP); e considerado irrelevante quando é ineficiente ou quando contribui de forma negativa, ou seja, identificando uma menor quantidade de correspondências positivas e, consequentemente, uma maior número de registros duplicados errados, isto é, Falsos Positivos (FP).

1.2 Soluções Atuais

Inicialmente, o processo de seleção de atributos para a etapa de indexação era puramente arbitrário, ou seja, sem nenhum critério [Hernández & Stolfo, 1995]. Atualmente, ele é baseado no conhecimento de usuários especialistas sobre o domínio dos dados [Christen, 2012b; Papadakis et al., 2015], i.e., na sua maior parte orientado ao domínio.

Existem pelo menos duas estratégias que utilizam atributos disponíveis para definir as BKs: *Schema-Agnostic*, que usa o valor de cada atributo para definir os valores de chave de bloco; e *Schema-Based Configurations*, que aplica um conjunto de regras nos valores extraídos dos atributos para definir as chaves como, por exemplo, o algoritmo *Soundex*, onde os valores são codificados baseados na pronúncia [Odell & Russell, 1918]. Um estudo comparativo entre tais métodos de indexação é encontrado em Papadakis et al. [2015]. No entanto, em ambas as estratégias de indexação, usuários especialistas são necessários para escolher manualmente os melhores atributos.

No quesito seleção de atributos, a maioria dos estudos têm focado em selecionar os melhores atributos para a etapa de classificação dos registros [Canalle et al., 2017; Chen et al., 2012; Fan et al., 2009; Su et al., 2010]. Nessa etapa, o objetivo é escolher os melhores atributos para classificar as duas instâncias. Soluções atuais incluem abordagens de aprendizagem de máquina [Chen et al., 2012] e métodos que utilizam informações dos dados, como densidade e repetição, para analisar os atributos [Canalle et al., 2017]. No entanto, não existe solução adaptada para a etapa de indexação.

1.3 Objetivos

O principal objetivo deste trabalho é desenvolver uma estratégia capaz de identificar automaticamente os atributos mais relevantes para a etapa de indexação do processo de deduplicação de dados. A estratégia proposta é baseada em critérios que buscam identificar a relevância individual dos atributos, ou seja, a estratégia avalia critérios relacionados aos dados (e.g., duplicidade e repetição dos valores dos atributos). Assim, os seguintes objetivos específicos são definidos para alcançar o objetivo principal:

- Avaliação do impacto do atributo sobre as etapas de indexação e clusterização do processo de deduplicação de dados;
- Definição de métricas e critérios para medir a relevância individual de cada atributo em um conjunto de dados;
- Aplicação das métricas em cada atributo para calcular sua relevância;
- Realização de experimentos contendo dados sintéticos e reais;
- Comparação dos resultados alcançados por meio da estratégia com *baselines*.

1.4 Contribuições

Nesta dissertação, nós propomos o *3DR-Indexing*[2]: um novo método para seleção de atributos orientado à indexação em deduplicação de dados. O método proposto é baseado em quatro métricas extraídas dos valores de cada atributo: **densidade**, **distintividade**, **duplicidade** e **repetição**. Suas principais vantagens e diferenças em relação às abordagens existentes (além de ser aplicado na etapa de indexação e não na de classificação) é que ele não usa algoritmos de aprendizagem de máquina nem necessita da intervenção de usuários especialistas. Portanto, pode ser adaptado para qualquer domínio porque não precisa de um conjunto de dados específico para treino.

Além disso, nós estudamos a importância da seleção de atributos sobre as etapas do processo de deduplicação e o desempenho do nosso método por meio de uma extensiva avaliação experimental. Especificamente, a avaliação experimental considera cinco questões:

- **Questão 1**. Qual parte possui maior impacto no processo de deduplicação de dados entre a função de indexação, o atributo de indexação e o algoritmo utilizado para clusterização dos registros?

[2]3DR = **D**ensidade + **D**uplicidade + **D**istintividade + **R**epetição

- **Questão 2**. O mesmo atributo de indexação é eficiente e eficaz nos métodos de indexação *Schema-Agnostic* e *Schema-Based Configurations*?
- **Questão 3**. Os resultados da deduplicação são eficazes utilizando o mesmo atributo de indexação nos algoritmos *Standard Blocking* e *Sorted Neighborhood*?
- **Questão 4**. O método *3DR-Indexing* encontra os melhores atributos para a etapa de indexação (baseado em características extraídas dos valores dos atributos)?
- **Questão 5**. Os resultados do processo de deduplicação de dados são melhorados com a utilização da combinação de atributos na etapa de indexação?

Nós avaliamos a eficiência e eficácia do atributo de indexação e do método proposto em 15 bases de dados contendo dados sintéticos e reais. Os experimentos consideram tanto critérios de escalabilidade, isto é, total de instâncias nos conjuntos de dados (10^2 até 10^6 instâncias), quanto o total de registros duplicados (10%, 30%, 50%, 70% e 90% de registros duplicados). Os *datasets* possuem vários domínios incluindo registros bibliográficos, CDs de música, informações sobre pessoas e dados de restaurante.

Os experimentos mostram que, em 10 de 13 conjuntos de dados, os resultados com melhor *trade-off* entre *F-Measure* e tempo de execução são conseguidos utilizando os atributos mais relevantes de acordo com o *3DR-Indexing* na etapa de indexação. Além disso, o melhor atributo de indexação difere do pior em aproximadamente 44% em termos de *F-Measure*, considerando a média de todos os conjuntos de dados. As implementações dos métodos e os conjuntos de dados utilizados nos experimentos estão publicamente disponíveis.[3]

1.5 Organização do Trabalho

O restante deste trabalho está organizado da seguinte maneira. O Capítulo 2 apresenta conceitos preliminares sobre o processo de deduplicação de dados, e o Capítulo 3 discute trabalhos relacionados. O Capítulo 4 introduz o método *3DR-Indexing*. A avaliação experimental está dividida em duas partes: Capítulo 5 com a metodologia, conjuntos de dados, métricas e configurações; e o Capítulo 6 com os resultados e discussões. Finalmente, concluímos o trabalho e apontamos trabalhos futuros no Capítulo 7.

[3]http://www.dcc.ufmg.br/~mirella/projs/deduplica

Capítulo 2

Processo de Deduplicação de Dados

Este capítulo apresenta os conceitos necessários para entendimento do método proposto e da avaliação experimental. A Seção 2.1 apresenta uma visão geral sobre deduplicação de dados, e as etapas do processo são detalhadas nas seções seguintes.

2.1 Visão Geral

O processo de deduplicação de dados pode ser dividido em várias etapas, incluindo *Indexação*, *Clusterização*, *Classificação* e *Avaliação* [Christen, 2012a]. Assim, considere D um conjunto de dados contendo i registros, tal que $D = \{r_1, r_2, .., r_i\}$. Cada registro r é definido por um conjunto de atributos $A_r = \{a_1, a_2, .., a_j\}$. Nesse trabalho, nós consideramos registros, tuplas e instâncias como sinônimos. Então, o processo de deduplicação acontece conforme ilustrado na Figura 2.1 e nas etapas a seguir.

1. Todos os registros r são indexados por um valor de chave de bloco. Para tal, atributos de A_r são escolhidos e seus valores são utilizados como chave (*Schema-Agnostic*) ou uma regra de codificação é aplicada no valor dos atributos para gerar a chave (*Schema-Based Configurations*). No fim da etapa, com exceção das instâncias que contêm valores nulos nos atributos, cada registro $r \in D$ está associado a uma BK.

2. Cada BK define grupos de registros que podem corresponder ao mesmo objeto no mundo real. Assim, todos os registros que contêm o mesmo valor de BK são agrupados por meio de abordagens baseadas em blocos ou vizinhos, por exemplo.

3. Os registros pertencentes a cada grupo são comparados com todos os outros do mesmo grupo, par a par, e classificados como *duplicados*, *não duplicados* e *possí-*

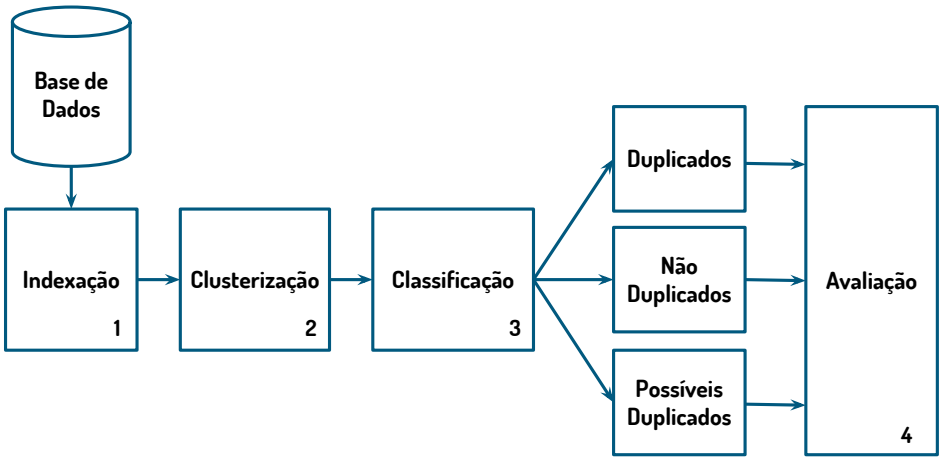


Figura 2.1: Processo de deduplicação de dados (adaptado de [Christen, 2012a])

veis duplicados utilizando critérios de similaridade entre os valores dos atributos dos registros comparados. A terceira classe necessita de uma avaliação mais detalhada, geralmente feita por um usuário especialista no domínio dos dados, porque os registros possuem alguns atributos diferentes mas parecem similares.

4. O processo de deduplicação de dados é avaliado para medir a eficácia dos algoritmos utilizados em cada uma das etapas. Métricas típicas de avaliação são Revocação, Precisão e *F-Measure* [Hand & Christen, 2018].

A seguir, as Seções 2.2, 2.3 e 2.4 apresentam alguns dos algoritmos utilizados em cada uma das etapas. Especificamente, depois de avaliar o estado da arte e as conclusões apresentadas por Silva et al. [2017], nós escolhemos os algoritmos mais eficientes e eficazes para cada uma das etapas, incluindo: codificação *Soundex* (etapa de indexação) [Odell & Russell, 1918], agrupamento por blocos e vizinhos (etapa de clusterização dos registros) [Fellegi & Sunter, 1969; Hernández & Stolfo, 1995], e o algoritmo de *Jaro Winkler* (etapa de classificação) [Winkler, 1990]. Segundo Christen [2012a], esses algoritmos têm sido utilizados por décadas.

2.2 Etapa de Indexação

Conforme apresentado na Figura 2.1, a indexação é composta da escolha de um atributo e da codificação do seu valor para produzir o valor da chave de bloco. Existem diversas técnicas para codificar os valores dos atributos [Christen, 2012a]. Entretanto, uma

Tabela 2.1: Codificação *Soundex* (adaptado de [Christen, 2012a])

Caractere	**Dígito**
a, e, i, o, u, y, h, w	0
b, f, p, v	1
c, g, j, k, q, s, x, z	2
d, t	3
l	4
m, n	5
r	6

Tabela 2.2: Exemplo de codificação *Soundex* e *Suffix*

#	**Gênero**	**Codificação *Soundex***	**Codificação *Suffix* ($K = 3$)**
1	Jazz	J200	JAZ
2	Classical	C422	CLA
3	Ballad	B430	BAL
4	Rock	R200	ROC
5	AlterRockn	A436	ALT

das principais é a codificação *Soundex* [Christen, 2012a], desenvolvida e patenteada por Odell & Russell [1918] e baseada na pronúncia da língua inglesa. A codificação *Soundex* pode ser adaptada para a língua portuguesa, porque a língua inglesa e a portuguesa partilham do mesmo alfabeto. Além disso, segundo Silva et al. [2017], os resultados da deduplicação utilizando a codificação *Soundex* sobre conjuntos de dados em português são significantes em termos de *F-Measure* por exemplo.

A técnica *Soundex* codifica o valor de um atributo mantendo o primeiro caractere e converte os demais para dígitos entre zero e seis, de acordo com o som (conforme Tabela 2.1). A codificação acontece da seguinte maneira: (i) com exceção do primeiro, todos os caracteres são codificados para dígitos; (ii) os dígitos 0 são removidos porque não influenciam na fonética da palavra; (iii) os dígitos duplicados são removidos, ou seja, apenas uma instância de cada número é mantida na sequência; (iv) a codificação deve conter quatro caracteres, sendo uma letra e três números; e (v) dígitos zeros são acrescidos ao final, caso a sequência não tenha tamanho quatro.

Outra opção de indexação é a função *Suffix*, que cria sufixos de tamanho K a partir de um valor. Via de regra, o valor de K é configurado manualmente. A Tabela 2.2 apresenta um exemplo de codificação *Soundex* e *Suffix* considerando a coluna *Gênero* do exemplo simples da Tabela 1.1. Uma avaliação experimental entre tais funções de indexação é encontrada em Silva et al. [2017].

2.3 Etapa de Clusterização

Depois da etapa de indexação, os registros são agrupados considerando o valor de cada chave de bloco. O principal objetivo da clusterização é reduzir a complexidade quadrática da deduplicação, comparando apenas os possíveis pares de duplicatas. Existem diferentes abordagem para esta etapa, entretanto os principais métodos utilizam conceitos de comparação por blocos e vizinhos mais próximos, como segue.

O algoritmo ***Standard Blocking*** cria um conjunto de blocos, onde cada bloco agrupa registros similares. Por este motivo, os registros são comparados entre eles apenas dentro de cada bloco (o que simplifica a complexidade quadrática original). Os grupos são definidos de acordo com o valor de BK gerado para os registros. Essa técnica é considerada livre de redundância, pois produz blocos distintos e cada registro está associado a apenas um bloco. Por exemplo, voltando ao exemplo da Tabela 1.1, se os registros 1, 2, 3, 4 e 5 são indexados por meio da coluna *Artista* aplicando a codificação *Soundex*, três blocos são criados: **K520** com os registros 1 e 2; **R626** com 3; e **A534** com os registros 4 e 5. Logo, apenas o registro 1 é comparado com o 2, bem como o 4 com o 5.

Diferente da técnica de blocagem, o algoritmo ***Sorted Neighborhood***, criado por Hernández & Stolfo [1995], combina os registros por meio de uma chave ordenada, que é similar a uma BK. Porém, antes de executar as comparações, todos os registros são ordenados de acordo com o valor de cada BK. Depois, uma janela deslizante de tamanho $w > 1$ percorre todos os registros de D, e o primeiro registro da janela é comparado com todos os outros dentro da mesma janela. Neste caso, o método de vizinhança não é livre de redundância, porque produz blocos sobrepostos, ou seja, um registro está associado a mais de um bloco. No entanto, um mesmo registro não é comparado mais de uma vez com outro.

A Figura 2.2 apresenta um exemplo do algoritmo de vizinhança. Considerando a janela corrente, $w1$, o registro 1 é comparado com os registros de 2 até 8. Na próxima iteração do algoritmo, janela $w2$, ele compara o registro 2 com os registros de 3 a 8. As próximas iterações seguem o mesmo procedimento, até o último registro do conjunto de dados. Uma versão melhorada do algoritmo de vizinhança com janela dinâmica é encontrada em Yan et al. [2007]. Por fim, avaliações experimentais entre os métodos de blocagem e janela deslizante são encontrados nos trabalhos de Baxter et al. [2003], Christen [2012b], Draisbach & Naumann [2009] e Silva et al. [2017].

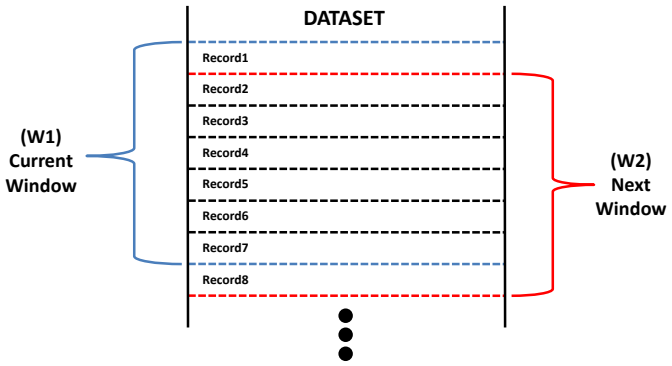


Figura 2.2: Exemplo do algoritmo *Sorted Neighborhood* (adaptado de [Yan et al., 2007])

2.4 Etapa de Classificação

Existem muitas maneiras de avaliar o quanto dois valores são similares. Via de regra, uma função de similaridade calcula a correspondência entre dois valores e retorna um valor no intervalo $[0, 1]$, no qual 1 representa o "par perfeito" (do inglês *perfect match*).

O algoritmo de ***Jaro Winkler*** é uma opção. Ele é uma extensão do algoritmo *Jaro* e foi proposto por Winkler [1990]. O tamanho das *strings* e os erros que comumente ocorrem com valores alfanuméricos são considerados para calcular a similaridade entre os valores. Além disso, no algoritmo de *Winkler* existe um valor $(0 \leq p \leq 4)$ que representa o número de elementos em comum nos quatro primeiros caracteres dos valores comparados, o que aumenta a similaridade para valores pequenos. Assim, a Equação 2.1 calcula a similaridade *Jaro*.

$$S_{jaro}(X, Y) = \frac{1}{3}\left(\frac{c}{|X|} + \frac{c}{|Y|} + \frac{c-t}{c}\right), \tag{2.1}$$

onde X e Y são os valores comparados, c o número de caracteres em comum, $|X|$ e $|Y|$ o tamanho das strings X e Y, e t o número de tranposições para alcançar a igualdade entre dois caracteres. Em seguida, a Equação 2.2 calcula a similaridade de *Winkler*.

$$S_{winkler}(X, Y) = S_{jaro}(X, Y) + (1 - S_{jaro}(X, Y))\frac{p}{10} \tag{2.2}$$

No fim dessa etapa, os registros são classificados como *duplicados*, *não duplicados* e *possíveis duplicados* de acordo com um limiar de similaridade aplicado nos valores dos atributos. Para classificá-los, todos os atributos disponíveis podem ser utilizados (e.g., *Fonte*, *Título*, *Artista*, *Categoria*, *Gênero*, *Extra*, *Ano*) ou apenas os atributos mais descritivos de cada conjunto de dados (e.g., *Título* e *Artista*) [Canalle et al., 2017]. Por fim, avaliações experimentais com funções de classificação são encontradas nos trabalhos apresentados por Christen [2006], Cohen et al. [2003] e Silva et al. [2017].

2.5 Considerações Finais

Este capítulo detalhou o processo de deduplicação de dados e cada uma das suas etapas. Além disso, os algoritmos de deduplicação adotados para a avaliação experimental foram apresentados. Finalmente, outros estudos comparativos contento algoritmos de deduplicação também foram citados.

O processo de deduplicação de dados apresentado neste capítulo é utilizado em todas as avaliações experimentais, tanto na avaliação do impacto da escolha do atributo de indexação sobre o processo de deduplicação, quanto na validação do método proposto *3DR-Indexing*. A seguir, o Capítulo 3 contém trabalhos relacionados separados de acordo com cada uma das etapas da deduplicação discutidas. Além disso, o capítulo posterior contém estudos que modelam o problema da deduplicação de maneira diferente ao processo apresentado neste capítulo.

Capítulo 3

Trabalhos Relacionados

Estudos de deduplicação são divididos em categorias da seguinte forma: estudos que analisam funções de indexação e definição de chaves de bloco (etapa de indexação) [Papadakis et al., 2015]; estudos que avaliam e desenvolvem algoritmos para clusterização dos registros (etapa de clusterização) [Fellegi & Sunter, 1969; Hernández & Stolfo, 1995; McCallum et al., 2000; Yan et al., 2007]; estudos que analisam funções para computar a semelhança entre dois valores (etapa de classificação) [Christen, 2006; Cohen et al., 2003]; estudos que propõem métodos de deduplicação específicos para determinados domínios de dados como, por exemplo, bibliotecas digitais e dispositivos móveis [Borges et al., 2017; Canalle et al., 2017; Canuto et al., 2013; Carvalho et al., 2015; Levin & Heuser, 2010; Vieira et al., 2016]; e estudos que realizam avaliações experimentais com os métodos existentes [Christen, 2012b].

Este capítulo sumariza trabalhos relacionados em diferentes aspectos e etapas do processo de deduplicação de dados. Os estudos são selecionados de acordo com os seguintes critérios: (i) estudos que apresentem soluções e avaliações experimentais para as etapas do processo de deduplicação; (ii) trabalhos que desenvolvam métodos de deduplicação para domínios de dados específicos; e (iii) estudos que analisem o problema da seleção de atributos relevantes na deduplicação de dados. As bibliotecas digitais *DBLP*[4] e *Google Acadêmico*[5] foram utilizadas para selecionar os estudos publicados até 05/2018. A busca é realizada com as palavras-chave *Deduplication*, *Data Deduplication*, *Record Linkage* e *Entity Resolution.* Além disso, título e resumo de cada trabalho foram analisados para selecionar os mais relevantes.

A seguir, a Seção 3.1 apresenta um *survey* com as principais metodologias utilizadas na identificação de duplicatas em um contexto de *Entity Resolution* e *Record*

[4]DBLP: http://dblp.uni-trier.de/
[5]Google Acadêmico: https://scholar.google.com.br/

Linkage. Os trabalhos de cada uma das etapas da deduplicação são apresentados nas Seções 3.2 e 3.3 respectivamente. A Seção 3.4 aborda trabalhos de seleção de atributos para a deduplicação. Trabalhos orientados ao domínio dos dados são apresentados na Seção 3.5. Por fim, a Seção 3.6 compara os trabalhos existentes com o nosso estudo.

3.1 Uma Visão Geral sobre Deduplicação de Dados

Deduplicação de dados é a tarefa de encontrar registros duplicados em bancos de dados. Como mencionado no Capítulo 1, esta tarefa é referenciada na literatura por diversas nomenclaturas incluindo *Record Linkage* e *Entity Resolution*. Neste contexto, Brizan & Tansel [2006] apresentam um *survey* de metodologias propostas para os processos de *Record Linkage* e *Entity Resolution*. Segundo os autores, o propósito da *Entity Resolution* é identificar uma representação única em um processo de relação entre duas instâncias e assim definir a instância que representa de fato a entidade. Por outro lado, *Record Linkage* tem o objetivo de explicitar *links* entre relacionamentos de tuplas que representam o mesmo objeto do mundo real.

Segundo os autores, os processos para execução de *Record Linkage* e *Entity Resolution* são divididos em duas categorias: (i) processos que definem um bom critério de casamento (*matching*) para qualquer par de tuplas; e (ii) processos que definem um critério de casamento específico sobre um ou mais relacionamentos entre instâncias. Esses critérios podem contemplar o casamento exato dos registros ou ainda uma distância de similaridade mínima entre as tuplas.

Além disso, diversas abordagens são propostas para o quesito desempenho dos algoritmos, incluindo *Brute Force*, *Sliding Window* e *Bucketing*. Com a abordagem *Brute Force*, cada tupla da base de dados é comparada com todas as outras. A metodologia *Sliding Window* ordena as tuplas de acordo com algum critério (por exemplo, o nome) e uma janela fixa de tamanho w desliza sobre os dados. Enquanto isso ocorre, os registros pertinentes à cada janela são comparados com todos os outros da mesma janela. Por outro lado, a técnica *Bucketing* define um critério de clusterização para os registros e forma vários blocos, e geralmente os blocos são criados de acordo com cada valor de BK distinto. Desse modo, apenas os registros presentes em cada um dos blocos são comparados com os demais do mesmo bloco. Por fim, para avaliar os algoritmos de deduplicação, métricas típicas de Precisão, Revocação e *F-Measure* são utilizadas. Atualmente, tais métricas ainda são amplamente adotadas [Hand & Christen, 2018].

3.2 Indexação e Clusterização dos Registros

No processo de deduplicação de dados, cada etapa produz entradas para a etapa seguinte. A indexação e a clusterização são etapas complementares pois elas têm o objetivo de criar grupos de registros que possivelmente possam ser similares e, após isso, classificá-los na etapa de classificação. O agrupamento é visto como um processo de clusterização, onde um conjunto de entidades é dividido em vários *clusters*, e os membros de cada *cluster* são similares de alguma forma [Zaït & Messatfa, 1997]. Clusterização é uma técnica bem conhecida na mineração de dados e útil em várias aplicações incluindo: análise de padrões, segmentação de imagens, recuperação de informação e deduplicação de dados [Nascimento et al., 2018]. Para clusterizar as entidades, cada registro $r \in D$ recebe uma chave identificadora, conhecida por *chave de bloco*.

Por exemplo, quando a deduplicação é executada em duas bases de dados, cada registro de uma base de dados é comparado com todos os demais registros da outra base de dados. Quando esta tarefa acontece em uma única base de dados, cada registro é comparado com todos os outros registros da mesma base de dados. A complexidade da tarefa de identificar registros duplicados cresce de forma quadrática, de acordo com o tamanho do conjunto de dados a ser verificado [Christen, 2012a]. Desse modo, indexar os registros é uma ótima estratégia para melhorar o desempenho dos algoritmos aplicados ao problema da deduplicação.

Considerando a etapa de **indexação**, existem pelo menos duas estratégias de indexação para definição de chaves de bloco: *Schema-Based Configuration*, que aplica um conjunto de regras nos valores dos atributos para definir as chaves (e.g., codificação *Soundex*); e *Schema-Agnostic*, que trata cada valor do atributo como um valor de chave de bloco. Nesse contexto, Papadakis et al. [2015] apresentam um estudo comparativo entre tais métodos. Os autores realizam uma avaliação experimental das duas técnicas combinadas com nove métodos de blocagem do estado da arte. Os experimentos são executados em diversos conjuntos de dados com tamanho variando entre 10.000 e 2.000.000 de registros. Os autores concluem que a técnica *Schema-Agnostic* oferece maior robustez para definição das chaves porque é não supervisionada e independente do domínio dos dados.

Na segunda etapa do processo, **clusterização dos registros**, há uma maior quantidade de trabalhos publicados. Existem estudos que propõem novos métodos e outros que realizam avaliações experimentais com os métodos já existentes. Nessa etapa, o objetivo é desenvolver algoritmos para clusterizar os registros de acordo com os valores de chave de bloco ou regras específicas. Tais algoritmos buscam reduzir a complexidade quadrática do problema da deduplicação de dados.

Considerando os estudos de avaliações experimentais, Baxter et al. [2003] comparam *Bigram Indexing* e *Canopy Clustering* com as abordagens tradicionais de bloco e vizinhos (i.e., *Standard Blocking* e *Sorted Neighborhood* algoritmo). Segundo os autores, *Bigram Indexing* provê um processo de deduplicação mais eficiente e melhora a acurácia. Draisbach & Naumann [2009] comparam os métodos de blocagem com os algoritmos de janela deslizante (e.g, *Sorted Neighborhood*). Os experimentos demostram que os métodos de janela são melhores que os métodos de bloco no quesito eficiência. Por fim, um estudo completo sobre os métodos de clusterização de registros é abordado por Christen [2012b]. O autor apresenta um *survey* de 12 variações de seis técnicas de agrupamento existentes. Além disso, o desempenho das técnicas é avaliado por meio de uma avaliação experimental. No estudo, os métodos analisados são: *Standard Blocking, Sorted Neighborhood, Suffix Array-Based Indexing, Canopy Clustering* e *String-Map-Based Indexing*. Resultados mostram que a técnica *Q-gram-Based Indexing* é uma das mais lentas e não é adaptável para grandes *datasets*. Por outro lado, as abordagens tradicionais são as mais rápidas (i.e., blocos e vizinhos).

Além dos estudos de avaliações experimentais, existem estudos que desenvolvem novos métodos. Por exemplo, Draisbach et al. [2012] apresentam uma adaptação do algoritmo *Sorted Neighborhood* com tamanho de janela variado. Na proposta, o tamanho da janela que desliza sobre os dados sofre alterações dinamicamente, dependendo da região do *dataset*. Os autores acreditam que há regiões com alto nível de similaridade e regiões com baixo nível de similaridade. Por este motivo, o tamanho da janela muda de acordo com o total de registros duplicados já encontrados na iteração atual do algoritmo. Quanto maior o número de duplicatas encontradas, maior o tamanho da janela para a próxima iteração. Por outro lado, quanto menor o total de duplicatas encontradas, menor o tamanho da janela. Segundo os autores, quando não existem registros duplicados nas proximidades da janela atual, a janela pode ser reduzida na próxima iteração. Os autores concluem que o método proposto consegue melhorar a eficiência do processo de deduplicação de dados quando comparado com o método *Sorted Neighborhood*, além de manter a eficácia do processo.

Igualmente, Evangelista et al. [2010] apresentam um novo método para blocagem de registros utilizando aprendizagem de máquina e programação genética. Na abordagem proposta, os blocos são representados como árvores e modelados como sequências de predicados e operações booleanas. Os predicados são formados por um conjunto de atributos de cada registro. Assim, dois registros estão no mesmo bloco e são considerados possíveis duplicados se apresentarem predicados em comum. Os autores comparam o método proposto com métodos do estado da arte e concluem que a metodologia é eficiente e pode lidar com um grande número de blocos de predicados. Segundo os

autores, usar um grande número de predicados é uma vantagem, porque o processo de blocagem é executado com maior flexibilidade e a identificação dos registros duplicados tem mais chances de sucesso.

Do mesmo modo, Bilenko et al. [2006] propõem uma metodologia para blocagem de registros adaptativa utilizando algoritmos de aprendizagem de máquina. Nessa abordagem, as funções de blocagens são selecionadas de um conjunto de predicados que são extraídos dos dados. O principal objetivo do trabalho é encontrar a melhor combinação de predicados que são capazes de capturar o maior número de verdadeiros positivos com uma menor quantidade de falsos positivos. Os predicados extraídos dos dados incluem: (i) *Contain Common Token*, que é responsável por dividir uma *string* em um conjunto de *tokens*, e (ii) *Three Chars*, onde os três primeiros caracteres de uma *string* são utilizados. No geral, os resultados demonstram que as técnicas de blocagem adaptativas melhoram a eficiência e eficácia do processo de deduplicação.

Além desses, McCallum et al. [2000] apresentam uma nova técnica para clusterização de referências bibliográficas aplicada ao contexto de *Record Linkage*. O principal objetivo do trabalho é utilizar métricas que calculam a distância dos registros baseando-se no valor de similaridade e, a partir disso, produzir a clusterização dos registros. Segundo os autores, os resultados demonstram que a função de similaridade *Edit Distance* apresenta os melhores resultados para clusterização.

No geral, a maioria dos estudos citados desenvolvem novos métodos e realizam avaliações experimentais com os métodos existentes. No entanto, existem estudos que melhoram a qualidade dos blocos criados por meio das técnicas tradicionais em termos de eficácia e eficiência [Papadakis et al., 2011, 2014, 2016]. Esse processo é conhecido como *Meta-Blocagem*. O objetivo é remover comparações supérfluas e redundantes de qualquer bloco criado com os algoritmos tradicionais. Uma comparação supérflua é formada por um par de registros que não são cópias (e.g., falsos positivos). Para tal, o conjunto de blocos originais é transformado em um Grafo de Blocagem, que contém um vértice para cada instância resultante da blocagem inicial e uma aresta que liga cada par de tuplas que co-ocorrem em pelo menos um bloco. Em seguida, as técnicas de meta-blocagem buscam descartar arestas do Grafo de Blocagem utilizando um algoritmo de poda, por exemplo. Com isso, comparações supérfluas são descartadas, melhorando a eficiência da deduplicação. Quatro técnicas de poda são encontradas no trabalho de Papadakis et al. [2014]. Por fim, trabalhos atuais propõem heurísticas que encontram as duplicatas à medida em que elas vão sendo inseridas na base de dados, processo conhecido como deduplicação de dados incremental [Nascimento et al., 2018].

3.3 Classificação dos Registros

Depois da clusterização, os registros são avaliados na etapa de classificação, par a par. Nessa etapa, o objetivo é analisar os valores dos atributos de cada instância para determinar se os registros são cópias. Para tal, funções de similaridade são utilizadas para computar a semelhança entre dois registros. Na etapa de classificação, existe uma menor quantidade de trabalhos publicados. Porém, há estudos que comparam diversas funções de classificação, as quais são úteis para deduplicação em vários domínios.

Inicialmente, Cohen et al. [2003] comparam várias funções de similaridade utilizando registros de nomes pessoais. Algumas das funções analisadas são: *TFIDF, SoftTFIDF, Levenshtein, Scaled Levenstein, Jaro, Jaro-Winkler, Jaccard* e *NaiveAvgOverlap*. Os resultados mostram que o melhor método para comparação de nomes pessoais é uma versão escalada do algoritmo de *Levenshtein* ou *Edit-Distance*.

Em seguida, Christen [2006] compara 20 funções de similaridade incluindo: *Soundex, Phonex, phonix, Jaro, Winkler* e *Edit Distance*. A avaliação experimental utiliza quatro *datasets* contendo nomes pessoais. Segundo os autores, a melhor função de classificação não é clara. Entretanto, a técnica *Simple Phonex* tem desempenho melhor que as técnicas *Complex Phonix* e *Double-Metaphone*. Além disso, os algoritmos de *Jaro* e *Jaro Winkler* são eficazes em todos os conjuntos de dados utilizados.

3.4 Seleção de Atributos Relevantes para a Deduplicação de Dados

Cada registro é formado por um conjunto de atributos, os quais são utilizados tanto na etapa de indexação para formar os blocos, quanto na etapa de classificação para determinar se duas entidades são cópias. Desse modo, o objetivo da seleção de atributos relevantes é encontrar os melhores atributos que podem ser utilizados em ambas as etapas de modo a melhorar a eficiência e a eficácia do processo. Então, uma tarefa fundamental é definir um conjunto de métricas que avaliem a importância de cada atributo no seu conjunto de dados. Além disso, busca-se mensurar se a eficácia do processo como um todo é superior àquela alcançada quando os atributos são escolhidos sem nenhum critério, e até mesmo quando todos os atributos são aplicados.

Nesse contexto, Canalle et al. [2017] apresentam uma abordagem de seleção de atributos relevantes para a etapa de classificação em um processo de integração de dados de bibliotecas digitais. Critérios são extraídos dos valores de cada atributo para computar a relevância dos atributos. Os critérios incluem métricas de repetição e

densidade nos valores dos atributos e métricas relacionadas à qualidade das fontes dos dados, por exemplo, cobertura e confiabilidade.

O principal diferencial apresentado por Canalle et al. [2017] é não utilizar algoritmos de aprendizagem de máquina, visto que não é tarefa trivial obter os dados para o processo de treinamento dos algoritmos. Diversos experimentos são executados para identificar se a quantidade de atributos utilizados na etapa de classificação é totalmente relacionada com a eficácia dos resultados. Segundo os autores, com a utilização da abordagem proposta é possível alcançar resultados eficazes na Resolução de Entidades, e o método proposto identifica atributos relevantes para a classificação.

De maneira similar, Chen et al. [2012] propõem um método baseado em aprendizagem de máquina para selecionar o atributo ou o grupo de atributos mais relevantes para a etapa de classificação. Além disso, o método proposto define automaticamente o limiar e os pesos adequados para cada função de classificação. Os autores utilizam as funções de similaridade *Jaccard Distance*, *Jaro Distance* e *Edit Distance* para comparar as instâncias. Os experimentos são executados na base de dados do *Cora*[6]. Os resultados mostram que o método proposto é eficiente e eficaz.

Por fim, Su et al. [2010] apresentam um método para identificação de registros duplicados em um cenário de consultas web considerando múltiplas fontes de dados. O principal objetivo do trabalho é ajustar o limiar para cálculo da similaridade entre duas instâncias. Os autores afirmam que é necessário atribuir diferentes pesos para cada um dos atributos de acordo com sua importância. Diversos experimentos são realizados em um conjunto de cinco bases de dados para avaliar a abordagem proposta. Os domínios dos conjuntos de dados incluem livros, filmes e hotéis. Os autores concluem que os resultados do método proposto é comparável com trabalhos anteriores.

3.5 Deduplicação de dados Orientada ao Domínio

Além dos estudos específicos para as etapas da deduplicação, outros estudos desenvolvem soluções para um domínio de dados restrito, incluindo bibliotecas digitais [Canalle et al., 2017], desambiguação de nomes [Carvalho et al., 2011; Levin & Heuser, 2010; Momeni & Mayr, 2016; Müller, 2017], dispositivos móveis [Borges et al., 2017], dados de profissionais de saúde [Carvalho et al., 2015] e dados sobre registros de homicídios [Sadinle & Fienberg, 2013].

Vários estudos são publicados no contexto de bibliotecas digitais. Via de regra, bibliotecas digitais são compostas por coleções de objetos digitais como, por exemplo,

[6]http://www.cs.umass.edu/~mccallum/data/cora-refs.tar.gz

documentos, imagens, mapas, etc., e oferecem serviços aos seus usuários de pesquisa e publicação desses objetos [Fox et al., 1995]. Um dos problemas existentes nessas bibliotecas é a presença de objetos duplicados devido à diversidade de fontes distintas que são integradas ao mesmo tempo.

Nesse contexto, Borges et al. [2008] apresentam uma abordagem para identificação de registros duplicados em bibliotecas digitais. No trabalho, funções de similaridade são especificadas incluindo *YearSim*, *IniSim*, *NameMatch* e *MetadataMatch*. Segundo os autores, essas funções identificam com maior eficácia os registros duplicados em um conjunto de metadados de objetos digitais e podem ser utilizadas para a etapa de classificação, porque apresentam ganhos significativos de até $31,5\%$ quando comparadas com outras propostas de classificação.

Igualmente, Levin & Heuser [2010] propõem funções de análise de redes sociais que, somadas à métricas de similaridade, melhoram o processo de deduplicação no contexto de desambiguação de nomes em bibliotecas digitais. No estudo, os autores modelam a relação entre os pesquisadores como um grafo de artigos e, em seguida, aplicam métricas no grafo para realizar a desambiguação. As métricas são: de distância, de relacionamento, e de força do relacionamento entre os nós. No final, os algoritmos utilizam a combinação das métricas para determinar se dois nomes representam a mesma pessoa, ou seja, são ambíguos. Os autores concluem que a eficácia do processo de desambiguação de nomes em bibliotecas digitais é melhorada com a utilização de métricas de relacionamento entre as publicações.

Do mesmo modo, Borges et al. [2011] apresentam uma abordagem que combina funções de similaridade e algoritmos de aprendizagem de máquina para identificar registros duplicados no mesmo contexto. A principal diferença do trabalho é que o limiar de similaridade é produzido automaticamente pelos algoritmos de aprendizagem de máquina, ao invés de ser configurado manualmente de acordo com cada função de similaridade. Diversos classificadores são utilizados nos experimentos incluindo árvores de decisão e teorema de *Bayes*. Segundo os autores, a proposta identifica corretamente 98.8% dos registros duplicados nas bases de dados.

Além desses, Carvalho et al. [2012] propõem um método baseado em programação genética para identificação de registros duplicados em um contexto de *Record Deduplication*. A abordagem utiliza evidências extraídas dos dados para prover uma função que é capaz de identificar se dois ou mais registros correspondem a uma mesma entidade. Cada evidência (E) é composta por um atributo e uma função de similaridade específica para aquele atributo, ou seja, $E = \{attribute, similarityfunction\}$. Segundo os autores, a metodologia é avaliada em um contexto de bibliotecas digitais e tem resultados eficazes.

Diferente dos trabalhos apresentados em Borges et al. [2008], Borges et al. [2011], Carvalho et al. [2012] e Levin & Heuser [2010], existem trabalhos que detectam as cópias à medida em que novas instâncias vão sendo inseridas nas bibliotecas digitais. Por exemplo, Carvalho et al. [2011] apresentam um novo algoritmo não supervisionado para desambiguação de nomes em registros de conferências bibliográficas de forma incremental. O diferencial é não ter que re-processar todo o conjunto de dados, o que acontece com a maioria dos métodos estáticos apresentados.

Finalmente, existem trabalhos que abordam o problema da deduplicação de dados em outros contextos [Borges et al., 2017; Carvalho et al., 2015; Sadinle & Fienberg, 2013]. Por exemplo, Borges et al. [2017] apresentam um novo método para deduplicação de contatos em dispositivos móveis utilizando similaridade textual e aprendizagem de máquina. Segundo os autores, o método proposto identifica 92% de contatos duplicados nos dispositivos móveis, mesmo quando eles não contêm números de telefones idênticos.

3.6 Discussões

Esta seção apresenta uma visão geral dos trabalhos relacionados e os discute frente ao que é proposto nesta dissertação. As Tabelas 3.1, 3.2 e 3.3 detalham as principais contribuições de cada um dos trabalhos relacionados considerando as etapas do processo de deduplicação de dados, as avaliações experimentais e o domínio dos dados respectivamente. Além disso, a contribuição desta dissertação é exibida ao final de cada uma das tabelas. Assim, as principais diferenças entre os trabalhos relacionados e este trabalho são esclarecidas.

Em resumo, os trabalhos da Tabela 3.1 Canalle et al. [2017], Chen et al. [2012] e Su et al. [2010] encontram atributos relevantes para a etapa de classificação. A proposta desta dissertação é diferente pois visa propor métricas para identificação de atributos relevantes, porém com foco na etapa de indexação, o que não é contemplado pelos trabalhos citados. Considerando os trabalhos de Carvalho et al. [2011], Draisbach et al. [2012] e Levin & Heuser [2010] que focam na etapa de clusterização, nosso estudo difere pois foca na indexação. Tendo em vista os trabalhos de Bilenko et al. [2006], Evangelista et al. [2010] e McCallum et al. [2000] que propõem métricas para a etapa de indexação, nosso estudo é diferente porque utiliza outras métricas para indexar os registros (e.g., densidade e repetição), em vez de predicados e funções de similaridade. Os trabalhos de Bilenko et al. [2006] e Evangelista et al. [2010] utilizam algoritmos de aprendizagem de máquina para definir os predicados, e nosso estudo não considera tais algoritmos. Ademais, nos trabalhos de Draisbach et al. [2012] e McCallum et al.

[2000], o objetivo é propor um novo método para criação e reorganização dos blocos, enquanto nosso foco é propor um método para seleção de atributos relevantes.

Considerando a Tabela 3.2, nos trabalhos apresentados por Baxter et al. [2003], Christen [2006], Christen [2012b], Cohen et al. [2003], Draisbach & Naumann [2009] e Papadakis et al. [2015], o objetivo é realizar avaliações experimentais com métodos já existentes. Por exemplo, Papadakis et al. [2015] comparam o desempenho de dois métodos para definição de chaves (*Agnostic* e *Based Configurations*). Nosso trabalho difere porque, além de propor um novo método para seleção de atributos de indexação, nós analisamos a eficiência e eficácia do atributo de indexação utilizando os dois mé-

Tabela 3.1: Visão geral dos trabalhos relacionados por etapa da deduplicação

Trabalhos	**Principal Contribuição**	**Etapa**	**Métricas**	**Aprendizagem de Máquina**
Evangelista et al. [2010]	Abordagem de indexação utilizando sequência de predicados e operações booleanas	Indexação	Predicados	SIM
Bilenko et al. [2006]	Definição do melhor conjunto de predicados extraídos dos dados para criação dos blocos	Indexação	Predicados	SIM
McCallum et al. [2000]	Criação de grupos utilizando distância de similaridade entre os registros da base de dados	Indexação	Funções de similaridade	NÃO
Draisbach et al. [2012]	Proposta de janela dinâmica para o algoritmo *Sorted Neighborhood*	Clusterização	Tamanho da Janela W	NÃO
Carvalho et al. [2011]	Detecção de registros duplicados sem processar todo o banco de dados	Clusterização	Heurísticas	NÃO
Levin & Heuser [2010]	Análise de funções de redes sociais para desambiguação de nomes em bibliotecas digitais	Clusterização	Distância, relacionamento e força	NÃO
Canalle et al. [2017]	Métricas para identificar atributos mais relevantes a serem utilizados na etapa de comparação dos registros	Classificação	Repetição, Densidade, Cobertura	NÃO
Chen et al. [2012]	Abordagem para identificar atributos mais importantes a serem utilizados na etapa de comparação dos registros	Classificação	Funções de similaridade	SIM
Su et al. [2010]	Ajuste do limiar de cada atributo para cálculo da similaridade entre duas duas instâncias	Classificação	Funções de similaridade	SIM
3DR-Indexing	Métricas para identificação de atributos mais relevantes para a etapa de indexação dos registros	Indexação	Densidade, Repetição, Duplicidade, Distintividade	NÃO

Tabela 3.2: Visão geral dos trabalhos relacionados de acordo com os experimentos

Trabalhos	**Principal Contribuição**	**Etapa**
Papadakis et al. [2015]	Avaliação experimental com dois métodos para indexação	Indexação
Baxter et al. [2003]	Comparação de métodos do estado da arte para agrupamento de registros	Clusterização
Christen [2012b]	Comparação de métodos do estado da arte para agrupamento de registros	Clusterização
Draisbach & Naumann [2009]	Comparação de métodos do estado da arte para agrupamento de registros	Clusterização
Cohen et al. [2003]	Comparação de funções de similaridade	Classificação
Christen [2006]	Comparação de funções de similaridade	Classificação
Esta dissertação	Análise do efeito do atributo de indexação sobre as etapas do processo de deduplicação	Indexação e Comparação

Tabela 3.3: Visão geral dos trabalhos relacionados considerando o domínio dos dados

Trabalhos	Principal Contribuição	Domínio	Aprendizagem de Máquina
Borges et al. [2008]	Identificar melhores funções de similaridade a serem utilizadas na comparação das instâncias de registros bibliográficos	Bibliotecas Digitais	NÃO
Borges et al. [2011]	Combinar algoritmos de aprendizagem de máquina e funções de similaridade para detectar duplicatas em referências bibliográficas	Bibliotecas Digitais	SIM
Carvalho et al. [2012]	Método de programação genética para deduplicação de dados	Bibliotecas Digitais	SIM
Sadinle & Fienberg [2013]	Modelo probabilístico para identificar elementos duplicados em sistemas de registros de homicídios	Segurança Pública	NÃO
Carvalho et al. [2015]	Detectar profissionais duplicados em sistemas de saúde	Saúde	NÃO
Borges et al. [2017]	Deduplicação de contatos em dispositivos móveis	Dispositivos Móveis	SIM
Esta dissertação	Identificação de registros duplicados independente do domínio dos dados	Diversos	NÃO

todos para definição de chaves de bloco. Além disso, estudamos o impacto do atributo de indexação sobre as outras etapas do processo.

Ainda na Tabela 3.2, os trabalhos de Baxter et al. [2003], Christen [2012b] e Draisbach & Naumann [2009] têm o objetivo de avaliar os algoritmos de clusterização sem considerar a mudança no atributo de indexação, ou seja, um atributo é escolhido e os experimentos são executados. Nosso trabalho é diferente pois analisa a eficiência e a eficácia que cada atributo de indexação possui nos dois principais métodos de clusterização (i.e., *Standard Blocking* e *Sorted Neighborhood*).

Considerando a Tabela 3.3, os trabalhos de Borges et al. [2017], Carvalho et al. [2015], Carvalho et al. [2011], Levin & Heuser [2010] e Sadinle & Fienberg [2013] são específicos para deduplicação de bibliotecas digitais, dispositivos móveis e outros domínios. Nosso trabalho vai além, pois pode ser adaptado para vários contextos. Além disso, considerando as avaliações experimentais, há trabalhos que analisam apenas a eficácia dos métodos [Borges et al., 2008; Canalle et al., 2017; Levin & Heuser, 2010]. Nosso estudo avalia tanto a eficiência quanto a eficácia do atributo de indexação e do processo de deduplicação de dados como um todo.

No geral, analisando estudos de ambas as tabelas, os trabalhos de Bilenko et al. [2006], Borges et al. [2011], Carvalho et al. [2012] e Evangelista et al. [2010] em suma têm o objetivo de propor soluções de aprendizagem de máquina para resolver o problema da deduplicação de dados em vários contextos, como por exemplo, bibliotecas digitais (Tabelas 3.2 e 3.3). A nossa proposta de trabalho não considera soluções que utilizem aprendizagem de máquina. Essa decisão fundamenta-se nos seguintes fatores: (i) não é tarefa trivial encontrar dados de treinos disponíveis principalmente se tratando de vários domínios; e (ii) o processo de detecção de registros duplicados em ambientes

onlines, por exemplo, tende a deixar de ser um processo *offline* e se torna muitas vezes um processo que precisa ser executado em tempo real como mencionado por Canalle et al. [2017]. Assim, soluções que precisam de treinamento prévio não são viáveis para o contexto da proposta. Por fim, considerando o trabalho de Borges et al. [2008] que em síntese identifica a melhor função de similaridade para registros duplicados em um contexto de bibliotecas digitais, e os trabalhos de Christen [2006] e Cohen et al. [2003] que avaliam funções de similaridade para desambiguação de nomes (Tabelas 3.1 e 3.3). A abordagem desta dissertação é diferente pois define métricas que são utilizadas para identificar atributos relevantes independente do contexto dos dados.

Capítulo 4

Método para Seleção de Atributos Relevantes

Este capítulo introduz o método *3DR-Indexing*: nossa proposta para seleção de atributos relevantes com foco na etapa de indexação do processo de deduplicação de dados. O *3DR-Indexing* cria um *ranking* para cada atributo $a \in A_r$ baseado em uma combinação de métricas extraídas dos valores de cada atributo. O objetivo é calcular a relevância de cada atributo $a \in A_r$ considerando o seu conjunto de dados. O método proposto não utiliza algoritmos de aprendizagem de máquina. Logo, ele pode ser adaptado para qualquer domínio de dados relacionais porque não precisa de um conjunto de dados específico para treino.

Nossa solução possibilita a seleção de atributos relevantes sem a necessidade de intervenção humana, outra vantagem sobre as abordagens existentes. Ademais, o método *3DR-Indexing* define critérios sobre quais atributos de indexação utilizar nos principais algoritmos de clusterização *Standard Blocking* e *Sorted Neighborhood*. Os algoritmos de blocos e vizinhos agrupam os registros de acordo com um valor de chave de bloco criado por meio de um atributo disponível. Assim, a seleção de atributos é fundamental para tais algoritmos e para o processo de deduplicação como um todo. Nossos experimentos demonstram que o resultado do processo de deduplicação de dados é melhorado com a utilização do método *3DR-Indexing* na etapa de indexação (conforme Capítulo 6). A seguir, a Seção 4.1 apresenta as métricas para cálculo da relevância de um atributo. O método *3DR-Indexing* é descrito na Seção 4.2. A Seção 4.3 descreve um exemplo de utilização da abordagem proposta considerando o exemplo do Capítulo 1. O processo de deduplicação utilizando o *3DR-Indexing* é apresentado na Seção 4.4.

4.1 Métricas para Identificação da Relevância de um Atributo no Conjunto de Dados

Nesta seção, definimos um conjunto de métricas baseadas em características extraídas dos valores de cada atributo. As métricas são utilizadas no cálculo da relevância de cada atributo $a \in A_r$ considerando a etapa de indexação.

Dessa forma, considere *notNull* o número de valores não nulos e válidos para um atributo *a*, *dupValues* o número de valores duplicados para *a*, *distValues* o número de valores distintos para *a*, e T o número total de instâncias do conjunto de dados. Então, as Equações 4.1, 4.2, 4.3 e 4.4 definem respectivamente quatro métricas nas quais o *3DR-Indexing* é baseado: (1) **Duplicidade**, que considera a divisão entre os valores duplicados e o total de valores não nulos; (2) **Distintividade**, que consiste da fração entre os valores distintos e o total de valores não nulos; (3) **Densidade**, que considera a fração entre o número de valores não nulos e o total de instâncias; e (4) **Repetição**, que consiste da divisão entre o total de valores repetidos e o total de valores distintos. As métricas são normalizadas por seu valor máximo em A_r e denotadas por $Dup(a)$, $Dist(a)$, $Dens(a)$ e $Rep(a)$, ou seja, as métricas que possuem a barra sobrescrita não estão normalizadas, e as métricas sem a barra contêm seu valor normalizado.

$$\overline{Dup}(a) = \frac{dupValues(a)}{notNull(a)} \qquad (4.1) \qquad \overline{Dist}(a) = \frac{distValues(a)}{notNull(a)} \qquad (4.2)$$

$$\overline{Dens}(a) = \frac{notNull(a)}{T} \qquad (4.3) \qquad \overline{Rep}(a) = \frac{T - distValues(a)}{distValues(a)} \qquad (4.4)$$

É importante notar que tais métricas podem ser facilmente computadas em qualquer conjunto de dados relacional. De fato, elas são comumente apresentadas em histogramas de sistemas relacionais. Por tal motivo, o método *3DR-Indexing* é adaptável para vários domínios de dados. As métricas *Densidade* e *Repetição* também são utilizadas em outros estudos para selecionar atributos relevantes com foco na etapa de classificação [Canalle et al., 2017] (mas esta é a primeira vez na etapa de indexação).

Além disso, um bom atributo de indexação deve ter boa qualidade (para evitar blocos com valores nulos) e boa frequência (para produzir um rápido tempo de deduplicação) [Christen, 2012a]. Assim, a métrica **Densidade** seleciona os atributos com maior *qualidade* porque atributos mais densos geram uma menor quantidade de blocos nulos. A **Duplicidade** melhora o processo de deduplicação em termos de eficácia, pois quando os atributos dos registros comparados são idênticos, a probabilidade de determinar se o par de instâncias se referem à mesma entidade é maior na etapa

de classificação. As métricas **Distintividade** e **Repetição** são complementares, pois baixa *Distintividade* produz alta *Repetição* e vice-versa. No entanto, ambas métricas são necessárias para classificar os atributos, porque garantem um melhor equilíbrio entre o total de blocos e o tamanho dos mesmos. A **Distintividade** evita que atributos com baixa distinção sejam selecionados, pois o número de pares candidatos nos blocos é alto e produz um elevando tempo de deduplicação.

Finalmente, a métrica de **Repetição** estima um valor médio de registros por bloco e aproxima o tempo de deduplicação de cada atributo. Via de regra, os blocos são criados conforme a distribuição dos registros, ou seja, o número de valores distintos para cada atributo determina o total de blocos e a repetição a quantidade de pares, considerando uma metodologia de indexação *Schema-Agnostic*, por exemplo. Segundo Christen [2012b], não é uma tarefa trivial predizer o número total de pares candidatos que são criados por meio da indexação, pois a maioria dos conjuntos de dados não segue uma distribuição normal.

4.2 O Método 3DR-Indexing

Nossa solução define um *score* para cada atributo baseado em seus valores de *Densidade*, *Duplicidade*, *Distintividade* e *Repetição*. O objetivo é melhorar o *trade-off* entre eficiência e eficácia do processo de deduplicação de dados, ou seja, selecionar os atributos de melhor eficácia que executem o processo de deduplicação mais rápido.

Assim, intuitivamente, as métricas *Densidade* e *Duplicidade* melhoram a eficácia porque atributos densos e duplicados geram blocos mais eficazes. Por outro lado, altos valores de *Repetição* e *Distintividade* produzem elevado tempo de processamento porque eles criam poucos blocos com muitos registros (i.e., muitas comparações) ou muitos blocos com poucos registros (i.e., muitos acessos ao banco de dados para selecionar os registros de cada bloco distinto). Dessa forma, nós consideramos o complemento dessas métricas para melhorar a eficiência, porque atributos sem muita *Repetição* e *Distintividade* executam o processo de deduplicação mais rápido. Além disso, nós multiplicamos $(1 - Dist(a))$ pela Dens(a) para evitar que baixos valores de *Distintividade* e *Densidade* produzam altos *scores*, ou seja, nós consideramos a *Densidade* de cada atributo como peso para a *Distintividade* do mesmo.

Especificamente, a Equação 4.5 calcula a relevância $\overline{R}(a)$ de um atributo. O valor $\overline{R}(a)$ é normalizado para obter o *score* de relevância do atributo $R(a)$ (i.e. cada valor de $\overline{R}(a)$ é normalizado pelo maior valor de $\overline{R}(a)$). Dessa forma, $\overline{R}(a)$, com a barra sobrescrita, representa a relevância de um atributo não normalizada e $R(a)$, sem

Tabela 4.1: Registros duplicados em duas fontes de dados (Tabela 1.1 novamente)

CD	ID	Fonte	Título	Artista	Categoria	Gênero	Extra	Ano
A	1	Atlantic Records	The Moment	Kenny G	Jazz	Jazz	ID3G:8	1996
	2	J Records	Moment	Kenny G	Null	Classical	Null	1996
B	3	Atlantic Records	Donde Hay Musica	Ramazzotti	Jazz	Ballad	Null	Null
C	4	Atlantic Records	Alien Ant Farm	Anthology	Rock	Rock	Null	Null
	5	J Records	Alien Ant Farm	Anthology	NewAge	AlternRock	ID3G:40	2001

Tabela 4.2: Seleção dos melhores atributos para a etapa de indexação

Atributo	Densidade	Duplicidade	Distintividade	Repetição	$\overline{\mathbf{R}}(a)$	$\mathbf{R}(a)$
Artista	1,00	0,67	0,60	0,44	2,62	1,00
Fonte	1,00	1,00	0,40	1,00	2,60	0,99
Título	1,00	0,33	0,80	0,17	2,37	0,90
Categoria	0,80	0,33	0,80	0,17	2,21	0,84
Ano	0,60	0,56	0,67	0,33	2,02	0,77
Gênero	1,00	0,00	1,00	0,00	2,00	0,76
Extra	0,40	0,00	1,00	0,00	1,40	0,53

a barra, é sua relevância normalizada. Além disso, note que a Equação 4.5 soma as métricas de relevância normalizadas, ou seja, sem a barra sobrescrita.

$$\overline{R}(a) = Dens(a) + Dup(a) + (1 - Dist(a)) * Dens(a) + (1 - Rep(a)) \tag{4.5}$$

4.3 Exemplo

Nesta seção, apresenta-se o método *3DR-Indexing* sobre o exemplo simples exibido na Tabela 1.1 do Capítulo 1, reapresentada na Tabela 4.1 para melhor leitura. Calculamos as métricas *Densidade*, *Duplicidade*, *Distintividade* e *Repetição* para cada atributo do conjunto de dados do exemplo (i.e., *Fonte*, *Título*, *Artista*, *Categoria*, *Gênero*, *Extra* e *Ano*) e calculamos sua relevância por meio da Equação 4.5. A Tabela 4.2 apresenta os valores das métricas, a relevância calculada e a relevância normalizada para cada atributo do exemplo, ordenados pelo valor da relevância. Note que os valores das métricas *Densidade*, *Duplicidade*, *Distintividade* e *Repetição* estão normalizados. Por exemplo, considerando a métrica *Duplicidade*, o atributo *Fonte* contém mais valores duplicados quando comparado com os demais atributos, por isso o valor de *Duplicidade* do atributo *Fonte* é $1,00$. Ademais, observe que as métricas *Distintividade* e *Repetição* são o complemento das Métricas $Dist(a)$ e $Rep(a)$, ou seja, o atributo *Fonte*, por exemplo, possui menos valores repetidos que os demais atributos. Logo, o valor de *Repetição* para o atributo *Fonte* é $1,00$.

Conforme apresentado na Tabela 4.2, o melhor atributo para ser utilizado na etapa de indexação é o atributo *Artista* (i.e., de maior relevância $R(a)$). Assim, espera-

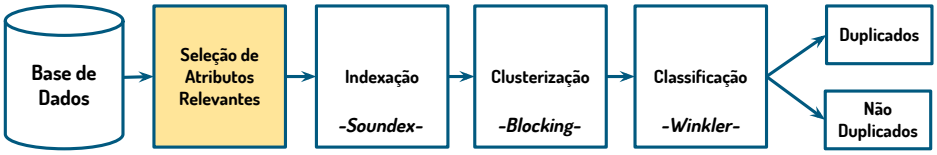


Figura 4.1: Aplicação do método *3DR-Indexing* no processo de deduplicação de dados

se que o resultado da deduplicação de dados considerando o *trade-off* entre eficiência e eficácia seja melhor com a utilização do atributo *Artista* na etapa de indexação. Por outro lado, o resultado do processo não seria eficiente e eficaz utilizando o atributo *Extra*, por exemplo, porque ele possui o menor valor de relevância. De forma geral, busca-se também que o método *3DR-Indexing* não classifique um atributo de menor importância como relevante, pois essa classificação diminuiria a qualidade dos resultados da deduplicação e a eficácia do método proposto. Essas expectativas são confirmadas na avaliação experimental apresentada do Capítulo 6.

4.4 Deduplicação de Dados Utilizando o Método *3DR-Indexing*

Esta seção apresenta a aplicação do método proposto no processo de deduplicação de dados. De forma geral, o *3DR-Indexing* pode ser visto como uma etapa de pré-processamento para a deduplicação, porque inicialmente o método classifica os atributos de acordo com a relevância e, em seguida, utiliza os mais relevantes na indexação.

A Figura 4.1 exibe o processo de deduplicação quando o *3DR-Indexing* é utilizado. O processo é executado da seguinte maneira.

- Inicialmente, a relevância de cada atributo do conjunto de dados é calculada utilizando a Equação 4.5;

- Em seguida, o atributo de maior relevância é utilizado na etapa de indexação, ou seja, o atributo que possui o maior valor de $R(a)$ segundo método *3DR-Indexing*. Além disso, o seu valor é codificado por meio da função *Soundex* produzindo um valor chave de bloco para cada instância;

- Depois, os registros são agrupados por valor de chave de bloco utilizando o algoritmo *Standard Blocking* na etapa de clusterização;

- Por fim, os dois atributos mais relevantes segundo o *3DR-Indexing* são analisados para classificar cada par de registros, ou seja, os dois atributos com maior valor de $R(a)$. Para tal, a função *Jaro Winkler* é aplicada sobre o valor dos dois atributos para computar o valor de similaridade. Depois, de acordo com o limiar definido, as duas instâncias são classificadas como duplicadas ou não duplicadas.

4.5 Considerações Finais

Este capítulo apresentou o método *3DR-Indexing*. Inicialmente, foram detalhadas as métricas utilizadas para cálculo da relevância de um atributo em um conjunto de dados. Em seguida, o método *3DR-Indexing* foi introduzido. Além disso, um exemplo de utilização do método proposto foi apresentado, considerando o exemplo do Capítulo 1. Por fim, a aplicação do *3DR-Indexing* no processo de deduplicação foi detalhada.

O capítulo seguinte detalha as avaliações experimentais para analisar o efeito da escolha do atributo de indexação sobre o processo de deduplicação de dados. Além disso, os experimentos de validação do método *3DR-Indexing* são descritos.

Capítulo 5

Projeto Experimental

Este capítulo apresenta o projeto experimental utilizado para avaliar o método proposto. Inicialmente, a Seção 5.1 detalha o método de avaliação do processo de deduplicação de dados, do atributo de indexação sobre as etapas do processo e do 3DR-Indexing. Os conjuntos de dados e a ferramenta para geração de dados sintéticos são discutidos na Seção 5.2. Por fim, a Seção 5.3 apresenta as configurações e métricas adotadas nos experimentos.

5.1 Metodologia dos Experimentos

Nós implementamos os algoritmos do processo de deduplicação de dados em Java (Capítulo 2) e executamos todos os experimentos em um computador *desktop* com processador Intel *i*7 (2.3 GHz) e 16GB de memória RAM, rodando o sistema operacional MAC OS X 10.11.3. Todos os algoritmos e *datasets* estão publicamente disponíveis no endereço `http://www.dcc.ufmg.br/~mirella/projs/deduplica`.

As avaliações experimentais são divididas em cinco categorias conforme questões de pesquisa elencadas no Capítulo 1. As questões de pesquisa são: **(Q1)** Qual parte possui maior impacto no processo de deduplicação de dados entre a função de indexação, o atributo de indexação e o algoritmo utilizado para clusterização dos registros? **(Q2)** O mesmo atributo de indexação é eficiente e eficaz nos métodos de indexação *Schema-Agnostic* e *Schema-Based Configurations*? **(Q3)** Os resultados da deduplicação são eficazes utilizando o mesmo atributo de indexação nos algoritmos *Standard Blocking* e *Sorted Neigborhood*? **(Q4)** Nossa solução encontra os melhores atributos para a etapa de indexação (baseada em características extraídas dos valores dos atributos)? **(Q5)** Os resultados do processo de deduplicação de dados são melhorados com a utilização da combinação de atributos na etapa de indexação? Nós não avaliamos o

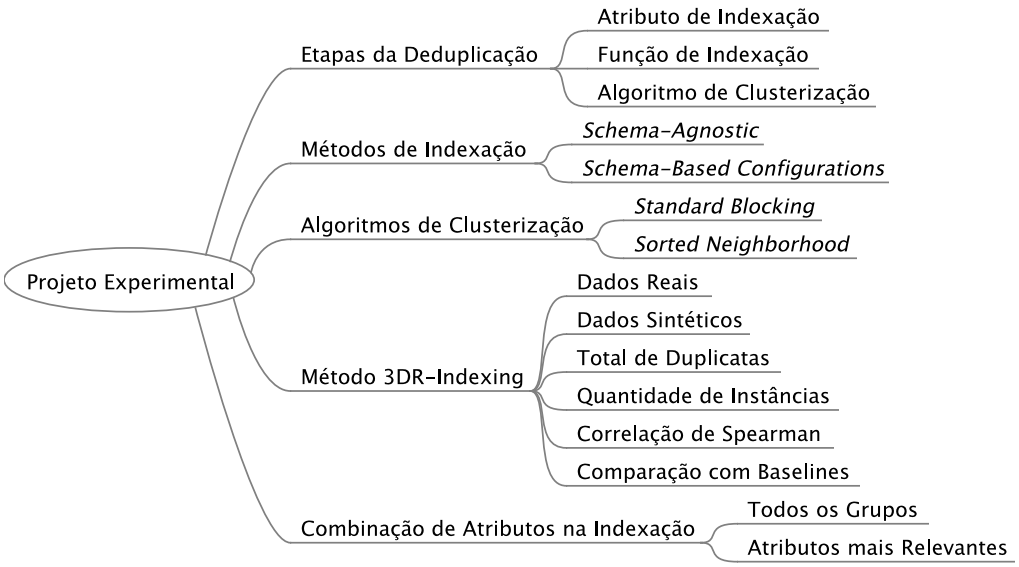


Figura 5.1: Projeto experimental com as dimensões avaliadas e variáveis consideradas em cada dimensão

impacto do atributo sobre a etapa de classificação porque um estudo é apresentado por Canalle et al. [2017]. A Figura 5.1 sumariza os experimentos, os quais são detalhados nas seções seguintes. Os experimentos consideram dois aspectos, eficiência e eficácia. Assim, a métrica *F-Measure* é adotada no quesito eficácia. Para a eficiência, nós calculamos o tempo para recuperar todos os valores de BK distintos, as tuplas de cada BK e executar o processo de deduplicação.

5.1.1 Análise do Atributo de Indexação sobre as Etapas do Processo de Deduplicação de Dados

Neste experimento, utilizamos a metodologia de projetos fatoriais para avaliar o impacto do atributo de indexação sobre o processo de deduplicação de dados. O objetivo é calcular não apenas o efeito do atributo de indexação, mas também o efeito dos algoritmos envolvidos nas outras etapas do processo. O projeto fatorial avalia o efeito de um fator X sobre uma variável resposta Y [Jain, 1992]. Desse modo, nós executamos um projeto fatorial 2^3r com dez replicações ($r = 10$), isto é, cada configuração do projeto é analisada sobre dez *datasets* sintéticos (descritos na Seção 5.2). As replicações são definidas com um intervalo de confiança de 95% e um erro máximo de 5%.

Um projeto fatorial 2^3r analisa três fatores sobre uma variável resposta Y. Desse modo, os fatores analisados são: o atributo de indexação, a função de indexação e o algoritmo de clusterização dos registros. Os dados sintéticos contêm oito atributos. As-

Tabela 5.1: Configurações do projeto fatorial

#**Exp.**	**Fator (A) Indexação**	**Fator (B) Algoritmo**	**Fator (C) Atributo**
1	Soundex	Standard Blocking	Given Name
2	Suffix	Standard Blocking	Given Name
3	Soundex	Sorted Neighborhood	Given Name
4	Suffix	Sorted Neighborhood	Given Name
5	Soundex	Standard Blocking	Address2
6	Suffix	Standard Blocking	Address2
7	Soundex	Sorted Neighborhood	Address2
8	Suffix	Sorted Neighborhood	Address2

sim, nós avaliamos empiricamente os atributos *Given Name* e *Address2* (i.e., o melhor e o pior atributo). Para a indexação, os métodos *Soundex* e *Suffix* são utilizados. Por fim, utilizamos os algoritmos *Standard Blocking* e *Sorted Neighborhood* para clusterização. A junção desses fatores é responsável por gerar todos os pares de registros que, posteriormente, são comparados na etapa de classificação, ou seja, quanto melhor a combinação deles, maior a probabilidade de detecção dos registros duplicados na etapa de classificação. De forma geral, os experimentos analisam quais dos fatores possuem maior impacto sobre o processo, além de quantificar o efeito do atributo de indexação.

Especificamente, um projeto fatorial 2^3r executa oito experimentos com a combinação dos fatores. Cada fator é representado por dois níveis, ou seja, duas opções de configuração. A Tabela 5.1 detalha cada experimento e as configurações de cada fator. Por exemplo, o experimento número **(1)** utiliza o atributo de indexação *Given Name*, codificado pela função *Soundex* e agrupa os registros com o algoritmo *Standard Blocking*. A função de *Jaro Winkler* é utilizada na etapa de classificação nos oito experimentos. Nós não realizamos variações na classificação, porque o objetivo é estudar as etapas do processo de clusterização dos registros, ou seja, primeira e segunda etapas e, especificamente, a etapa de indexação. A classificação determina apenas se o par de registros é uma cópia, e um estudo é encontrado em Canalle et al. [2017]. Ao final do processo, a métrica *F-Measure* é computada, isto é, variável resposta Y. Os experimentos 2 a 8 seguem a mesma lógica.

5.1.2 Análise do Atributo de Indexação nos Métodos *Schema-Agnostic* e *Schema-Based Configurations*

Existem diversas abordagens para definir as chaves de bloco. As mais tradicionais utilizam todo o valor dos atributos para gerar as chaves (e.g., *Schema-Agnostic*) e outras baseiam-se em regras extraídas dos valores dos atributos para definir as BKs (e.g., *Schema-Based Configurations*).

Nesse contexto, esta avaliação experimental analisa a eficiência e eficácia dos atributos de indexação utilizando as duas abordagens (*agnostic* e *configurations-based*). Assim, na etapa de indexação do processo de deduplicação, nós utilizamos os valores dos atributos como valor de chave de bloco, ou aplicamos a codificação *Soundex* sobre os valores dos atributos para gerar as chaves. Nas outras etapas, nós aplicamos os algoritmos *Standard Blocking* ou *Sorted Neighborhood* e *Jaro Winkler* para classificação. Especificamente, o objetivo é avaliar se o mesmo atributo de indexação tem resultados eficientes e eficazes independente da configuração utilizada na primeira etapa. Para isso, nós computamos a *F-Measure* e o tempo de execução de cada configuração de indexação utilizando os atributos disponíveis nos conjuntos de dados.

5.1.3 Análise do Atributo de Indexação nos Algoritmos *Standard Blocking* e *Sorted Neighborhood*

Na segunda etapa do processo, o objetivo é agrupar os registros de acordo com um critério, geralmente uma chave de bloco. Abordagens tradicionais formam os grupos por valor de BK ou por vizinhos mais próximos. Desse modo, o objetivo dessa avaliação experimental é analisar o impacto do atributo de indexação nos dois principais algoritmos de clusterização (*Standard Blocking* e *Sorted Neighborhood*), ou seja, quantificar se o mesmo atributo de indexação é eficiente e eficaz em ambas abordagens. Especificamente, no algoritmo *Sorted Neighborhood*, nós computamos apenas o valor de eficácia em termos de *F-Measure*, porque o tempo de execução do algoritmo está relacionado com o tamanho da janela w [Hernández & Stolfo, 1995]. Devido ao escopo do trabalho, a análise da janela w não é contemplada nas avaliações experimentais, porque o objetivo do trabalho é estudar a eficiência e eficácia do processo de deduplicação de dados quando o atributo de indexação muda em vez da mudança no tamanho da janela. Análises do tamanho da janela do algoritmo de vizinhança são encontradas em [Baxter et al., 2003] e [Draisbach & Naumann, 2009].

Assim, nós executamos o processo de deduplicação de dados variando tanto o atributo de indexação (utilizando todos os disponíveis em cada *dataset*) quanto o algoritmo de clusterização (blocos e vizinhos), ou seja, combinamos cada atributo de indexação com cada algoritmo de agrupamento. A função de *Jaro Winkler* é utilizada para calcular a similaridade na etapa de classificação. Por fim, nós computamos a *F-Measure* para cada atributo em cada algoritmo.

5.1.4 Avaliação do Método *3DR-Indexing*

A avaliação do método *3DR-Indexing* consiste em executar todo o processo de deduplicação de dados utilizando diferentes configurações para o atributo de indexação. Nas outras etapas do processo, os algoritmos *Standard Blocking* e *Jaro Winkler* são adotados para clusterização e classificação respectivamente. Na etapa de classificação, a similaridade é calculada entre os atributos mais descritivos de cada *dataset*, conforme descrito na Seção 5.3. Em seguida, o tempo de execução e a *F-Measure* são computados para cada atributo de indexação utilizado na primeira etapa do processo.

O método proposto é avaliado em dados sintéticos e reais. Nos conjuntos de dados sintéticos, a análise considera a variação do total de instâncias ($10^2 - 10^6$) e da quantidade de registros duplicados ($10\% - 90\%$). O objetivo é mostrar que os melhores resultados da deduplicação de dados são conseguidos utilizando os atributos mais relevantes segundo o método *3DR-Indexing* na etapa de indexação (considerando o *trade-off* entre eficiência e eficácia), ou seja, o método proposto seleciona os atributos mais relevantes para a etapa de indexação.

5.1.5 Análise da Combinação de Atributos na Etapa de Indexação do Processo de Deduplicação

Além de analisar individualmente o atributo de indexação, os experimentos avaliam a combinação de atributos na etapa de indexação, ou seja, utilizar mais de um atributo na etapa de indexação melhora a eficiência e eficácia do processo de deduplicação de dados? Tais experimentações aplicam-se apenas ao algoritmo *Standard Blocking* porque no algoritmo *Sorted Neighborhood* os resultados do processo não são melhorados com a utilização de mais de um atributo de indexação [Hernández & Stolfo, 1995].

Desse modo, nós executamos o processo de deduplicação utilizando uma combinação de atributos na indexação, o algoritmo *Standard Blocking* para clusterização e o algoritmo de *Jaro Winkler* para classificação. Especificamente, para definir os grupos de atributos, alguns conjuntos de dados consideram todas as combinações possíveis e outros avaliam a combinação dos atributos mais relevantes de acordo com o método *3DR-Indexing* (conforme apresentado posteriormente no Capítulo 6). Por exemplo, a eficácia do processo de deduplicação é melhorada com a combinação dos três atributos mais relevantes segundo o *3DR-Indexing*? Assim, nós calculamos o tempo de execução e a *F-Measure* para cada combinação de atributos utilizada na etapa de indexação.

5.2 Conjuntos de Dados Experimentais

Os experimentos utilizam dados sintéticos e reais, os quais contêm uma diversidade de contextos e tipos de atributos. Assim, o método proposto e o atributo de indexação são avaliados em uma perspectiva de vários domínios.

Não é uma tarefa trivial adquirir dados reais contendo registros originais e duplicados. Muitos conjuntos de dados reais contêm informações pessoais, então problemas de privacidade e confidencialidade fazem com que eles não estejam disponíveis publicamente. Por essa razão, quando *datasets* reais não estão disponíveis, a alternativa é utilizar dados sintéticos. Tal escolha tem a vantagem de conhecer previamente o número de erros e modificações, o que leva a experimentos mais controlados.

Nesse contexto, nós produzimos dados sintéticos utilizando a ferramenta *Data Set Generator Program*[7] [Christen, 2012a], que cria registros originais e duplicados de acordo com os parâmetros configurados pelo usuário. Essa ferramenta é parte do projeto Febrl[8] e é baseada nas ideias apresentadas por Hernández & Stolfo [1995]. Com a ferramenta é possível gerar um *dataset* contendo os seguintes atributos: nome (baseado em uma tabela de frequência para nomes e sobrenomes), endereços (baseado em tabelas de frequência para endereços, códigos postais, ruas, estados e territórios), datas (como data de nascimento), número de telefones e registros identificadores.

Além disso, os seguintes parâmetros são definidos para gerar um conjunto de dados sintético: *number of original records, number of duplicate records, maximum amount of duplicates by record, maximum amount of changes by field, maximum amount of changes by instances, probability distribution (uniform, poisson or zipf), type of change (type, ocr, phonetic, all)*, e *number of households*. Considerando os parâmetros numéricos (com exceção da *probability distribution* e do *type of change*), o usuário escolhe tanto o total de modificações quanto o número de registros originais e duplicados. Por exemplo, cada registro duplicado pode ser criado com cinco modificações por atributo e três modificações por instância. Especificamente, a avaliação experimental dos dados sintéticos considera oito atributos: *Given Name*, *Surname*, *Address1*, *Address2*, *Suburb*, *Culture*, *State* e *Title*. Um estudo analisando o *Data Set Generator Program* é encontrado em Silva & Moro [2017].

Ademais, são utilizadas as configurações descritas na Tabela 5.2 para criar os dados sintéticos. Para defini-las, nós analisamos estudos que utilizam o *Data Set Generator Program* e, em seguida, escolhemos empiricamente os valores mais comuns para

[7]Dataset Generator: `https://cs.anu.edu.au/people/Peter.Christen/Febrl/febrl-0.3/febrldoc-0.3/node70.html`

[8]Febrl: `http://sourceforge.net/projects/febrl`

Tabela 5.2: Configurações para geração dos conjuntos de dados sintéticos

Conjunto de Dados	MAXDUP	MODBYFIELD	MODBYINST	TYOFCH	DIST	H
FactorialProject	3	5	5	all	Uniform	1
Duplicate 10% a 90%	2	2	2	all	Uniform	1
Dataset 10^2 a 10^6	3	3	3	all	Uniform	1
DatasetSynthetic	3	2	2	all	Uniform	1

MAXDUP = quantidade máxima de duplicatas por instância (*maximum amount of duplicates by record*)
MODBYFIELD = número máximo de modificações por atributo (*maximum amount of changes by field*)
MODBYINST = quantidade máxima de modificações por instância (*maximum amount of changes by instances*)
TYOFCH = tipo de modificação: erros tipográficos, fonéticos, etc. (*type of change*)
DIST = distribuição de probabilidade (*distribution*)
H = quantidade máxima de registros em um grupo (*households*)

cada parâmetro [Beskales et al., 2009; Christen, 2008; Carvalho et al., 2012; Draisbach et al., 2012; Goiser & Christen, 2006; Sadinle, 2017; Silva & Moro, 2017; Steorts et al., 2014]. Os conjuntos de dados que variam o total de duplicados são criados com a mesma configuração (i.e., do *Duplicate10%* a *Duplicate90%*), a única diferença é o total de registros duplicados. Do mesmo modo, os *datasets* com instâncias variadas (i.e., do 10^2 a 10^6), onde a diferença é o total de instâncias em cada conjunto de dados. Os outros parâmetros da ferramenta são descritos na Tabela 5.3 para cada conjunto de dados sintético (i.e., total de registros originais e duplicados).

Além dos dados sintéticos, a avaliação experimental considera dados reais. Assim, os conjuntos *Cora*, *Restaurant* e *CDs* information do DuDe toolkit[9] são utilizados. Os *datasets* escolhidos são amplamente utilizados em pesquisas de deduplicação de dados, e.g., [Bilenko & Mooney, 2003], [Canalle et al., 2017], [Canuto et al., 2013], [Chen et al., 2012] , [Christen, 2012b], [Cohen & Richman, 2002], [Draisbach et al., 2012], [Hand & Christen, 2018], [Leitão et al., 2007], [Levin & Heuser, 2010], [McCallum et al., 2000], [Papadakis et al., 2015], [Su et al., 2010] e [Vieira et al., 2017].

Especificamente, o conjunto de dados *Cora* inclui informações bibliográficas sobre artigos científicos, contendo 1.879 instâncias, das quais 1.688 são duplicadas. Nós analisamos os atributos: *Author*, *Title*, *Booktitle*, *Journal*, *Publisher*, *Address*, *Editor*, *Tech*. O *dataset Restaurant* é uma coleção de 864 registros de restaurantes do guia Fodor's e Zagat's e contém 112 registros duplicados. Os atributos avaliados são: *Name*, *Address*, *City* e *Type*. O conjunto de dados *CDs* contém 9.763 registros de CDs extraídos aleatoriamente do *freeDB* e possui 222 tuplas duplicadas. Nesse *dataset*, nós avaliamos os atributos: *Artist*, *Title*, *Category*, *Genre*, *CDextra*. A Tabela 5.3 sumariza os *datasets* onde os três primeiros são reais e os demais sintéticos. Os conjuntos de dados *Duplicate10%, 30%, 50%, 70%* e *90%* avaliam a quantidade de registros duplicados na base de dados. O quesito escalabilidade é analisado nos *datasets Dataset*10^2 ao

[9]Duplicate Detection (DuDe) toolkit: `https://hpi.de/naumann/projects/data-quality-and-cleansing/dude-duplicate-detection.html`

Tabela 5.3: Conjuntos de dados experimentais

Conjunto de Dados				Replicações	
#	**Dataset**	**#Instâncias**	**#Duplicados**	**#Eficiência**	**#Eficácia**
1	Cora	1.879	1.688	50	1
2	Cds	9.763	222	50	1
3	Restaurant	864	112	50	1
4	Duplicate10%	11.000	1.000	30	10
5	Duplicate30%	13.000	3.000	30	10
6	Duplicate50%	15.000	5.000	30	10
7	Duplicate70%	17.000	7.000	30	10
8	Duplicate90%	19.000	9.000	30	10
9	Dataset10^2	110	10	100	5
10	Dataset10^3	1.100	100	30	5
11	Dataset10^4	11.000	1.000	30	5
12	Dataset10^5	110.000	10.000	30	5
13	Dataset10^6	1.100.000	100.000	10	5
14	FactorialProject	11.000	1.000	não se aplica	10
15	DatasetSynthetic	11.000	1.000	30	10

Dataset10^6. O conjunto *FactorialProject* avalia o projeto fatorial, que analisa apenas a eficácia do processo (variável resposta Y), e as avaliações da combinação de atributos e dos métodos *Agnostic* e *Based Configurations* utilizam o conjunto de dados *DatasetSynthetic*.

Por fim, nós utilizamos um intervalo de confiança de 95% e um erro máximo de 5% nas avaliações experimentais. Desse modo, os experimentos de eficiência e eficácia são replicados conforme valores descritos na Tabela 5.3 (colunas *#Eficiência* e *#Eficácia*). Por exemplo, para cada atributo do conjunto de dados *Cora* (e.g., *Title*), o tempo de execução é calculado 50 vezes. Tais replicações ocorrem tanto para o algoritmo *Standard Blocking* quanto para o *Sorted Neighborhood*. Considerando os experimentos de eficácia, a *F-Measure* é computada apenas uma vez nos conjuntos de dados reais (i.e., *Cora*, *CDs* e *Resturant*), porque os algoritmos aplicados no processo de deduplicação de dados são determinísticos. Para os sintéticos, dez ou cinco *datasets* são criados e a *F-Measure* média é computada (i.e., do número quatro ao número 15).

5.3 Configurações e Métricas de Avaliação

Os algoritmos utilizados na avaliação experimental precisam ser configurados em três parâmetros: o limiar de similaridade para o algoritmo *Jaro Winkler*, o tamanho da janela w para o algoritmo *Sorted Neighborhood* e o tamanho K dos sufixos. Nós adotamos as configurações apresentadas por Silva et al. [2017]: similaridade de $0,9$, sufixos de tamanho *cinco*, e o tamanho da janela w é definido de acordo com o número de instâncias do conjunto de dados, ou seja, $w = 50$ para *datasets* com menos de 10.000

instâncias; $w = 150$ para conjuntos de dados com mais de 10.000 e menos que 50.000 instâncias; e $w = 500$ para *datasets* com mais de 50.000 instâncias.

Considerando a etapa de classificação, nós utilizamos a função *Jaro Winkler* em todos os experimentos. Para determinar se os registros são cópias, os dois atributos mais significativos de cada conjunto de dados são comparados (conforme *ranking* do método *3DR-Indexing*, ou seja, os dois atributos com maior valor de $R(a)$). Assim, a etapa de classificação compara *Given Name* e *Surname* nos conjuntos de dados sintéticos, *Author* e *Title* no *dataset Cora*, *Artist* e *Genre* para *CDs* e *Name* e *Address* no conjunto de dados *Resturant*. Ademais, os experimentos consideram apenas atributos não numéricos (i.e., categóricos), por conta da função de indexação *Soundex*, que é específica para tais tipos atributos.

Por fim, nós utilizamos as métricas Revocação, Precisão, *F-Measure* e correlação de *Spearman* para avaliar o processo de deduplicação de dados, a eficácia do atributo de indexação e o *3DR-Indexing*. A Revocação calcula o total de duplicatas corretas que são identificadas, ou seja, verdadeiros positivos (Equação 5.1). Os falsos positivos acrescidos dos verdadeiros positivos são calculados pela métrica Precisão (Equação 5.2), e a *F-Measure* é uma média harmônica entre Revocação e Precisão (Equação 5.3). Os *scores* estão no intervalo $[0, 1]$, onde 1 representa a melhor eficácia. Segundo Hand & Christen [2018], essas métricas são amplamente usadas em experimentos de deduplicação de dados. O coeficiente de correlação de *Spearman* (ρ) indica uma relação de ordem entre duas variáveis X e Y [Cohen, 1988]. A Equação 5.4 calcula a correlação ρ, onde d_i^2 representa a diferença entre as posições de cada observação das variáveis. A métrica está no intervalo $[-1, 1]$, onde 1 significa que a ordenação da variável Y segue a ordenação da variável X, -1 significa que Y segue a ordenação inversa da variável X, e 0 que não existe correlação de ordem entre as variáveis X e Y.

$$Revocação = \frac{VP_{classificados}}{Positivos} \tag{5.1}$$

$$Precisão = \frac{VP_{classificados}}{VP_{classificados} + FP_{classificados}} \tag{5.2}$$

$$F\text{-}Measure = 2 * \left(\frac{Precisão * Revocação}{Precisão + Revocação} \right) \tag{5.3}$$

$$\rho = 1 - \frac{6 * \sum d_i^2}{n * (n^2 - 1)} \tag{5.4}$$

5.4 Considerações Finais

Este capítulo apresentou o projeto das avaliações experimentais da escolha do atributo de indexação sobre o processo de deduplicação de dados e do método *3DR-Indexing*. Além disso, foram apresentados os conjuntos de dados sintéticos e reais utilizados nos experimentos, bem como a ferramenta para criação de dados sintéticos. Por fim, as configurações experimentais e as métricas de avaliação foram também apresentadas.

Com base nessas configurações, o capítulo seguinte apresenta e discute os resultados experimentais de cada uma das avaliações. Ademais, o capítulo posterior compara os resultados da deduplicação utilizando o método *3DR-Indexing* com dois *baselines*.

Capítulo 6

Resultados e Discussões

Este capítulo apresenta os resultados experimentais e discussões. Inicialmente, a Seção 6.1 analisa o efeito das etapas sobre processo de deduplicação de dados. O atributo de indexação é avaliado nos métodos de indexação *Schema-Agnostic* e *Schema-Based Configurations* e nos algoritmos *Standard Blocking* e *Sorted Neighborhood* na Seção 6.2. A Seção 6.3 discute as análises do método *3DR-Indexing*. A combinação de atributos é avaliada na Seção 6.4. Por fim, a Seção 6.5 apresenta discussões.

6.1 Projeto Fatorial

O processo de deduplicação de dados é composto da escolha de vários fatores incluindo o atributo de indexação, a função de indexação e o algoritmo de clusterização dos registros. Nesse contexto, esta seção apresenta o resultado do projeto fatorial 2^3r que avalia quais fatores do processo de deduplicação de dados possui maior efeito. O objetivo é responder a questão de pesquisa **Q1**: Qual parte possui maior impacto no processo de deduplicação de dados entre a função de indexação, o atributo de indexação e o algoritmo utilizado para clusterização dos registros?

Para abordar **Q1**, nós analisamos os seguintes fatores e níveis: (A) as funções de indexação *Soundex* e *Suffix*, (B) os algoritmos de clusterização *Standard Blocking* e *Sorted Neighborhood*, e (C) o melhor e o pior atributo de indexação, *Given Name* e *Address2*. A Figura 6.1 exibe os resultados do projeto fatorial. O eixo-X representa cada fator analisado, bem como suas iterações, e o eixo-Y a porção de efeito explicada por cada fator e suas iterações. Note que o intervalo de confiança e o erro experimental também são exibidos. Os resultados mostram que o atributo de indexação possui maior efeito sobre o processo - efeito de $96,09 \pm (3,46)$, ou seja, o atributo usado na indexação é responsável por explicar a maior variação nos resultados em termos

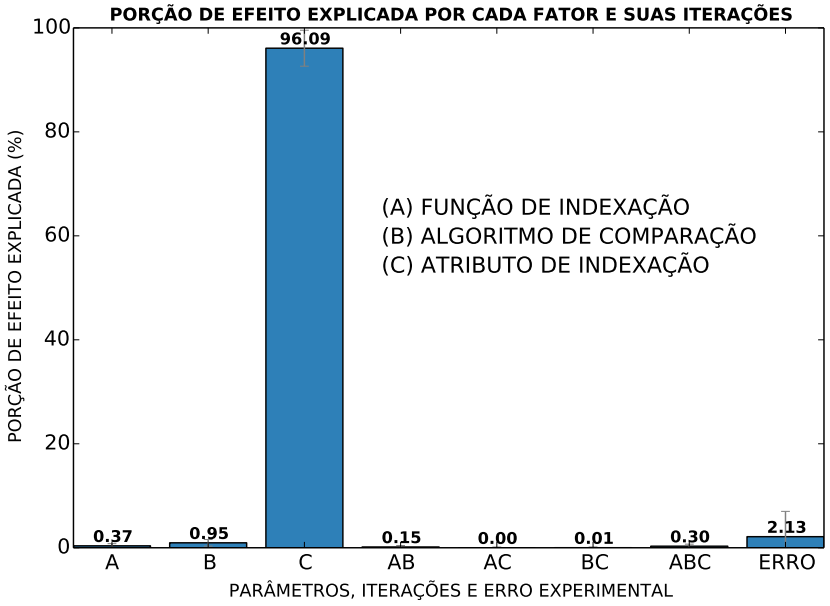


Figura 6.1: Efeito explicado por cada fator e suas iterações no projeto fatorial 2^3r

de *F-Measure*. Assim, conclui-se que, dependendo do atributo escolhido na etapa de indexação, os valores de *F-Measure* podem aumentar ou diminuir prejudicando a eficácia da deduplicação, pois utilizar o melhor atributo de indexação ou o pior altera significativamente os valores de *F-Measure*. Além disso, as iterações entre os fatores são insignificantes, porque a porção de efeito explicada pelas iterações não atinge nem 1% dos resultados.

Complementando tais resultados, a Tabela 6.1 exibe os valores de *F-Measure* alcançados em cada configuração do projeto fatorial. Os resultados estão ordenados pelo fator C para destacar as diferenças de *F-Measure* entre os atributos *Given Name* e *Address2*. O maior valor de *F-Measure* é alcançado no experimento *quatro* - $0,7343 \pm (0,0141)$, onde o atributo de indexação *Given Name* é utilizado. Por outro lado, o experimento *seis* tem o resultado mais ineficaz para a deduplicação - atributo *Address2* e *F-Measure* de $0,3835 \pm (0,0311)$. O melhor resultado de *Given Name* e o pior de resultado *Address2* diferem em aproximadamente $0,3508$ em termos de *F-Measure* (experimentos *quatro* e *seis* respectivamente), ou seja, quando o atributo *Given Name* é utilizado na indexação, a eficácia do processo aumenta em cerca de 91%.

Por fim, comparando os níveis da função de indexação (*Soundex* e *Suffix*) e do algoritmo de clusterização (blocos e vizinhos), os resultados mostram que a mudança das técnicas não produz uma variação significante de *F-Measure* porque as diferenças são mínimas. Por exemplo, os resultados do processo são eficazes utilizando o algoritmo *Standard Blocking* (experimento *um*) e o algoritmo *Sorted Neighborhood* (experimento

Tabela 6.1: Resultados do projeto fatorial em termos de *F-Measure*

#Exp.	Fator (A) Indexação	Fator (B) Algoritmo	Fator (C) Atributo	*F-Measure*
1	Soundex	Standard Blocking	Given Name	$0,7189 \pm (0,0115)$
2	Suffix	Standard Blocking	Given Name	$0,6713 \pm (0,0179)$
3	Soundex	Sorted Neighborhood	Given Name	$0,7235 \pm (0,0110)$
4	Suffix	Sorted Neighborhood	Given Name	$0,7343 \pm (0,0141)$
5	Soundex	Standard Blocking	Address2	$0,3979 \pm (0,0320)$
6	Suffix	Standard Blocking	Address2	$0,3835 \pm (0,0311)$
7	Soundex	Sorted Neighborhood	Address2	$0,4306 \pm (0,0308)$
8	Suffix	Sorted Neighborhood	Address2	$0,4057 \pm (0,0308)$

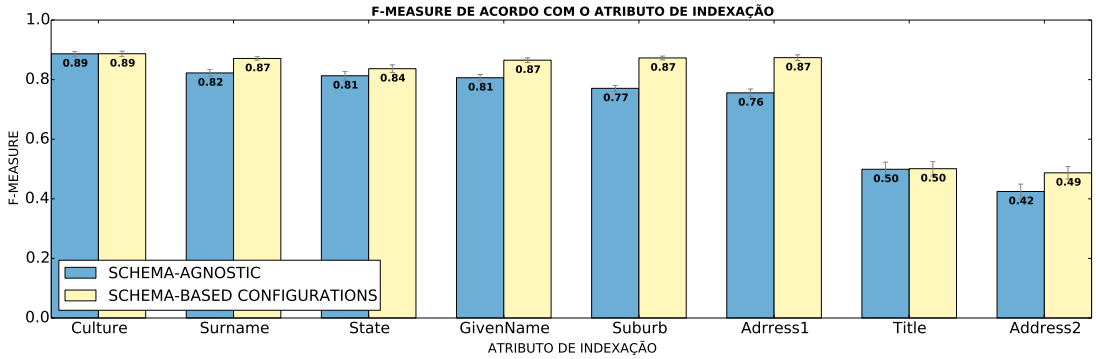
três), e a diferença é apenas $0,0158$ de *F-Measure*. O mesmo acontece com as funções de indexação *Soundex* e *Suffix*, porque nos experimentos *um* e *dois*, e *três* e *quatro*, os resultados são eficazes em termos de *F-Measure* quando comparado com os demais resultados. Por outro lado, os piores resultados da deduplicação são alcançados quando o atributo *Given Name* é trocado pelo atributo *Address2* em qualquer combinação (experimentos *cinco*, *seis*, *sete* e *oito*). Dessa forma, conclui-se que a eficácia do processo de deduplicação de dados está também relacionada com a escolha do atributo de indexação que é *combinado* com as outras funções, pois os resultados sofrem variações significantes quando o atributo é mudado em ambas as técnicas.

6.2 Análise do Atributo de Indexação na Deduplicação de Dados

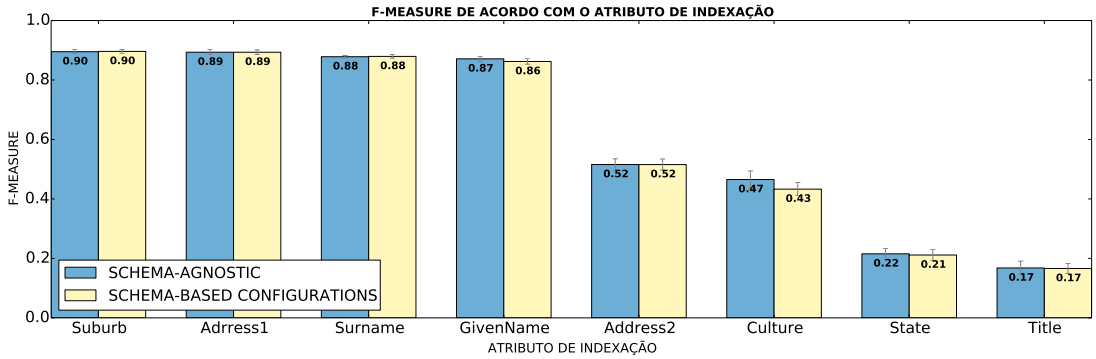
Esta seção apresenta uma análise do impacto do atributo de indexação sobre as etapas do processo de deduplicação de dados. Especificamente, o atributo de indexação é analisado nos métodos *Schema-Agnostic* e *Schema-Based Configurations* na Seção 6.2.1. Por fim, a Seção 6.2.2 avalia o atributo de indexação nos algoritmos *Standard Blocking* e *Sorted Neighborhood*.

6.2.1 Análise do Atributo de Indexação nas Abordagens *Schema-Agnostic* e *Schema-Based Configurations*

Na etapa de indexação, um critério de chave de bloco é atribuído a cada registro do banco de dados. Uma das formas para definir as chaves de bloco é utilizar o valor dos atributos como BK. Outra maneira é aplicar uma regra de codificação nos valores dos atributos e definir a chave, por exemplo, a codificação *Soundex*. Essas duas técni-



(a) *F-Measure* do atributo de indexação no algoritmo *Standard Blocking*



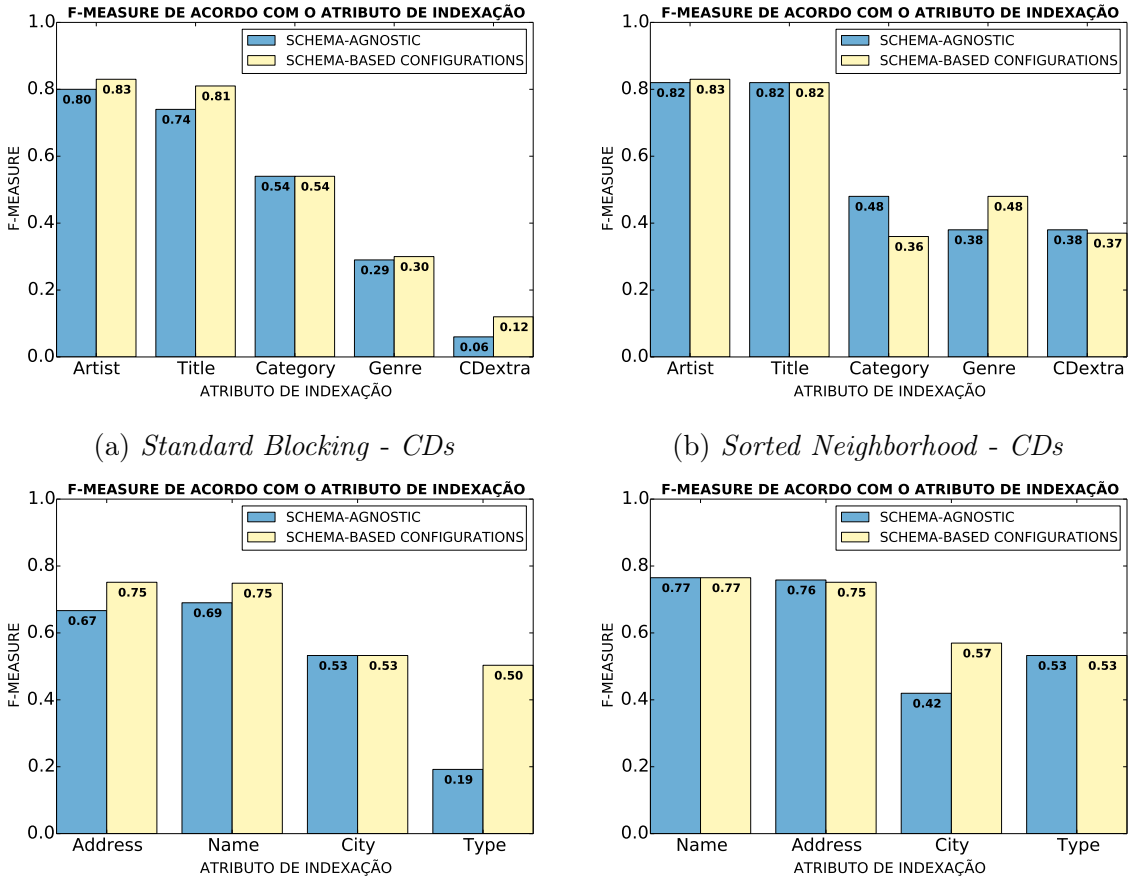
(b) *F-Measure* do atributo de indexação no algoritmo *Sorted Neighborhood*

Figura 6.2: Avaliação do atributo de indexação nos métodos *Schema-Agnostic* e *Schema-Based Configurations* no conjunto de dados *DatasetSynthetic*

cas de indexação são conhecidas por *Schema-Agnostic* e *Schema-Based Configurations* respectivamente e são amplamente utilizadas na deduplicação.

Nesse contexto, esta seção analisa o impacto do atributo de indexação nos dois métodos de indexação. Os experimentos são executamos considerando os dois algoritmos de clusterização *Standard Blocking* e *Sorted Neighborhood*. Especificamente, o objetivo é responder à questão de pesquisa **Q2**: O mesmo atributo de indexação é eficiente e eficaz nas abordagens de indexação *Schema-Agnostic* e *Schema-Based Configurations*? No quesito eficácia, considera-se a métrica *F-Measure*. Para a eficiência, calcula-se o tempo médio para executar o processo de deduplicação de dados nos métodos *Agnostic* e *Configurations-Based*.

Para responder **Q2** no quesito eficácia, nós executamos experimentos em dados sintéticos e reais variando o atributo e o método de indexação. Os conjuntos de dados utilizados são *CDs*, *Restaurant* e *DatasetSynthetic*. Nós utilizamos o valor do atributo como chave de bloco ou a codificação *Soundex* sobre os valores dos atributos. A Figura 6.2(a)-(b) apresenta os resultados nos dados sintéticos e a Figura 6.3(a)-(d) nos



(a) *Standard Blocking - CDs* (b) *Sorted Neighborhood - CDs*

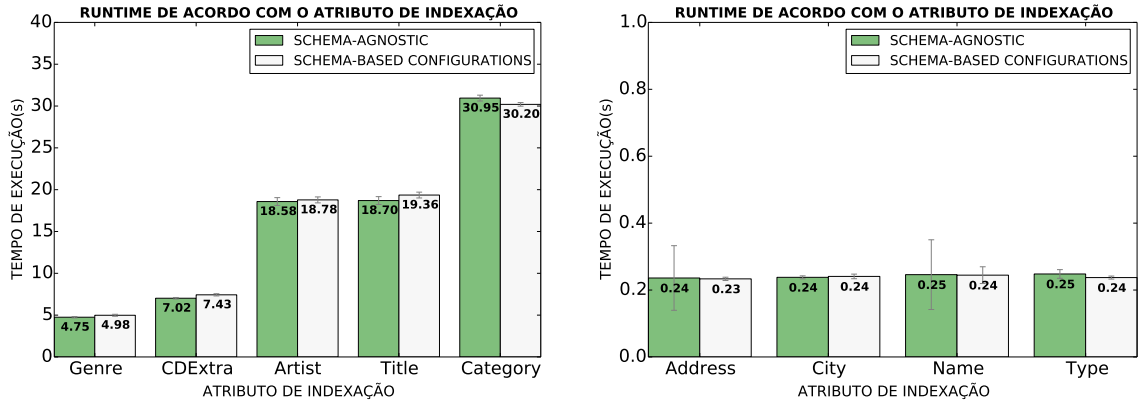
(c) *Standard Blocking - Restaurant* (d) *Sorted Neighborhood - Restaurant*

Figura 6.3: Avaliação do atributo de indexação nos métodos *Schema-Agnostic* e *Schema-Based Configurations* em dados reais

dados reais. Em ambas, o eixo-X exibe os atributos de indexação e o eixo-Y o valor de *F-Measure* em cada configuração de indexação para cada atributo.

Considerando os dados sintéticos (Figura 6.2), os resultados demostram que os atributos mais eficazes são os mesmos em ambas técnicas de indexação. Por exemplo, o atributo *Culture* está entre os mais eficazes nos métodos *Agnostic* e *Configurations-Based* para o algoritmo *Standard Blocking*. Da mesma forma, o atributo *Suburb* no algoritmo *Sorted Neighborhood*. Além disso, os atributos ineficazes não melhoram seus resultados quando os métodos de indexação são mudados (e.g., atributos *State* e *Title* no algoritmo de vizinhança).

Do mesmo modo, os resultados são semelhantes em termos de *F-Measure* nos conjuntos de dados reais em sua maioria. Analisando a Figura 6.3(a), por exemplo, o atributo *Artist* é o mais eficaz em termos de *F-Measure* tanto em *Schema-Agnostic* quanto em *Shema-Based Configurations*. Além disso, comparando os algoritmos de



(a) *Standard Blocking* - CDs (b) *Standard Blocking* - Restaurant

Figura 6.4: Tempo de execução do atributo de indexação nos métodos *Schema-Agnostic* e *Schema-Based Configurations*

blocos e vizinhos, os atributos mais eficazes mantêm a eficácia em ambos algoritmos de clusterização com os dois métodos de indexação. Por exemplo, o atributo *Name* é eficaz no algoritmo *Standard Blocking* e no algoritmo *Sorted Neighborhood*, independente da configuração de indexação utilizada, conforme Figura 6.3(c)-(d).

Em seguida, para abordar **Q2** em eficiência, nós computamos o tempo médio de cada atributo de indexação para executar a deduplicação utilizando os métodos *Agnostic* e *Configurations-Based*. Os experimentos são realizados nos conjuntos de dados *CDs* e *Restaurant*. A Figura 6.4 apresenta os resultados, onde o eixo-X exibe os atributos de indexação e o eixo-Y o tempo médio de execução de cada atributo de indexação nos dois métodos. O intervalo de confiança também é exibido. Os resultados mostram que o tempo médio de execução do processo de deduplicação de dados é estatisticamente igual nos métodos *Agnostic* e *Configuratinos-Based* (considerando um intervalo de confiança de 95%), tanto no conjunto de dados de *CDs* quanto em *Restaurant*. Por exemplo, em *CDs*, o tempo de *Artist* é $18,58s \pm (0,47)$ no método *Agnostic* e $18,77s \pm (0,35)$ em *Based Configurations*. Esses valores são estatisticamente iguais com uma confiança de 95%.

No geral, conclui-se que as técnicas de indexação *Schema-Agnostic* e *Schema-Based Configurations* têm resultados similares em termos de *F-Measure* e tempo médio de execução, tanto em dados sintéticos quanto em reais, porque a diferença de *F-Measure* é insignificante. Ademais, os atributos mais eficazes nos conjuntos de dados apresentados mantêm a eficácia independente do método de indexação escolhido. Por fim, as abordagens *Agnostic* e *Configurations-Based* têm resultados significativos em termos de *F-Measure* nos algoritmos de clusterização por blocos e vizinhos.

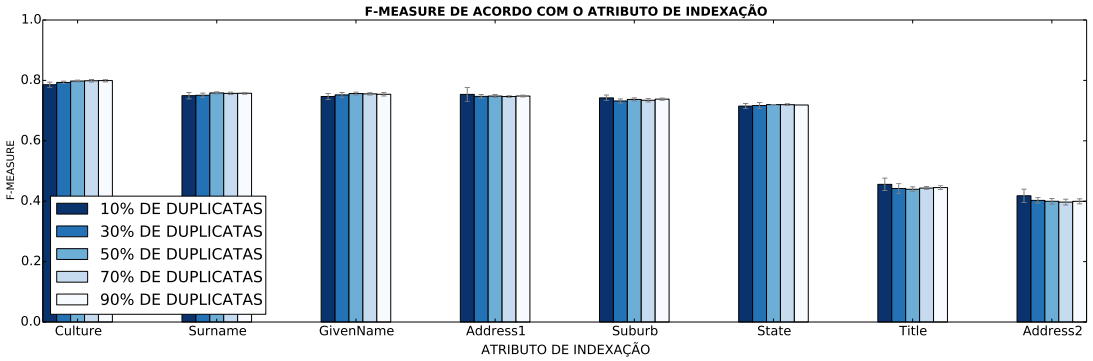
6.2.2 Avaliação do Atributo de Indexação nos Algoritmos *Standard Blocking* e *Sorted Neighborhood*

Existem diversas formas para agrupar os registros na segunda etapa do processo de deduplicação. Via de regra, as instâncias são clusterizadas por vizinho mais próximo, ou a clusterização é baseada no valor de chave de bloco dos registros. Nesta seção, o objetivo é responder a questão de pesquisa **Q3**: O processo de deduplicação de dados é eficaz utilizando o mesmo atributo de indexação nos algoritmos de clusterização *Standard Blocking* e *Sorted Neigborhood*?

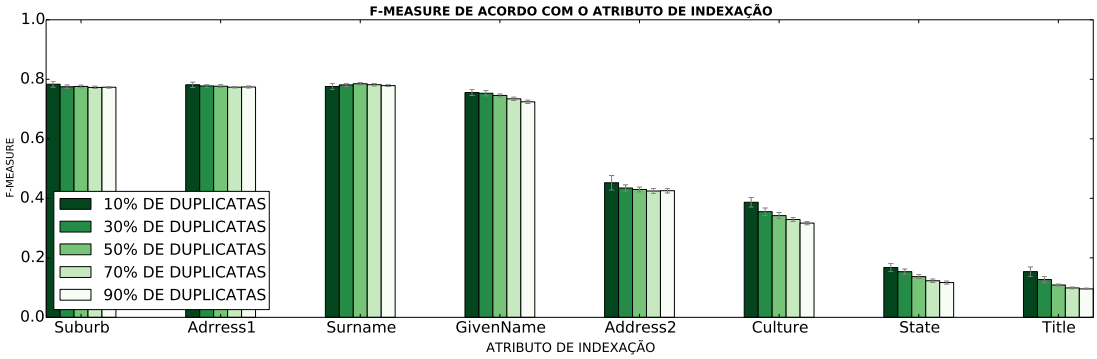
Para responder **Q3**, são utilizados dados sintéticos e reais. Nos sintéticos, variamos o total de registros duplicados em cada conjunto de dados (10% - 90%). O objetivo é analisar se os atributos eficazes mantêm a eficácia mesmo quando o conjunto de dados possui muitos registros duplicados (e.g., 90%). A Figura 6.5(a)-(b) apresenta os resultados em termos de *F-Measure* para cada atributo de indexação nos dois algoritmos. Os resultados estão agrupados por atributo de indexação (eixo-X). O eixo-Y exibe o valor de *F-Measure*, e as cores das barras representam o conjunto de dados utilizado em cada experimento. Note que o intervalo de confiança também é exibido.

Os resultados demostram que: (i) atributos eficazes mantêm a eficácia em ambos algoritmos independente da quantidade de duplicatas no *dataset*, porque os valores de *F-Measure* são similares em todos os conjuntos de dados (e.g., atributo *Culture* em *Standard Blocking* e atributo *Suburb* em *Sorted Neighborhood*); e (ii) atributos ineficazes o são em todos os conjuntos de dados, porque os baixos valores de *F-Measure* são mantidos quando o *dataset* muda (e.g., atributo *Address2* no algoritmo *Standard Blocking* e *Title* no algoritmo *Sorted Neighborhood*). Além disso, comparando os atributos de indexação entre os algoritmos, os resultados demonstram que alguns atributos de indexação são eficazes em um algoritmo e no outro não. Por exemplo, o atributo *Culture* é eficaz no algoritmo *Standard Blocking* e ineficaz no algoritmo *Sorted Neighborhood*. O atributo *State* é eficaz na abordagem de blocos e ineficaz no método de vizinhança. No entanto, há atributos que possuem resultados significantes em termos de *F-Measure* em ambos algoritmos (e.g., *Given Name*, *Surname* e *Address1*).

Em seguida, nós avaliamos o atributo de indexação em conjuntos de dados reais. O objetivo é verificar se os resultados contendo dados sintéticos refletem nos dados reais, ou seja, averiguar se a escolha do atributo de indexação e do algoritmo de clusterização afeta os valores de *F-Measure* em dados reais. Dessa forma, nós analisamos o atributo de indexação nos conjuntos de dados *CDs* e *Restaurant*. A Figura 6.6(a)-(b) apresenta os resultados onde o eixo-X representa o atributo de indexação, e o eixo-Y exibe o valor de *F-Measure* para cada atributo em cada algoritmo.



(a) *F-Measure* do atributo de indexação no algoritmo *Standard Blocking*

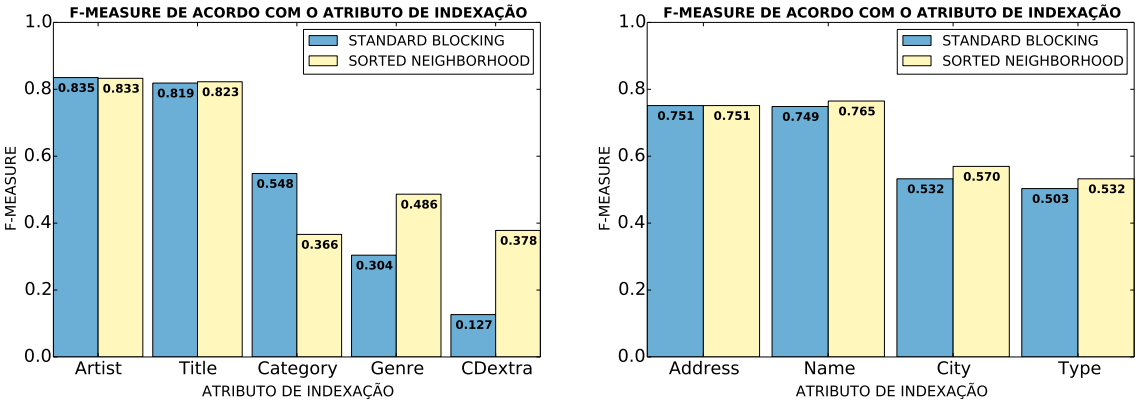


(b) *F-Measure* do atributo no algoritmo *Sorted Neighborhood*

Figura 6.5: Avaliação do atributo de indexação nos conjuntos de dados sintéticos

Semelhante aos dados sintéticos, os resultados com dados reais mostram que os atributos mais eficazes possuem valores similares de *F-Measure* em ambos algoritmos de clusterização. Por exemplo, no conjunto de dados *CDs*, os atributos mais eficazes têm *F-Measure* significativas em ambos os algoritmos (e.g., *Artist* e *Title*). Por outro lado, os atributos ineficazes possuem variações significantes em termos de *F-Measure* dependendo do algoritmo utilizado (e.g., *CDextra*). Considerando o conjunto de dados *Restaurant*, os resultados são semelhantes, onde os atributos mais eficazes são eficazes em ambos algoritmos (e.g., *Address* e *Name*).

No geral, conclui-se que a escolha do atributo de indexação e do algoritmo de clusterização altera os resultados do processo de deduplicação de dados porque os valores *F-Measure* têm variações significantes dependendo da combinação escolhida. Ademais, o atributo de indexação possui maior impacto no processo quando comparado com o algoritmo de clusterização. Por exemplo, no *dataset* com 10% de registros duplicados, o atributo *Culture* difere em cerca de $0,399$ em termos de *F-Measure* nos dois algoritmos. Assim, os resultados de eficácia da deduplicação de dados são prejudicados em torno de 40% quando o atributo *Culture* é combinado com o algoritmo *Sorted Neighborhood*.



(a) CDs (b) Restaurant

Figura 6.6: Avaliação do atributo de indexação nos conjuntos de dados reais

Desse modo, a tarefa de escolher o atributo de indexação e o método de clusterização correto é fundamental para o processo de deduplicação de dados como um todo. Por fim, essa conclusões aplicam-se tanto a dados sintéticos quanto a reais.

6.3 Avaliação do Método *3DR-Indexing*

Conforme apresentado nos resultados da Seção 6.2, o processo de deduplicação de dados é mais eficaz quando se escolhe o atributo de indexação mais apropriado. Além disso, a deduplicação é prejudicada quando um atributo ineficaz é combinado com os demais algoritmos, ainda que eficazes. Nesse contexto, esta seção apresenta os resultados da avaliação do método *3DR-Indexing*. O objetivo é analisar se o método proposto seleciona os melhores atributos para a etapa de indexação. Especificamente, nós avaliamos a questão **Q4**: Nossa solução encontra os melhores atributos para a etapa de indexação (baseada em características extraídas dos valores dos atributos)?

Para abordar **Q4**, a avaliação experimental considera a eficiência e a eficácia de cada atributo de indexação em dados sintéticos e reais. Nos sintéticos, variamos o total de registros duplicados e o total instâncias em cada *dataset*. Os experimentos são executados com os algoritmos *Soundex*, *Standard Blocking* e *Jaro Winkler* em cada uma das etapas do processo respectivamente. A seguir, a Seção 6.3.1 apresenta o *trade-off* entre as métricas de eficiência e eficácia. O *3DR-Indexing* é avaliado em dados sintéticos e reais nas Seções 6.3.2 e 6.3.3 respectivamente. A Seção 6.3.4 analisa o *3DR-Indexing* considerando outras métricas de qualidade. Os *rankings* de atributos são comparados com os *rankings* dos resultados experimentais usando a correlação de *Spearman* na Seção 6.3.5. Por fim, a Seção 6.3.6 compara o *3DR-Indexing* com *baselines*.

6.3.1 *Trade-off* entre Eficiência e Eficácia

A avaliação experimental considera tanto eficiência (tempo de execução do processo de deduplicação de dados) quanto eficácia de cada atributo de indexação (*F-Measure*). Dessa forma, considere $F(a_i)$ o valor de *F-Measure* alcançado por um atributo $a_i \in A_r$ e $F_{max} = \max_{a_i \in A_r} F(a_i)$. Nós consideramos empiricamente como candidatos a melhores atributos de indexação o conjunto $B = \{a_j \in A_r : F_{max} - F(a_j) < 0.1\}$, ou seja, o grupo de atributos que têm uma diferença mínima de 10% em termos de *F-Measure* considerando o maior valor obtido.

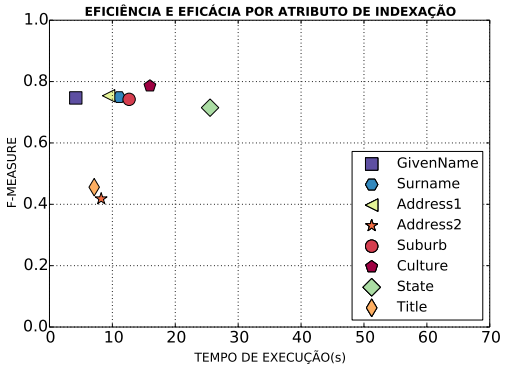
Desse modo, o **melhor atributo de indexação** é definido como o candidato em B que possui o menor tempo de execução. Com essa abordagem, nós garantimos a eficiência sem prejudicar a eficácia do processo de deduplicação. Por exemplo, considerando a Figura 6.3.2(a), o atributo *Culture* tem o maior valor de *F-Measure* (conforme discutido posteriormente). No entanto, outros atributos alcançam valores de *F-Measure* estatisticamente iguais, ou seja, conjunto B (e.g., *Given Name*, *Address1*, *Surname*, *Suburb*), dos quais o atributo *Given Name* é o mais rápido.

6.3.2 Avaliação do *3DR-Indexing* em Dados Sintéticos

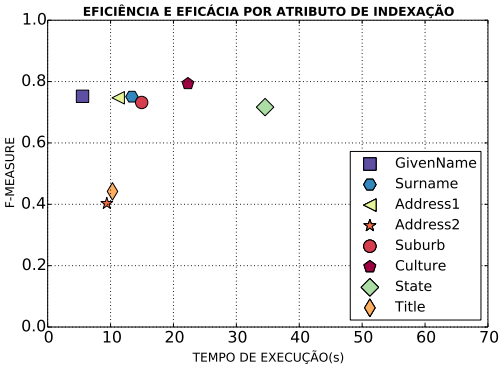
Esta seção apresenta os resultados do método *3DR-Indexing* nos conjuntos de dados sintéticos. Nós analisamos tanto o total de registros duplicados nos *datasets* (10%, 30%, 50%, 70% e 90%) quanto o total de instâncias (10^2 - 10^6) para responder à questão **Q4**. O objetivo é verificar se o método *3DR-Indexing* seleciona os melhores atributos de indexação independente da quantidade de registros duplicados e do tamanho do conjunto de dados.

Dessa forma, nós executamos o processo de deduplicação de dados usando cada atributo separadamente na etapa de indexação. A Figura 6.7 apresenta os resultados em termos de *F-Measure* (eixo-Y) e tempo médio de execução (eixo-X) considerando o total de registros duplicados. Os melhores resultados do processo de deduplicação de dados são localizados na parte superior esquerda de cada gráfico (i.e., atributos de indexação mais rápidos e eficazes). O método proposto tem o objetivo de identificar esses atributos como mais relevantes.

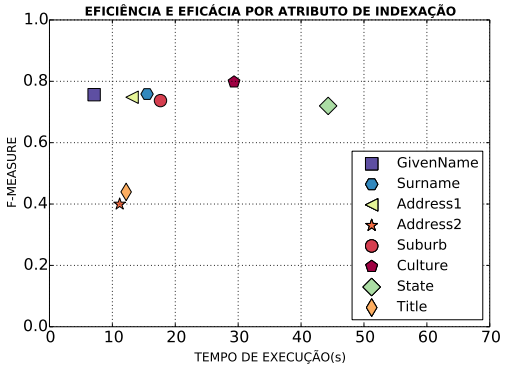
Complementando tais resultados, a Tabela 6.2 exibe o *ranking* de atributos gerados pelo método proposto em cada conjunto de dados. Especificamente, nós calculamos as métricas (*densidade, duplicidade, distintividade e repetição*) para cada atributo dos *datasets* (10%, 30%, 50%, 70% e 90% de registros duplicados) e aplicamos a Equação 4.5. Apenas o valor relevância $R(a)$ é mostrado e está ordenado.



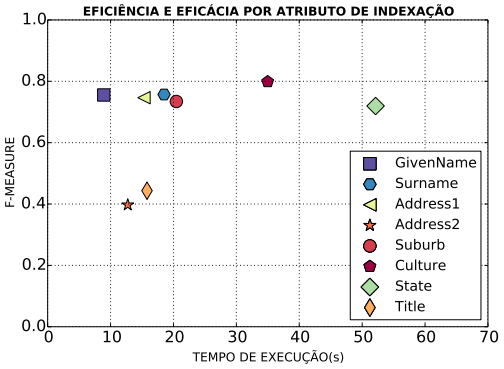
(a) 10% de registros duplicados



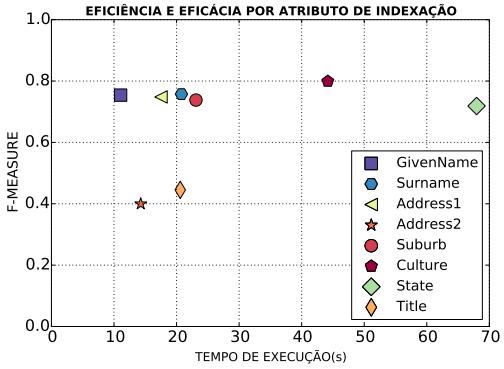
(b) 30% de registros duplicados



(c) 50% de registros duplicados



(d) 70% de registros duplicados



(e) 90% de registros duplicados

Figura 6.7: Avaliação do *3DR-Indexing* de acordo com o total de registros duplicados

Tabela 6.2: *Ranking* de atributos produzidos pelo método *3RD-Indexing* considerando o total de registros duplicados

10% de Duplicatas		30% de Duplicatas		50% de Duplicatas		70% de Duplicatas		90% de Duplicatas	
Atributo	**R(a)**	**Atributo**	**R(a)**	**Atributo**	**R(a)**	**Atributo**	**R(a)**	**Atributo**	**R(a)**
GivenName	1.0000	GivenName	1.0000	GivenName	1.0000	GivenName	1.0000	GivenName	1.0000
Address1	0.9553	Address1	0.9578	Address1	0.9515	Address1	0.9517	Surname	0.9522
Suburb	0.9269	Suburb	0.9370	Suburb	0.9460	Suburb	0.9466	Suburb	0.9520
Surname	0.9184	Surname	0.9331	Surname	0.9443	Surname	0.9447	Address1	0.9470
State	0.8118	State	0.8908	State	0.9272	State	0.9369	State	0.9452
Culture	0.6709	Culture	0.6783	Culture	0.6816	Culture	0.6844	Culture	0.6883
Address2	0.5670	Address2	0.5716	Address2	0.5842	Address2	0.5820	Address2	0.5920
Title	0.3037	Title	0.4273	Title	0.4345	Title	0.4784	Title	0.5007

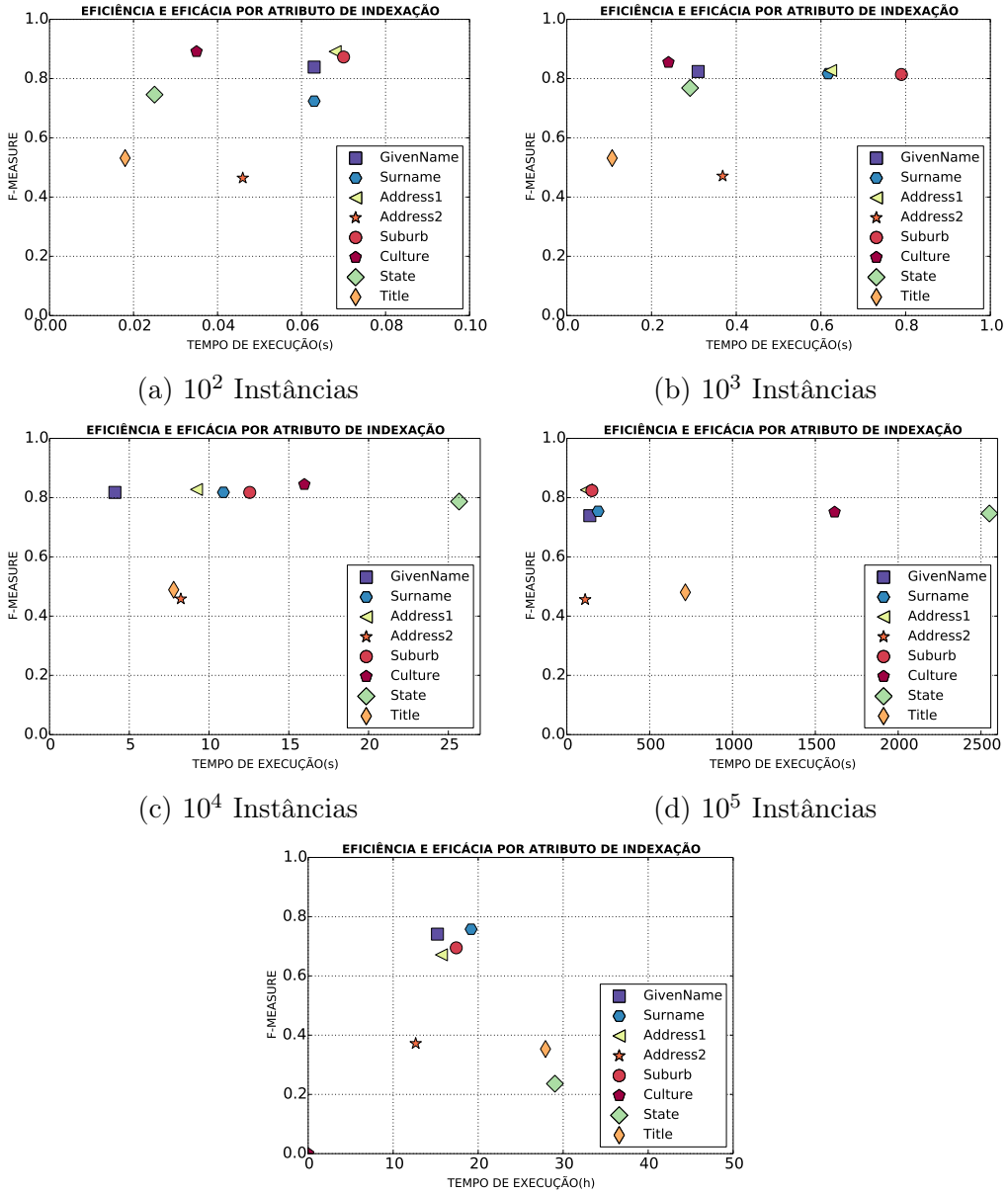
No quesito total de registros duplicados, observa-se que o método proposto seleciona os melhores atributos de indexação em todos os conjuntos de dados, porque os atributos mais relevantes são aqueles com os melhores resultados na deduplicação de dados (Figura 6.7). Por exemplo, em todos os conjuntos de dados, o melhor atributo de indexação é *Given Name* (i.e., eficaz e mais rápido). Esse atributo é o mais relevante em todos os *datasets* conforme Tabela 6.2.

Analisando o *dataset* com 10% de duplicatas, por exemplo, o atributo *Culture* é o mais eficaz com $0,7857 \pm (0,0084)$. Porém, ele tem o segundo maior tempo de deduplicação - $15,969s \pm (0,515)$. Comparando *Given Name* e *Culture* em termos de *F-Measure*, *Culture* é superior em aproximadamente $0,039$, diferença insignificante para o processo de deduplicação. Por outro lado, *Given Name* é eficaz e quase quatro vezes mais rápido que o atributo *Culture*. O atributo *Given Name* tem uma *F-Measure* de $0,7467 \pm (0,0101)$ e um tempo de execução de $4,184s \pm (0,0350)$. *Given Name* é quase duas vezes mais eficaz que *Address2* e seis vezes mais rápido que *State*. Dessa forma, o atributo *Given Name* é o melhor atributo para indexar considerando o *trade-off* de eficiência e eficácia. Nos outros conjuntos de dados, os resultados são similares.

Avaliando os piores atributos de indexação (*Title* e *Address2*), que são rápidos mas ineficazes, o método proposto classifica-os como menos relevantes em todos os conjuntos de dados. Assim, conclui-se que o método *3DR-Indexing* seleciona atributos eficientes e eficazes em *datasets* com poucos registros duplicados (i.e., 10%) e para aqueles com muitos registros duplicados (i.e., 90%), porque em todos os conjuntos de dados os atributos mais relevantes apresentam os melhores resultados.

Em seguida, nós avaliamos o *3DR-Indexing* de acordo com o total de instâncias em cada conjunto de dados (10^2 a 10^6 instâncias). O objetivo é verificar se o método proposto escala para grandes *datasets*. A Figura 6.8 apresenta os resultados. Note que as escalas do eixo-X são diferentes porque o objetivo é comparar o melhor atributo em cada gráfico separadamente, em vez de comparar o valor absoluto de cada ponto entre os gráficos. Os quatro primeiros gráficos estão com o eixo-X em segundos e o último em horas. Do mesmo modo, a Tabela 6.3 exibe o *ranking* de atributos para cada conjunto de dados produzido pelo método proposto.

Dessa forma, os resultados mostram que o método proposto seleciona os melhores atributos na maioria dos conjuntos de dados, com exceção daqueles com 10^3 e 10^6 instâncias, onde o melhor atributo de indexação é o segundo melhor ranqueado conforme o método *3DR-Indexing*. Especificamente, no *dataset* com 10^2 instâncias, os atributos mais eficazes são *Culture*, *Suburb* e *Address1*. Entretanto, *Culture* é o mais rápido. No *dataset* com 10^4 instâncias, os atributos *Given Name*, *Address1*, *Surname*, *Suburb* e *Culture* são os mais eficazes. Novamente, a diferença em termos de *F-Measure* entre



(a) 10^2 Instâncias

(b) 10^3 Instâncias

(c) 10^4 Instâncias

(d) 10^5 Instâncias

(e) 10^6 Instâncias

Figura 6.8: Avaliação do *3DR-Indexing* de acordo com o total de instâncias

Tabela 6.3: *Ranking* de atributos produzidos pelo método *3RD-Indexing* considerando o total de instâncias

10^2 **Instâncias**		10^3 **Instâncias**		10^4 **Instâncias**		10^5 **Instâncias**		10^6 **Instâncias**	
Atributo	**R(a)**	**Atributo**	**R(a)**	**Atributo**	**R(a)**	**Atributo**	**R(a)**	**Atributo**	**R(a)**
Culture	1.0000	GivenName	1.0000	GivenName	1.0000	Suburb	1.0000	Suburb	1.0000
GivenName	0.9393	Culture	0.9035	Address1	0.9632	GivenName	0.9945	GivenName	0.9976
Address1	0.8387	Address1	0.8510	Suburb	0.9333	Address1	0.9785	Surname	0.9902
Surname	0.8351	Surname	0.8418	Surname	0.9217	Surname	0.9778	Address1	0.9806
Suburb	0.8323	Suburb	0.8146	State	0.8074	State	0.9040	State	0.9348
State	0.7518	State	0.6571	Culture	0.6707	Culture	0.6588	Culture	0.6563
Address2	0.5983	Address2	0.5520	Address2	0.5723	Address2	0.6039	Address2	0.6049
Title	0.5090	Title	0.3513	Title	0.2831	Title	0.4668	Title	0.5394

os atributos *Given Name* e *Culture* é insignificante. O atributo *Given Name* é o mais rápido e eficaz. No conjunto de dados com 10^5 instâncias, os melhores atributos são *Suburb* e *Address2*, que possuem resultados estatisticamente iguais.

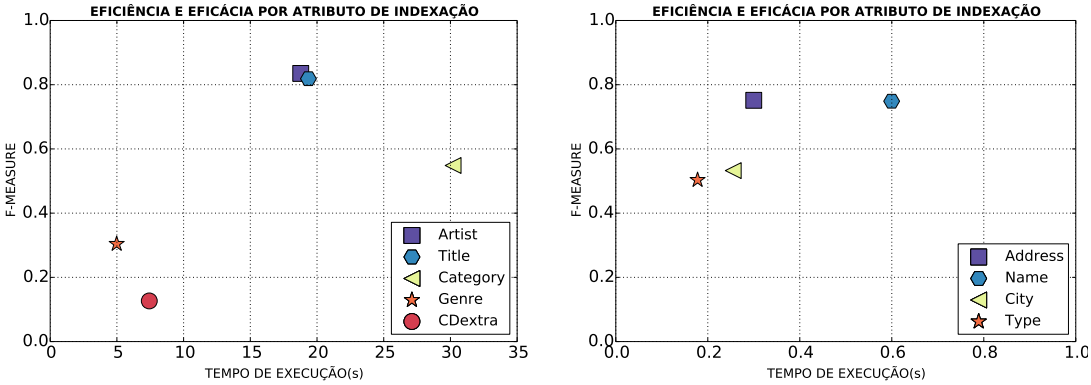
No conjunto de dados com 10^3 instâncias, o método proposto não seleciona o melhor atributo de indexação (*Culture*). Porém, o atributo com segunda maior relevância tem resultados significantes para o processo (*Given Name*). Por exemplo, a diferença entre *Culture* e *Given Name* em termos de *F-Measure* é $0,03$, comparando o tempo de execução, a diferença é $0,07s$, ou seja, os atributos têm diferenças insignificantes. No *dataset* com 10^6, o método proposto mais uma vez não seleciona o melhor atributo (*Given Name*), mas o segundo atributo mais relevante possui resultados significantes em termos de *F-Measure* e tempo médio de execução. Especificamente, o atributo *Suburb* difere de *Given Name* em $0,04$ de *F-Measure* e em aproximadamente duas horas em tempo de execução.

No geral, mesmo quando o método proposto não seleciona o melhor atributo de indexação (i.e., 10^3 e 10^6 instâncias), os resultados da deduplicação são significantes considerando o *trade-off* de eficiência e eficácia com os atributos mais relevantes. Além disso, os piores atributos são ranqueados como menos relevantes pelo método proposto em todos os datasets (*Title* e *Address2*). O atributo *Address2* é rápido mas ineficaz, enquanto que *Title* é ineficaz em todos os *datasets*. No conjunto de dados com 10^6 instâncias, existem atributos que consomem mais de um dia de processamento para executar a deduplicação (*State*), enquanto que outros gastam aproximadamente 15 horas (*Given Name* e *Address1*). Tal resultado destaca a importância de selecionar os melhores atributos para a etapa de indexação. Por fim, os resultados indicam que o 3*DR-Indexing* é eficiente e eficaz em conjuntos de dados com diferentes escalas.

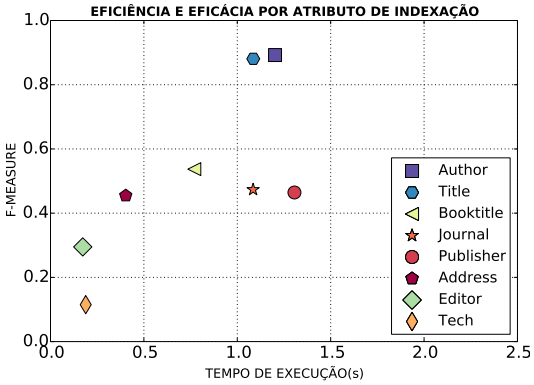
6.3.3 Avaliação *3DR-Indexing* em Dados Reais

Esta seção avalia o método proposto em dados reais. Como já mencionado, nós usamos os conjuntos de dados *Cora*, *CDs* e *Restaurant* porque eles possuem domínios distintos e são amplamente utilizados em experimentos de deduplicação de dados. O objetivo é abordar **Q4** e testar se o método proposto seleciona os atributos mais eficientes e eficazes em dados reais com diferentes domínios. A Figura 6.9 apresenta os resultados em termos de *F-Measure* (eixo-Y) e tempo médio de execução (eixo-X). Da mesma forma, a Tabela 6.4 mostra o *ranking* de atributos gerado pelo método proposto para cada conjunto de dados.

Os resultados mostram que o *3DR-Indexing* também identifica os atributos mais relevantes nos conjuntos de dados reais. Por exemplo, no conjunto de dados de *CDs*,



(a) Conjunto de dados CDs (b) Conjunto de dados Restaurant



(c) Conjunto de dados Cora

Figura 6.9: Avaliação do *3DR-Indexing* em conjuntos de dados reais

Tabela 6.4: *Ranking* de atributos produzidos pelo *3RD-Indexing* em dados reais

CDs		**Restaurant**		**Cora**	
Atributo	**R(a)**	**Atributo**	**R(a)**	**Atributo**	**R(a)**
Artist	1.0000	Type	1.0000	Author	1.0000
Genre	0.9711	Address	0.9123	Title	0.9567
Title	0.8884	Name	0.9120	BookTitle	0.6505
Category	0.8594	City	0.8207	Address	0.5706
CDExtra	0.6491	-	-	Editor	0.4922
-	-	-	-	Publisher	0.4798
-	-	-	-	Tech	0.3668
-	-	-	-	Journal	0.2507

o atributo mais relevante é *Artist* (Tabela 6.4) que possui o melhor resultado na deduplicação na Figure 6.9(a). No conjunto de dados *Restaurant*, o método proposto não identifica o melhor atributo. Entretanto, o segundo atributo mais relevante possui resultados significantes em termos de eficiência e eficácia (*Address*). No *dataset Cora*, os atributos mais relevantes possuem o melhor resultado na deduplicação (*Author* e *Title*), mas não necessariamente o menor tempo. Dessa forma, conclui-se que o método proposto apresenta resultados significantes em dados reais de diferentes domínios,

porque os atributos mais relevantes possuem os melhores resultados da deduplicação de dados (considerando o *trade-off* de eficiência e eficácia). Os resultados apontam que o método *3DR-Indexing* é adaptável para qualquer conjunto de dados relacional, novamente enfatizando a vantagem sobre as abordagens de aprendizagem de máquina que necessitam de dados para treino.

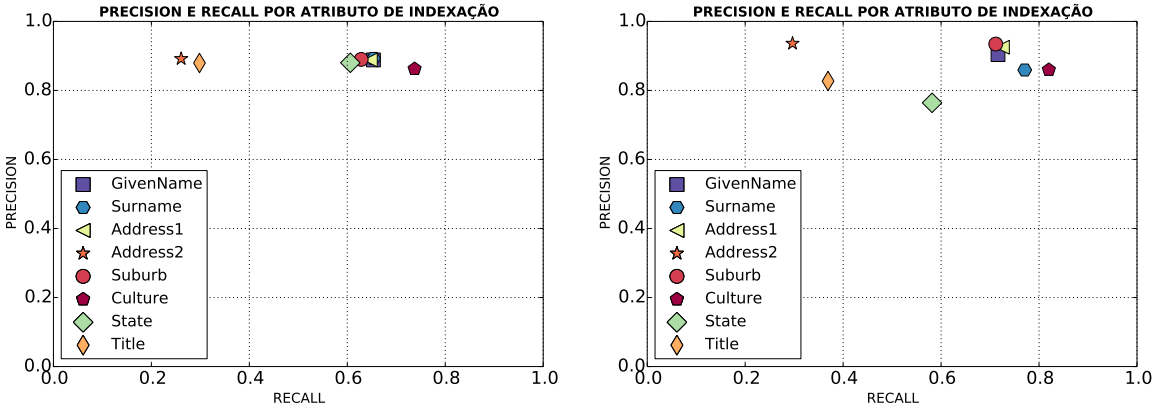
No geral, conclui-se que método o proposto seleciona os atributos mais relevantes na maioria dos conjuntos de dados (i.e., 10 de 13 *datasets*). Além disso, mesmo quando o *3DR-Indexing* falha, ou seja, não classifica o melhor atributo como mais relevante, os atributos de maior relevância têm resultados significantes considerando o *trade-off* de *F-Measure* e tempo de execução. Por fim, destacamos que o *3DR-Indexing* não classifica os piores atributos como mais relevantes, pois os piores atributos de indexação possuem sempre a menor relevância em todos os conjuntos de dados.

6.3.4 Avaliação do *3DR-Idexing* Considerando outra Perspectiva de Qualidade

Apenas a métrica *F-Measure* não é suficiente para analisar a qualidade dos algoritmos aplicados no processo de duplicação de dados. Outras métricas típicas de avaliação são Revocação e Precisão. A Revocação é a fração de registros duplicados que são corretamente classificados (VP), ou seja, considerando o total de duplicatas possíveis, quantas são identificadas pelos algoritmos? A Precisão considera tanto os verdadeiros positivos quanto os registros duplicados que foram classificados de forma errônea (FP) e é calculada pela divisão entre os verdadeiros positivos e todos os registros duplicados classificados, incluindo os falsos positivos. Via de regra, o objetivo da deduplicação de dados é identificar o maior número de VP com uma menor quantidade de FP.

Nesse contexto, nós avaliamos as duas métricas de qualidade para cada atributo de indexação nos conjuntos de dados sintéticos e reais. O objetivo é abordar a questão de pesquisa **Q4** e mostrar que os atributos mais relevantes de acordo com o *3DR-Indexing* têm valores de Revocação e Precisão significantes. A Figura 6.10(a)-(b) apresenta os resultados nos conjuntos de dados sintéticos onde o eixo-X exibe a métrica de Revocação e o eixo-Y a Precisão. Dois cenários são considerados: a quantidade de registros duplicados em cada conjunto de dados (10%, 30%, 50%, 70% e 90% de duplicatas) e o total de instâncias nos conjuntos de dados (10^2 - 10^6 instâncias). Dessa forma, nós computamos o valor médio de Revocação e Precisão para cada atributo de indexação em cada cenário separadamente.

Analisando o primeiro cenário na Figura 6.10(a), todos os atributos possuem precisão significantes, porque os valores são superiores a $0,8$ em todos os atributos,



(a) Revocação e precisão média considerando o total de registros duplicados

(b) Revocação e precisão média considerando o total de instâncias

Figura 6.10: Revocação e precisão por atributo de indexação em dados sintéticos

por exemplo. Entretanto, os atributos *State*, *Suburb*, *Given Name*, *Address2* e *Culture* identificam uma maior quantidade de verdadeiros positivos, porque seus valores de Revocação são superiores aos valores dos atributos *Title* e *Address2*. O atributo *Culture* tem o melhor *trade-off* entre Revocação e Precisão, pois identifica uma maior quantidade de registros duplicados corretamente com uma menor quantidade de falsos positivos. Para o segundo cenário na Figura 6.10(b), os melhores atributos em termos de Revocação e Precisão são *Given Name*, *Suburb*, *Address1*, *Surname* e *Culture*, porque apresentam o melhor *trade-off* entre as métricas. Por exemplo, os atributos *Given Name*, *Address* e *Suburb* identificam mais de 70% dos registros duplicados e aproximadamente 90% das classificações estão corretas.

Em seguida, analisamos ambas métricas para cada atributo de indexação nos conjuntos de dados reais. A Tabela 6.5 exibe a Revocação, Precisão, *F-Measure* e o Tempo de Execução médio para cada conjunto de dados (os valores estão ordenados considerando o melhor *trade-off* entre *F-Measure* e Tempo de Execução). Do mesmo modo que nos dados sintéticos, os atributos que possuem melhor Revocação têm melhor *trade-off* entre as duas métricas em todos os conjuntos de dados reais (i.e., os dois primeiros atributos de cada conjunto de dados). Por exemplo, no conjunto de dados *CDs*, os atributos *Artist* e *Title* identificam aproximadamente 70% dos registros duplicados com uma precisão de mais de 95%. No *dataset Restaurant*, os atributos *Name* e *Address* têm melhor *trade-off* em termos de Revocação e Precisão. No conjunto de dados *Cora*, os atributos *Author* e *Title* também possuem os melhores resultados.

No geral, observa-se que a maioria dos atributos têm resultados significantes em termos de Precisão, porque todos os valores são iguais ou superiores a $0,9$. Entretanto,

Tabela 6.5: Revocação, Precisão, *F-Measure* e Tempo Médio de execução de acordo com o atributo de indexação em dados reais

Conjunto de Dados *CDs*				
Atributo de Indexação	**Revocação**	**Precisão**	***F-Measure***	**Tempo de Execução**
*Artist	0,73	0,98	0,83	$18,77 \pm (0,35)$
Title	0,71	0,96	0,81	$19,35 \pm (0,34)$
Category	0,38	0,97	0,54	$30,19 \pm (0,21)$
Genre	0,18	0,98	0,30	$4,98 \pm (0,11)$
CDExtra	0,07	1,00	0,12	$7,42 \pm (0,13)$
Conjunto de Dados *Restaurant*				
Atributo de Indexação	**Revocação**	**Precisão**	***F-Measure***	**Tempo de Execução**
*Address	0,61	0,99	0,75	$0,29 \pm (0,03)$
Name	0,60	1,00	0,74	$0,59 \pm (0,06)$
City	0,37	0,98	0,53	$0,25 \pm (0,01)$
Type	0,34	0,97	0,50	$0,17 \pm (0,01)$
Conjunto de Dados *Cora*				
Atributo de Indexação	**Revocação**	**Precisão**	***F-Measure***	**Tempo de Execução**
*Author	0,87	0,91	0,89	$1,20 \pm (0,10)$
Title	0,86	0,90	0,88	$1,08 \pm (0,14)$
BookTitle	0,37	0,99	0,53	$0,77 \pm (0,06)$
Address	0,29	0,99	0,54	$0,40 \pm (0,05)$
Editor	0,17	1,00	0,29	$0,17 \pm (0,03)$
Publisher	0,31	0,97	0,46	$1,30 \pm (0,10)$
Tech	0,06	1,00	0,11	$0,18 \pm (0,03)$
Journal	0,31	0,95	0,47	$1,08 \pm (0,05)$

(*) atributos de maior relevância de acordo com o método *3DR-Indexing*

existem atributos que possuem os valores de Revocação insignificantes, o que prejudica o *trade-off* entre as métricas. Por exemplo, com exceção dos atributos *Author* e *Title*, os demais atributos do conjuntos de dados *Cora* possuem o valor de Revocação baixo, pois identificam menos de 40% dos registros duplicados. Desse modo, apesar de boa precisão, tais atributos são ineficazes para o processo de deduplicação como um todo, considerando o *trade-off* entre as duas métricas. Por fim, abordando a questão **Q4** (o método proposto seleciona os melhores atributos?), conclui-se que o *3DR-Indexing* seleciona atributos com resultados significantes em termos de Revocação e Precisão, porque os atributos que possuem maior relevância têm o melhor *trade-off* entre as duas métricas, tanto em dados sintéticos quanto em reais.

6.3.5 Avaliação da Correlação entre *Rankings*

Depois das avaliações experimentais, os resultados de cada conjunto de dados podem ser ordenados considerando o atributo de indexação utilizado. Por exemplo, um critério de ordenação é classificar os resultados do melhor atributo indexação para o pior de acordo com o *trade-off* entre as métricas de eficiência e eficácia. Nesse contexto,

esta seção avalia a correlação entre os *rankings* dos resultados experimentais e os *rankings* produzidos pelo *3DR-Indexing*. O objetivo é abordar a questão **Q4** (o método proposto seleciona os melhores atributos para a etapa de indexação?) e mostrar que o *3DR-Indexing* possui a melhor correlação entre os *rankings* produzidos e os resultados experimentais ordenados por atributo de indexação. Além disso, nós analisamos empiricamente outras combinações de métricas contendo as métricas adotadas no Capítulo 4 para destacar que o método proposto agrega a melhor combinação (i.e., *baselines*).

Assim, nós produzimos um *ranking* de atributos para cada conjunto de dados conforme *trade-off* de eficiência e eficácia, ou seja, os atributos são ordenados do melhor para o pior conforme *trade-off* apresentado na Seção 6.3.1. Por exemplo, no conjunto de dados de *Restaurant*, os resultados experimentais mostram que o melhor atributo de indexação é *Address*, o segundo melhor *Name*, o terceiro *Type* e o pior *City*. Nós ordenamos todos os conjuntos com esse critério. Em seguida, nós computamos o coeficiente de correlação de *Spearman* entre o *ranking* dos resultados e o *ranking* da combinação das métricas para cada conjunto de dados.

A correlação de *Spearman* mede a equivalência da ordem entre duas variáveis X e Y e está no intervalo $[-1, 1]$. Quanto mais próximo de 1, maior a correlação entre as variáveis. Quanto mais próximo de -1, significa que existe correlação mas inversa. Por outro lado, uma correlação próxima de 0 significa que não existe correlação entre as variáveis [Cohen, 1988]. A Tabela 6.6 apresenta o coeficiente de correlação de *Spearman* para cada combinação em cada *dataset*. A última linha da tabela exibe o total de conjuntos de dados em que o valor da correlação é considerado **muito alto** para cada combinação. Nós seguimos as convenções de Cohen [1988] para interpretar o coeficiente de correlação: maior que $0,7$ é **muito alto**; no intervalo $[0,5; 0,7)$ é **alto**; em $[0,3; 0,5)$ é **moderado**; no intervalo $[0,1; 0,3)$ é **pequeno**; e menor que $0,1$ é **inexistente** (esses intervalos também são válidos para a correlação negativa).

Os resultados mostram que o método *3DR-Indexing* tem coeficiente de correlação muito alto em 10 dos 13 *datasets* (*Cora*, 10%, 30%, 50%, 70%, 90%, 10^2, 10^3, 10^4 e 10^5). No conjunto de dados de *CDs* e 10^6, o coeficiente é alto. Em *Restaurant*, o coeficiente é moderado. Além disso, as combinações (*Rep + Dup*), (*Dens + Rep*) e (*Dens + Dist*) têm coeficientes muito alto para poucos conjuntos de dados. Na combinação (*Rep + Dist*) o coeficiente é muito alto para sete *datasets*. Para a combinação (*Rep + Dist + Dup*), o coeficiente é muito alto para a maioria dos *datasets*. No entanto, comparando-a com o método *3DR-Indexing*, o método proposto possui coeficiente menor apenas no conjunto de dados (10ˆ6), igual no *dataset Restaurant* e superior para os demais.

No geral, conclui-se que o método proposto seleciona os melhores atributos de indexação porque o coeficiente de correlação de *Spearman* é muito alto em 10 dos

Tabela 6.6: Coeficiente de correlação de *Spearman* entre o *ranking* dos resultados e o *ranking* produzido pela combinação das métricas em cada conjunto de dados

Datasets	**Combinação de Métricas - *Baselines***					
	Rep+Dup	**Rep+Dist**	**Dens+Rep**	**Dens+Dist**	**Rep+Dist+Dup**	***3DR-Indexing***
CDs	0,60	-0,10	0,70	0,50	-0,10	0,70
Restaurant	0,80	0,40	0,80	0,40	0,40	0,40
Cora	0,16	0,76	0,80	0,92	0,80	0,83
10%	0,40	0,88	0,66	0,54	0,85	0,92
30%	0,40	0,66	0,66	0,54	0,85	0,92
50%	0,38	0,66	0,66	0,42	0,76	0,92
70%	0,38	0,66	0,66	0,54	0,66	0,95
90%	0,45	0,64	0,64	0,40	0,64	0,90
10^2	0,40	0,83	0,54	0,71	0,83	0,90
10^3	0,47	0,88	0,47	0,76	0,88	0,92
10^4	0,40	0,88	0,64	0,54	0,85	0,92
10^5	0,42	0,78	0,69	0,59	0,78	0,97
10^6	0,71	0,83	0,78	0,42	0,85	0,66
Muito Alta	2	7	3	3	9	10

Tabela 6.7: Valor máximo de *F-Measure* em dados reais: *3DR-Indexing* vs. *Baselines*

Estudos	**Conjunto de Dados**		
	Cora	***Restaurant***	***CDs***
[Christen, 2012b]	*0,55	*0,20	-
[McCallum et al., 2000]	*0,78	-	-
[Hand & Christen, 2018]	*0,80	-	-
[Papadakis et al., 2015]	*0,83	-	-
[Bilenko & Mooney, 2003]	*0,86	0,92	-
[Su et al., 2010]	0,92	-	-
[Levin & Heuser, 2010]	0,94	-	-
[Chen et al., 2012]	0,95	0,91	-
[Cohen & Richman, 2002]	0,96	1,00	-
[Draisbach et al., 2012]	0,97	-	-
[Canuto et al., 2013]	0,97	-	-
[Leitão et al., 2007]	-	-	*0,68
[Draisbach & Naumann, 2009]	-	-	0,87
3DR-Indexing	0,89	0,83	0,75

(*) Método *3DR-Indexing* é superior em termos de *F-Measure*

13 conjuntos de dados. Comparando o *3DR-Indexing* com as outras combinações, o método proposto tem coeficientes superiores na maioria dos *datasets*. Por fim, observa-se que à medida em que agregam-se novas métricas a uma combinação, o número de *datasets* com coeficientes muito alto aumenta. Por exemplo, a combinação (*Rep+Dist*) tem coeficiente muito alto para sete conjuntos de dados, (*Rep+Dist+Dup*) para nove e o método *3DR-Indexing*, que considera quatro métricas, em dez *datasets*.

6.3.6 Comparação do Método *3RD-Indexing* com *Baselines*

Diversas soluções têm sido propostas para o problema da deduplicação de dados. Algumas são específicas para determinados domínios e outras têm resultados significantes em vários contextos.

Nesse sentido, esta seção compara o método *3DR-Indexing* com *baselines*. Para tal, nós selecionamos estudos que utilizam nos experimentos os mesmos *datasets* das nossas avaliações. Em seguida, nós verificamos o maior valor de *F-Measure* informado nos trabalhos para os *datasets*. Alguns estudos disponibilizam apenas valores de Revocação e Precisão. Assim, nós computamos o valor de *F-Measure* com os valores informados. Em outros, o valor de *F-Measure* é apresentado apenas na forma gráfica, sem a precisão de cada ponto. Desse modo, nós aproximamos o valor máximo de *F-Measure* por meio da inspeção visual em cada gráfico. A maior parte dos trabalhos utiliza o conjunto de dados do *Cora* nos experimentos. Outros estudos utilizam dois dos *datasets* (*Cora* e *Restaurant*) e alguns apenas o conjunto de dados de *CDs*. A Tabela 6.7 apresenta o valor máximo de *F-Measure* obtido para cada estudo.

Analisando o conjunto de dados *Cora*, o método proposto tem valor de *F-Measure* superior aos estudos apresentados por Bilenko & Mooney [2003], Christen [2012b], Hand & Christen [2018], McCallum et al. [2000] e Papadakis et al. [2015]. Para o conjunto de dados *Restaurant*, o *3DR-Indexing* é superior ao estudo apresentado por Christen [2012b]. Segundo o autor, o valor de *F-Measure* é insignificante porque o *dataset* contém muitos erros e variações. No entanto, nosso estudo apresenta resultados eficazes no mesmo *dataset*, bem como trabalhos anteriores que analisam o mesmo conjunto de dados [Bilenko & Mooney, 2003; Chen et al., 2012; Cohen & Richman, 2002]. Em *CDs*, o método proposto é superior ao trabalho de Leitão et al. [2007]. Ademais, é importante destacar que alguns estudos são específicos para o domínio de bibliotecas digitais, o que justifica os altos valores de *F-Measure* [Canuto et al., 2013; Levin & Heuser, 2010]. Tais estudos não são adaptáveis para outros domínios. Por outro lado, o método *3DR-Indexing* apresenta resultados significantes em vários domínios de dados.

Por fim, não comparamos os resultados do *3DR-Indexing* com o trabalho de Canalle et al. [2017], porque os autores não exibem o resultado da deduplicação considerando o conjunto de dados inteiro do *Cora*. Ao invés disso, os autores criam subconjuntos com o *Cora* e executam experimentos em cada subconjunto de forma separada. Porém, nós comparamos os resultados com o trabalho completo dos autores[10], onde é informado um valor máximo de *F-Measure* de 0.60 para o *dataset* inteiro do Cora (especificamente na Figura 5.5 da página 71). O método *3DR-Indexing* obtém um *F-Measure* de 0.89, ou seja, nosso método é aproximadamente 40% mais eficaz que o método proposto pelos autores.

No geral, os resultados mostram que o desempenho do nosso método é comparável com os trabalhos relacionados porque os valores em termos de *F-Measure* são próximos.

[10]Dissertação de Mestrado - https://repositorio.ufpe.br/handle/123456789/18362 - Acesso 04/2018

Além disso, apenas os trabalhos de Chen et al. [2012], McCallum et al. [2000] e Su et al. [2010] informam o tempo para deduplicação dos *datasets*. Nosso estudo detecta as duplicatas de *Cora*, *Restaurant* e *CDs* em $1,20$, $0,29$ e $18,77$ segundos respectivamente.

6.4 Análise da Combinação de Atributos na Etapa de Indexação

Na etapa de indexação, os blocos de registros podem ser formados com base em um atributo de indexação ou em uma combinação dos mesmos (i.e., um registro recebe uma combinação de chaves de bloco). Nesse sentido, esta seção aborda a questão de pesquisa **Q5**: Os resultados do processo de deduplicação de dados são melhorados com a utilização da combinação de atributos na etapa de indexação?

Para responder **Q5**, executamos o processo de deduplicação de dados utilizando uma combinação de atributos na etapa de indexação. Para tal, os registros do *dataset* recebem várias BKs, uma para cada atributo presente no grupo escolhido. Em seguida, o valor de cada atributo é codificado por meio do algoritmo *Soundex* e, na etapa de clusterização, os grupos são formados pelas instâncias que contêm pelo menos um valor de BK em comum (i.e., um registro pode pertencer a mais de um grupo).

Assim, os experimentos adotam dois critérios: (i) utilizar todas as combinações possíveis entre os atributos de um conjunto de dados; e (ii) combinar os atributos mais relevantes de acordo com o método *3DR-Indexing*. Para o segundo critério, nós avaliamos empiricamente a combinação de grupos formados com os cinco primeiros atributos mais relevantes, ou seja, os grupos são criados pelos dois primeiros, três primeiros, quatro primeiros e cinco primeiros atributos mais relevantes. De forma geral, o objetivo é avaliar a eficiência e a eficácia da combinação de atributos na indexação.

Nesse contexto, a Tabela 6.8 apresenta os valores de Revocação, Precisão e *F-Measure* para cada combinação no conjunto de dados *Restaurant* (i.e., primeiro critério). Esse *dataset* tem quatro atributos: *Name*, *Address*, *City* e *Type*, os quais formam um total de dez combinações distintas com pelo menos dois atributos. Os resultados mostram que a eficácia da deduplicação não tem aumento significativo com a combinação de atributos na indexação. Por exemplo, o valor de *F-Measure* conseguido com quatro atributos é $0,77$ (*Name*, *Address*, *City*, *Type*). No entanto, quando apenas um atributo é utilizado, a *F-Measure* é $0,75$ (*Address* ou *Name*), ou seja, os valores diferem em apenas $0,02$. Tais resultados são explicados pelos valores de Revocação e Precisão, pois não há aumento significativo em ambas as métricas. No geral, os valores de Revocação estão próximos de $0,60$ e a Precisão em torno de $0,99$ (i.e., considerando

Tabela 6.8: Avaliação da eficácia da combinação de atributos no *dataset Restaurant*

Combinação	**Revocação**	**Precisão**	***F-Measure***
Name-Address-City-Type	0,63	0,99	0,77
Name-Address-City	0,63	0,99	0,77
Name-Address-Type	0,63	0,99	0,77
Address-City-Type	0,63	0,99	0,77
Name-Address	0,63	0,99	0,77
Name-City	0,61	0,99	0,75
Name-Type	0,61	0,99	0,75
Address-City	0,63	0,99	0,77
Address-Type	0,62	0,99	0,76
City- Type	0,53	0,98	0,69
Address	0,60	1,00	0,75
Name	0,61	0,99	0,75

Tabela 6.9: Avaliação da eficácia da combinação dos atributos mais relevantes segundo o método *3DR-Indexing* nos *datasets CDs* e *DatasetSynthetic*

Conjunto de Dados *CDs*			
Combinação	**Revocação**	**Precisão**	***F-Measure***
Artist-Genre-Title-Category-CDExtra	0,72	0,96	0,82
Artist-Genre-Title-Category	0,72	0,96	0,82
Artist-Genre-Title	0,72	0,96	0,82
Artist-Genre	0,72	0,96	0,82
Artist	0,72	0,97	0,84
Conjunto de Dados *DatasetSynthetic*			
Combinação	**Revocação**	**Precisão**	**F-Measure**
GivenName-Address1-Suburb-Surname-State	0,89	0,90	0,90
GivenName-Address1-Suburb-Surname	0,89	0,90	0,90
GivenName-Address1-Suburb	0,89	0,90	0,90
GivenName-Address1	0,89	0,91	0,90
GivenName	0,81	0,92	0,87

todas as combinações). Mesmo combinando quatro atributos, o total de duplicatas corretas está em torno de 60%, as quais também são detectadas utilizando apenas um atributo de indexação (*Address*).

A Tabela 6.9 apresenta os valores de Revocação, Precisão e *F-Measure* considerando a combinação dos atributos mais relevantes de acordo com o método *3DR-Indexing* (i.e., segundo critério). Nós utilizamos os conjuntos de dados *CDs* e *Dataset-Sintético*. O primeiro contém cinco atributos: *Artist*, *Title*, *Category*, *Genre*, *CDExtra*. De acordo com o método *3DR-Indexing*, a ordem de relevância dos atributos é: *Artist*, *Genre*, *Title*, *Category*, *CDExtra*. Novamente, os resultados do processo de deduplicação de dados não são melhorados com a combinação dos atributos mais relevantes, porque a *F-Measure* de apenas um atributo é superior aos valores das outras combinações (e.g., atributo *Artist*).

Tabela 6.10: Avaliação do tempo médio de execução da combinação de atributos mais relevantes na etapa de indexação

Conjunto de Dados *CDs*			
Combinação	**#BKs**	**#Pares Candidatos**	**Tempo de Execução(s)**
Artist-Title-Genre-Category-CDExtra	3.963	12.923.632	$69,52 \pm (0,56) \uparrow 3$
Artist-Title-Genre-Category	3.892	11.877.030	$65,64 \pm (0,42) \uparrow 3$
Artist-Title-Genre	3.892	1.428.157	$37,24 \pm (0,65) \uparrow 2$
Artist-Title	3.861	532.186	$33,77 \pm (0,49) \uparrow 2$
Artist	2.750	456.575	$18,77 \pm (0,35)$
Conjunto de Dados *DatasetSynthetic*			
Combinação	**#BKs**	**#Pares Candidatos**	**Tempo de Execução(s)**
GivenName-Address1-Suburb-Surname-State	3.430	12.496.685	$59,76 \pm (0,64) \uparrow 15$
GivenName-Address1-Suburb-Surname	3.423	1.078.646	$29,30 \pm (0,35) \uparrow 7$
GivenName-Address1-Suburb	2.808	836.950	$27,64 \pm (0,38) \uparrow 7$
GivenName-Address1	1.985	683.075	$26,01 \pm (0,38) \uparrow 6$
GivenName	449	428.548	$3,79 \pm (0,13)$

(↑) - número aproximado de vezes em que o tempo da combinação é superior comparado a um atributo

Para o *DatasetSintético*, existem oito atributos: *GivenName*, *Address1*, *Suburb*, *Surname*, *State*, *Address2*, *Culture* e *Title*, dos quais os cinco primeiros possuem maior relevância. Os grupos são formados do mesmo modo que no conjunto de dados *CDs*. Mais uma vez, a combinação dos atributos mais relevantes não melhora a eficácia do processo, devido às mesmas razões apresentadas para o conjunto de dados de *Restaurant*. No geral, considerando ambos os *datasets*, os valores de Revocação e Precisão são semelhantes para todas as combinações, pois isso o valor de *F-Measure* não tem aumento significativo. O grupo de pares candidatos criados por dois atributos de indexação está contido no grupo de pares candidatos criados por três atributos e assim sucessivamente, ou seja, um *cluster* está contido em outro. Dessa forma, os valores de Revocação, Precisão e *F-Measure* não melhoram porque as comparações ocorrem com os mesmos pares candidatos. No entanto, mais de uma vez, ou seja, os blocos não são livres de redundância (o que justifica o aumento do tempo de execução conforme apresentado posteriormente).

Por fim, analisamos o tempo médio de execução do processo de deduplicação de dados utilizando uma combinação de atributos na etapa de indexação. A Tabela 6.10 apresenta o total de BKs distintas, a quantidade de pares candidatos e o tempo médio de execução para cada combinação. Os resultados mostram que o tempo de execução aumenta significativamente quando vários atributos são utilizados. Por exemplo, no conjunto de dados *DatasetSynthetic*, o atributo *Given Name* é aproximadamente 15 vezes mais rápido do que uma combinação de cinco atributos (*Given Name*, *Address1*, *Suburb*, *Surname* e *State*).

Além disso, quando dois atributos são utilizados (*Given Name* e *Address1*), o tempo da deduplicação aumenta em torno de seis vezes. Para o *dataset CDs*, os re-

Tabela 6.11: Análise do melhor e pior valor de *F-Measure* nos conjuntos de dados

Conjunto de Dados	**Melhor *F-Measure***	**Pior *F-Measure***	**Diferença**
CDs	0,8351	0,1266	0,7085
Cora	0,8924	0,1150	0,7774
Restaurant	0,7514	0,5033	0,2481
10% de Duplicados	0,7856	0,4179	0,3677
30% de Duplicados	0,7932	0,4027	0,3905
50% de Duplicados	0,7978	0,3998	0,3980
70% de Duplicados	0,7987	0,3970	0,4017
90% de Duplicados	0,7993	0,3996	0,3997
10^2 Instâncias	0,8912	0,4642	0,4270
10^3 Instâncias	0,8545	0,4710	0,3835
10^4 Instâncias	0,8444	0,4580	0,3864
10^5 Instâncias	0,8257	0,4557	0,3700
10^6 Instâncias	0,7576	0,2362	0,5214
Média	$0,8175 \pm (0,0448)$	$0,3728 \pm (0,1290)$	$0,4446 \pm (0,1451)$

sultados são semelhantes porque utilizar mais de um atributo na indexação também aumenta o tempo do processo. O atributo *Artist* é quase duas vezes mais rápido que a combinação *Artist* e *Title* e aproximadamente três vezes mais rápido que a combinação de cinco atributos (*Artist*, *Genre*, *Title*, *Category* e *CDExtra*). Tais resultados são explicados pelo número de pares candidatos criados por cada combinação. Por exemplo, enquanto um atributo produz aproximadamente 400 mil pares candidatos, uma combinação quíntupla gera cerca de 12 milhões de pares candidatos (considerando o conjunto de dados *CDs*). Além disso, o número de BKs distintas indica que os blocos não são uniformemente distribuídos, pois há combinações com o mesmo número de chaves de bloco mas o total de pares candidatos aumenta em cerca de oito vezes (e.g. grupo de três e quatro atributos no *dataset CDs*).

No geral, conclui-se que o processo de deduplicação de dados não melhora com a combinação de atributos na indexação no algoritmo *Standard Blocking*, porque tanto o tempo de execução aumenta, quanto a eficácia não tem ganho significativo. Além disso, os resultados também não são melhorados com a combinação dos atributos mais relevantes conforme *3DR-Indexing*, devido às mesmas razões. Tais resultados destacam a importância do método proposto, pois apenas os atributos mais relevantes são identificados e, em seguida, utilizados na etapa de indexação.

6.5 Discussões

Conforme apresentado neste capítulo, o atributo de indexação possui grande impacto no processo de deduplicação de dados, porque tanto a eficácia quanto o tempo de execução do processo muda dependendo do atributo escolhido. Nesse contexto, esta seção apresenta discussões sobre os atributos estudados classificando-os quanto às métricas adotadas no Capítulo 4 e quanto aos resultados experimentais.

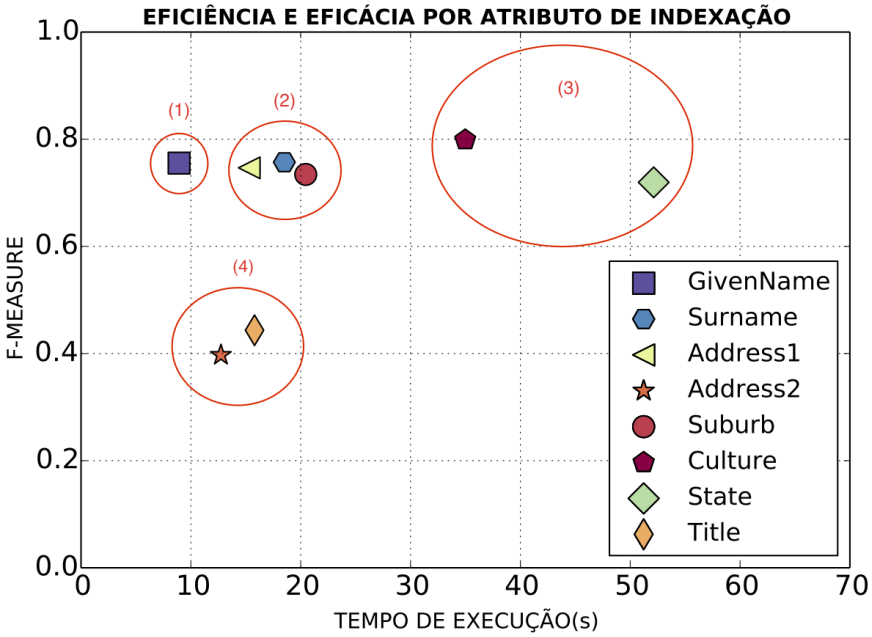


Figura 6.11: Classificação geral dos atributos de indexação

Inicialmente, a Tabela 6.11 apresenta o melhor e o pior resultados em termos de *F-Measure* considerando o melhor e o pior atributos de indexação em cada conjunto de dados. Em média, os resultados mostram que o total de duplicatas detectadas pode aumentar cerca de 44% escolhendo-se o melhor atributo de indexação, porque o melhor valor de *F-Measure* difere do pior em uma média de $0,4446 \pm (0,1451)$.

No geral, nós classificamos os atributos de indexação em quatro cenários quanto aos resultados do processo: (1) atributos eficazes e eficientes (*e.g.,* Given Name); (2) atributos eficazes porém menos eficientes (e.g., *Address1*, *Surname* e *Suburb*); (3) atributos eficazes porém ineficientes (e.g., *Culture* e *State*); e (4) atributos eficientes mas ineficazes (e.g., *Title* and *Address2*). A Figura 6.11 apresenta tal classificação onde os cenários estão enumerados de um a quatro.

Por fim, considerando a combinação das métricas do Capítulo 4, os atributos de indexação são classificados em cinco tipos, os quais apresentam quatro cenários nos resultados da deduplicação. A Tabela 6.12 detalha cada tipo de atributo de indexação bem como os resultados experimentais como segue.

- **Tipo 1** - atributos com alta densidade e duplicidade mas com alto nível de repetição e distintividade: são eficazes porém ineficientes;

- **Tipo 2** - atributos que possuem as quatro métricas com valores baixos: são eficientes porém ineficazes;

Tabela 6.12: Avaliação dos resultados do processo de deduplicação de dados considerando os tipos de atributos classificados pelas métricas apresentadas no Capítulo 4

Tipos de Atributos	**Conjunto de Métricas**				**Deduplicação**	
	Densidade	**Duplicidade**	**Repetição**	**Distintividade**	**Eficiente**	**Eficaz**
Tipo 1	ALTA	ALTA	ALTA	ALTA	X	✓
Tipo 2	BAIXA	BAIXA	BAIXA	BAIXA	✓	X
Tipo 3	ALTA	ALTA	BAIXA	ALTA	X	✓
Tipo 4	ALTA	ALTA	ALTA	BAIXA	X	✓
Tipo 5	ALTA	ALTA	BAIXA	BAIXA	✓	✓

*ALTA/BAIXA = o atributo têm melhor/pior valor da métrica comparado aos demais respectivamente
*✓/X = o resultado do processo é/não é eficiente ou eficaz

- **Tipos 3 e 4** - atributos que têm alta densidade e duplicidade mas a repetição ou a distintividade é alta: são eficazes mas ineficientes; e

- **Tipo 5** - atributos que possuem alta densidade e duplicidade mas não têm alta repetição ou distintividade: são os melhores atributos para a etapa de indexação, porque possuem o melhor *trade-off* entre eficiência e eficácia. O atributo ideal é do tipo 5, o qual é identificado e classificado automaticamente como mais relevante de acordo com o método *3DR-Indexing*.

Capítulo 7

Conclusões e Trabalhos Futuros

O foco deste trabalho foi analisar o impacto da escolha do atributo de indexação sobre as etapas do processo de deduplicação de dados. A principal contribuição é um método para selecionar automaticamente os melhores atributos para a etapa de indexação (*3DR-Indexing*). Especificamente, o método *3DR-Indexing* e as métricas de *Densidade*, *Distintividade*, *Duplicidade* e *Repetição* foram apresentados no Capítulo 4, bem como os critérios para cálculo da relevância de cada atributo em um conjunto de dados foram definidos. Também foi realizada uma extensa avaliação experimental, com projeto detalhado no Capítulo 5 e resultados no Capítulo 6. Os experimentos foram realizados em dados sintéticos e em dados reais, e consideraram os seguintes aspectos: (i) avaliação do efeito das etapas do processo de deduplicação de dados; (ii) análise do impacto do atributo de indexação sobre as técnicas *Schema-Agnostic* e *Schema-Based Configurations*; (iii) avaliação do efeito do atributo de indexação nos métodos de clusterização *Standard Blocking* e *Sorted Neighborhood*; (iv) validação do método proposto *3DR-Indexing*; e (v) avaliação da combinação de atributos na etapa de indexação do processo de deduplicação. Essas avaliações utilizaram métricas de Revocação, Precisão, *F-Measure*, correlação de *Spearman* e tempo médio de execução. Ademais, o método proposto foi comparado com *baselines* e com outras combinações empíricas de métricas.

As conclusões são guiadas pelas questões de pesquisas, conforme a seguir.

- **Questão 1 - Efeito do Atributo de Indexação**. Como visto nas avaliações experimentais, o atributo de indexação possui maior efeito sobre o processo de deduplicação de dados, e a escolha do atributo adequado é fundamental para garantir um processo de deduplicação eficaz e ao mesmo tempo eficiente. Em média, os melhores atributos de indexação diferem dos piores em 44% em termos de *F-Measure*, ou seja, a escolha aleatória ou sem nenhum critério de um atributo

de indexação faz com que aproximadamente 44% dos registros duplicados não sejam identificados. Ademais, comparando o efeito do atributo de indexação, da função de indexação e do algoritmo de clusterização sobre o processo, o atributo de indexação tem maior efeito (96%), ou seja, o processo de deduplicação de dados é prejudicado em termos de *F-Measure* quando o atributo de indexação é escolhido de forma errada, mesmo que combinado com métodos eficazes das outras etapas.

- **Questão 2 - *Schema-Agnostic vs. Based Configurations***. Segundo as avaliações experimentais, os atributos mais eficazes de cada conjunto de dados mantêm a eficácia em ambos os métodos de indexação, tanto utilizando o valor do atributo como chave de bloco (*Schema-Agnostic*), quanto codificando os valores por um função de BK (*Based Configurations*), ou seja, a função de indexação utilizada não influencia os resultados experimentais porque os valores de *F-Measure* são semelhantes em ambos métodos. Além disso, atributos ineficazes continuam ineficazes mesmo com a mudança de estratégia adotada na indexação, ou seja, os resultados experimentais são afetados pela qualidade dos atributos e não pelo método de indexação, pois os valores de *F-Measure* são ineficazes em ambos os métodos. No quesito tempo de execução, os dois métodos de indexação analisados têm tempo médio de execução estatisticamente igual. Por fim, nossos experimentos confirmam os resultados apresentados por Papadakis et al. [2015]. Segundo os autores, o método de indexação *Schema-Agnostic* é livre de domínio e mantém resultados significantes em vários domínios de dados. Nos experimentos, utilizamos conjuntos de dados com variados domínios, e o método *Schema-Agnostic* tem resultados significantes em termos de *F-Measure* para todos.

- **Questão 3 - *Standard Blocking vs. Sorted Neighborhood***. Os algoritmos de clusterização por blocos e vizinhos mais próximos têm resultados semelhantes considerando o atributo de indexação utilizado. No geral, os atributos mais eficazes mantêm a eficácia em ambos os algoritmos. Por exemplo, nos conjuntos de dados sintéticos, os atributos *Given Name*, *Surname* e *Address1* são eficazes em termos de *F-Measure* em ambas as técnicas. No entanto, existem atributos que diferem o valor de *F-Measure* obtido quando o algoritmo de agrupamento muda. Por exemplo, o atributo *State* é eficaz no algoritmo *Standard Blocking* mas ineficaz no *Sorted Neighborhood* (considerando os conjuntos de dados sintéticos). Dessa forma, conclui-se que atributos com alta *Repetição* não são eficazes no algoritmo de vizinhança porque altos valores de repetição distanciam os registros duplicados dos originais. Logo, a eficácia é prejudicada dependendo do tamanho

da janela escolhido, porque ela pode não conter todas as instâncias duplicadas; o que não acontece no algoritmo de blocos, porque todos os registros repetidos estão no mesmo bloco (e.g., atributos *Culture*, *State* e *Title* nos *datasets* sintéticos). Por fim, nossos experimentos diferem de *uma* avaliação experimental apresentada por Christen [2012b], onde os autores justificam que as técnicas de clusterização tradicionais não são eficazes no conjunto de dados *Restaurant* devido à enorme quantidade de erros. No entanto, nós realizamos experimentos com o mesmo conjunto de dados e com técnicas comumente usadas, e obtivemos resultados significantes em termos de Revocação, Precisão e *F-Measure*.

Questão 4 - O Método *3DR-Indexing*. Nossos resultados mostram que o método *3DR-Indexing* seleciona os melhores atributos para a etapa de indexação porque os atributos mais relevantes têm o melhor *trade-off* entre eficiência e eficácia na deduplicação. No geral, os resultados são significantes em 10 de 13 conjuntos de dados. Além disso, mesmo quando o método falha, ou seja, não seleciona o melhor atributo, os outros atributos de maior relevância ainda possuem resultados significantes em termos de *F-Measure* e tempo médio de execução. O método proposto é eficaz e eficiente em conjuntos de dados com poucas duplicatas (e.g., 10%), variados números de instâncias ($10^2 - 10^6$) e em quatro domínios de dados distintos (registros bibliográficos, informações pessoais, dados de CDs de música e informações de restaurantes). Ademais, as métricas utilizadas no *3DR-Indexing* apresentam um correlação *muito alta* com os resultados experimentais (considerando o coeficiente de correlação de *Spearman*) e são superiores aos coeficientes de correlação de outras combinações. Por fim, os resultados do método proposto superam os trabalhos apresentados por Bilenko & Mooney [2003], Christen [2012b], Hand & Christen [2018], Leitão et al. [2007], McCallum et al. [2000] e Papadakis et al. [2015].

Questão 5 - Combinação de Atributos na Etapa de Indexação. Nossos experimentos também avaliam a combinação de atributos na indexação utilizando o algoritmo *Standard Blocking*. Nós analisamos todas as combinações de atributos possíveis e a combinação de atributos mais relevantes segundo o método proposto. No geral, conclui-se que os resultados do processo de deduplicação não têm aumento significante quando uma combinação de atributos é utilizada na indexação, pois o valor de Revocação, Precisão e *F-Measure* é semelhante ao conseguido com apenas um atributo. Além disso, o tempo médio do processo de deduplicação é 15 vezes maior quando uma combinação de atributos é utilizada (e.g., conjunto de dados *DatasetSynthetic*).

Planeja-se realizar as seguintes avaliações experimentais como trabalhos futuros.

- **Conjuntos de Dados**. Nós avaliamos o atributo de indexação, o processo de deduplicação e o método *3DR-Indexing* em vários domínios de dados incluindo registros bibliográficos, dados sobre CDs, informações sobre restaurantes e dados de pessoas. Novos experimentos podem ser realizados em outros domínios contendo, por exemplo, dados geográficos, produtos de lojas online e informações de registros de crimes;

- **Testes Estatísticos**. Conforme resultados discutidos, o método *3DR-Indexing* tem resultados superiores ao dos trabalhos relacionados nos conjuntos de dados reais em termos de *F-Measure*. No entanto, o desempenho do método pode ser comparado com trabalhos relacionados utilizando testes estatísticos, como por exemplo o teste-t;

- **Casos de Falha**. Em alguns conjuntos de dados, o método proposto não conseguiu identificar como atributo mais relevante aquele que possui melhor *trade-off* entre eficiência e eficácia (i.e., um caso de falha). Assim, trabalhos futuros podem investigar o motivo dos casos de falha.

Os resultados desta dissertação estão publicados nos seguintes artigos.

- Souza, L.; Murai, F.; da Silva, A. P. C. & Moro, M. M. (2018). **Automatic Identification of Best Attributes for Indexing in Data Deduplication**. In *Proceedings of the 12th Alberto Mendelzon International Workshop on Foundations of Data Management*, Cali, Colômbia.

- Silva, L. S.; do Amaral, D. C. & Moro, M. M. (2017). **Uma avaliação de eficiência e eficácia da combinação de técnicas para deduplicação de dados**. In *Proceedings of the 32nd Brazilian Symposium on Databases*, pp. 160–171, Uberlândia, Brasil.

- Silva, L. S. & Moro, M. M. (2017). **Análise do impacto do gerador de conjuntos de dados em experimentos de deduplicação de dados**. In *Proceedings of the 32nd Brazilian Symposium on Databases*, pp. 228–233, Uberlândia, Brasil.

Além desses, mais dois estão em planejamento: *Combining Data Deduplication Techniques for Evaluating the whole Process* e *3DR-Indexing: a method for automatic identification of the best indexing attributes in data deduplication*.

Referências Bibliográficas

Baxter, R.; Christen, P. & Churches, T. (2003). A comparison of fast blocking methods for record linkage. Em *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, volume 3, pp. 25–27, Washington, USA.

Beskales, G.; Soliman, M. A.; Ilyas, I. F. & Ben-David, S. (2009). Modeling and querying possible repairs in duplicate detection. *Proceedings of the VLDB Endowment*, 2(1):598–609.

Bilenko, M.; Kamath, B. & Mooney, R. J. (2006). Adaptive blocking: Learning to scale up record linkage. Em *Proceedings of the 6th International Conference on Data Mining*, pp. 87–96, Hong Kong, China.

Bilenko, M. & Mooney, R. J. (2003). Adaptive duplicate detection using learnable string similarity measures. Em *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 39–48, Washington, USA.

Borges, E. N.; Becker, K.; Heuser, C. A. & Galante, R. (2011). An automatic approach for duplicate bibliographic metadata identification using classification. Em *Proceedings of the 30th International Conference of the Chilean on Computer Science Society*, pp. 47–53, Curico, Chile.

Borges, E. N.; Galante, R. M. & Gonçalves, M. A. (2008). Uma abordagem efetiva e eficiente para deduplicação de metadados bibliográficos de objetos digitais. Em *Proceedings of the 23rd Brazilian Symposium on Databases*, pp. 76–90, Campinas, Brasil.

Borges, E. N.; Pinheiro, R. F. & Dimuro, G. P. (2017). Contact deduplication in mobile devices using textual similarity and machine learning. Em *Proceedings of the 19th International Conference on Enterprise Information Systems*, pp. 64–72, Porto, Portugal.

Brizan, D. G. & Tansel, A. U. (2006). A survey of entity resolution and record linkage methodologies. *Communications of the IIMA*, pp. 41–50.

Canalle, G. K.; Lóscio, B. F. & Salgado, A. C. (2017). A strategy for selecting relevant attributes for entity resolution in data integration systems. Em *Proceedings of the 19th International Conference on Enterprise Information Systems*, pp. 80–88, Porto, Portugal.

Canuto, S. D.; Bianco, G. D.; Gonçalves, M. A.; Almeida, J. M. & Couto, T. (2013). UDRB: uma nova heurística eficaz para deduplicação de referências bibliográficas. Em *Anais do 28º Simpósio Brasileiro de Banco de Dados*, pp. 9:1–9:6, Recife, Brasil.

Carvalho, A. P.; Ferreira, A. A.; Laender, A. H. & Gonçalves, M. A. (2011). Incremental unsupervised name disambiguation in cleaned digital libraries. *Journal of Information and Data Management*, 2(3):289–304.

Carvalho, L. F. M.; Laender, A. H. F. & Meira Jr., W. (2015). Entity matching: A case study in the medical domain. Em *Proceedings of the 9th Alberto Mendelzon International Workshop on Foundations of Data Management*, pp. 57–68, Lima, Peru.

Carvalho, M. G.; Laender, A. H.; Gonçalves, M. A. & da Silva, A. S. (2012). A genetic programming approach to record deduplication. *IEEE Transactions on Knowledge and Data Engineering*, pp. 399–412.

Chen, J.; Jin, C.; Zhang, R. & Zhou, A. (2012). A learning method for entity matching. Em *Proceedings of 10th International Workshop on Quality in Databases*, Istanbul, Turkey.

Christen, P. (2006). A comparison of personal name matching: Techniques and practical issues. Em *Proceedings of the 6th IEEE International Conference on Data Mining*, pp. 290–294, Hong Kong, China.

Christen, P. (2008). Automatic record linkage using seeded nearest neighbour and support vector machine classification. Em *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 151–159, Las Vegas, USA.

Christen, P. (2012a). *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Springer, Berlin.

Christen, P. (2012b). A survey of indexing techniques for scalable record linkage and deduplication. *IEEE Transactions on Knowledge and Data Engineering*, 24(9):1537–1555.

Cohen, J. (1988). *Statistical Power Analysis for the Behavioral Sciences*. Academic Press.

Cohen, W. W.; Ravikumar, P. & Fienberg, S. E. (2003). A comparison of string distance metrics for name-matching tasks. Em *Proceedings of Workshop on Information Integration on the Web*, pp. 73–78, Acapulco, México.

Cohen, W. W. & Richman, J. (2002). Learning to match and cluster large high-dimensional data sets for data integration. Em *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 475–480, Edmonton, Canada.

Davis, P. T.; Elson, D. K. & Klavans, J. L. (2003). Methods for precise named entity matching in digital collections. Em *Proceedings of the 3rd ACM/IEEE Joint Conference on Digital Libraries*, pp. 125–127, Houston, USA.

Draisbach, U. & Naumann, F. (2009). A comparison and generalization of blocking and windowing algorithms for duplicate detection. Em *Proceedings of the International Workshop on Quality in Databases*, pp. 51–56, Lyon, France.

Draisbach, U.; Naumann, F.; Szott, S. & Wonneberg, O. (2012). Adaptive windows for duplicate detection. Em *Proceedings of the 28th International Conference on Data Engineering*, pp. 1073–1083, Arlington, Virginia.

Evangelista, L. O.; Cortez, E.; da Silva, A. S. & Meira Jr., W. (2010). Adaptive and flexible blocking for record linkage tasks. *Journal of Information and Data Management*, 1(2):167–182.

Fan, W.; Jia, X.; Li, J. & Ma, S. (2009). Reasoning about record matching rules. *Proceedings of the VLDB Endowment*, 2(1):407–418.

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

Fox, E. A.; Akscyn, R. M.; Furuta, R. K. & Leggett, J. J. (1995). Digital libraries. *Communications of the ACM*, pp. 22–28.

Goiser, K. & Christen, P. (2006). Towards automated record linkage. Em *Proceedings of the Fifth Australasian Data Mining Conference*, pp. 23–31, Sydney, Australia.

Hajishirzi, H.; Yih, W. & Kolcz, A. (2010). Adaptive near-duplicate detection via similarity learning. Em *Proceeding of the 33rd ACM SIGIR International Conference of Research and Development on Information Retrieval*, pp. 419–426, Geneva, Switzerland.

Hand, D. & Christen, P. (2018). A note on using the f-measure for evaluating record linkage algorithms. *Statistics and Computing*, 28(3):539–547.

Hernández, M. A. & Stolfo, S. J. (1995). The merge/purge problem for large databases. Em *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, pp. 127–138, New York, USA.

Jain, R. (1992). *The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling*. Wiley.

Leitão, L.; Calado, P. & Weis, M. (2007). Structure-based inference of xml similarity for fuzzy duplicate detection. Em *Proceedings of the 6th ACM Conference on Information and Knowledge Management*, pp. 293–302, Lisbon, Portugal.

Levin, F. H. & Heuser, C. A. (2010). Evaluating the use of social networks in author name disambiguation in digital libraries. *Journal of Information and Data Management*, 1(2):183–198.

McCallum, A.; Nigam, K. & Ungar, L. H. (2000). Efficient clustering of high-dimensional data sets with application to reference matching. Em *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 169–178, Boston, USA.

Momeni, F. & Mayr, P. (2016). Evaluating co-authorship networks in author name disambiguation for common names. Em *Proceedings of the 20th International Conference on Theory and Practice of Digital Libraries*, pp. 386–391, Hannover, Germany.

Müller, M.-C. (2017). Semantic author name disambiguation with word embeddings. Em *Proceedings of the 21st International Conference on Theory and Practice of Digital Libraries*, pp. 300–311, Thessaloniki, Greece.

Nascimento, D. C.; Pires, C. E. S. & Mestre, D. G. (2018). Heuristic-based approaches for speeding up incremental record linkage. *Journal of Systems and Software*, 137:335–354.

Odell, M. & Russell, R. (1918). The soundex coding system. *US Patents*, 1261167.

Papadakis, G.; Alexiou, G.; Papastefanatos, G. & Koutrika, G. (2015). Schema-agnostic vs schema-based configurations for blocking methods on homogeneous data. *Proceedings of the VLDB Endowment*, 9(4):312–323.

Papadakis, G.; Ioannou, E.; Niederée, C.; Palpanas, T. & Nejdl, W. (2011). Eliminating the redundancy in blocking-based entity resolution methods. Em *Proceedings of the 11th Annual International Joint Conference on Digital Libraries*, pp. 85–94, Ottawa, Canada.

Papadakis, G.; Koutrika, G.; Palpanas, T. & Nejdl, W. (2014). Meta-blocking: Taking entity resolutionto the next level. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1946–1960.

Papadakis, G.; Papastefanatos, G.; Palpanas, T. & Koubarakis, M. (2016). Scaling entity resolution to large, heterogeneous data with enhanced meta-blocking. Em *Proceedings of the 19th International Conference on Extending Database Technology*, pp. 221–232, Bordeaux, France.

Sadinle, M. (2017). Bayesian estimation of bipartite matchings for record linkage. *Journal of the American Statistical Association*, 112(518):600–612.

Sadinle, M. & Fienberg, S. E. (2013). A generalized fellegi-sunter framework for multiple record linkage with application to homicide record systems. *Journal of the American Statistical Association*, 108(502):385–397.

Silva, L. S.; do Amaral, D. C. & Moro, M. M. (2017). Uma avaliação de eficiência e eficácia da combinação de técnicas para deduplicação de dados. Em *Proceedings of the 32nd Brazilian Symposium on Databases*, pp. 160–171, Uberlândia, Brasil.

Silva, L. S. & Moro, M. M. (2017). Análise do impacto do gerador de conjuntos de dados em experimentos de deduplicação de dados. Em *Proceedings of the 32nd Brazilian Symposium on Databases*, pp. 228–233, Uberlândia, Brasil.

Souza, L.; Murai, F.; Silva, A. P. C. D. & Moro, M. M. (2018). Automatic identification of best attributes for indexing in data deduplication. Em *Proceedings of the 12th Alberto Mendelzon International Workshop on Foundations of Data Management*, Cali, Colombia.

Steorts, R. C.; Ventura, S. L.; Sadinle, M. & Fienberg, S. E. (2014). A comparison of blocking methods for record linkage. Em *Proceedings of the International Conference on Privacy in Statistical Databases*, pp. 253–268, Ibiza, Spain.

Su, W.; Wang, J. & Lochovsky, F. H. (2010). Record matching over query results from multiple web databases. *IEEE Transactions on Knowledge and Data Engineering*, 22(4):578–589.

Vieira, P.; Salgado, A. C. & Lóscio, B. F. (2016). A query-driven, incremental process for entity resolution. Em *Proceedings of the 10th Alberto Mendelzon International Workshop on Foundations of Data Management*, Panama City, Panama.

Vieira, P. K. M.; Lóscio, B. F. & Salgado, A. C. (2017). Dynamic indexing for incremental entity resolution in data integration systems. Em *Proceedings of the 19th International Conference on Enterprise Information Systems*, pp. 185–192, Porto, Portugal.

Winkler, W. E. (1990). String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. Em *Proceedings of the Section on Survey Research*, pp. 354–359, Anaheim, California.

Yan, S.; Lee, D.; Kan, M.-Y. & Giles, L. C. (2007). Adaptive sorted neighborhood methods for efficient record linkage. Em *Proceedings of the 7th ACM/IEEE Joint Conference on Digital Libraries*, pp. 185–194, Vancouver, Canada.

Zaït, M. & Messatfa, H. (1997). A comparative study of clustering methods. *Future Generation Computer Systems*, 13(2-3):149–159.