Explorando uma Nova Métrica para Calcular a Precisão Acurada de Resultados de Blocagem em Tarefas de Resolução de Entidades

Dimas Cassimiro Nascimento , Vítor Alan Bezerra da Silva

[1]Universidade Federal do Agreste de Pernambuco (UFAPE)
Garanhuns – PE – Brazil

dimas.cassimiro@ufape.edu.br,vitoralan.bs@gmail.com

Abstract. *Entity resolution is a crucial task in data integration, aiming to identify records that refer to the same real-world entity. Blocking techniques are widely used to improve efficiency by reducing the number of record comparisons. However, traditional metrics, such as Pair Quality (PQ), fail to account for redundant comparisons, potentially distorting the assessment of blocking effectiveness. This paper introduces the PQ^* metric, designed to provide a more accurate precision measure by eliminating the impact of redundant comparisons. We also propose the PQ^*C and $\widehat{PQ^*}E$ algorithms, which efficiently calculate and estimate PQ^*, respectively. Experimental results show that, for all evaluated datasets, PQ and PQ^* yield different results in every scenario where more than one blocking key is used. Furthermore, the difference between the two metrics increases as more blocking keys are employed for indexing.*

Resumo. *A resolução de entidades é uma tarefa fundamental na integração de dados, visando identificar registros que se referem à mesma entidade do mundo real. Técnicas de blocagem são amplamente empregadas para aumentar a eficiência, reduzindo o número de comparações entre registros. No entanto, métricas tradicionais, como Pair Quality (PQ), não consideram comparações redundantes, o que pode distorcer a avaliação da eficácia da blocagem. Este artigo apresenta a métrica PQ^*, projetada para fornecer uma medida mais precisa ao eliminar o impacto de comparações redundantes. Além disso, propomos os algoritmos PQ^*C e $\widehat{PQ^*}E$, projetados para calcular e estimar de forma eficiente a métrica PQ^*, respectivamente. Os experimentos realizados mostram que, para todos os conjuntos de dados avaliados, as métricas PQ e PQ^* produzem resultados distintos em todos os cenários em que mais de uma chave de blocagem é utilizada. Além disso, a diferença entre os resultados das métricas PQ e PQ^* aumenta à medida que mais chaves de blocagem são empregadas na indexação.*

1. Introdução

A Resolução de Entidades (RE) é uma tarefa fundamental na integração de dados, responsável por identificar registros que representam a mesma entidade no mundo real [Christen and Christen 2012]. Essa tarefa é particularmente desafiadora devido à presença de dados sujos, variações em formatos de escrita e ausência de identificadores únicos. Em

aplicações como sistemas de saúde, bases governamentais e consolidação de censos populacionais, a eficiência e a precisão na RE são essenciais para garantir a integridade dos dados [Getoor and Machanavajjhala 2012].

Devido ao alto custo computacional da RE, técnicas de blocagem são amplamente empregadas para reduzir o número de comparações entre pares de entidades [Li et al. 2020, Papadakis et al. 2020]. O objetivo da blocagem é agrupar registros semelhantes em blocos menores, dentro dos quais as comparações são realizadas, evitando a necessidade de um processo exaustivo de comparação entre todos os pares possíveis [Mestre et al. 2017a, Araújo et al. 2019].

A avaliação da precisão da blocagem é frequentemente realizada por meio das métricas PC (*Pairs Completeness*), RR (*Reduction Ratio*) e PQ (*Pairs Quality*) [Nascimento et al. 2020], sendo esta última responsável por medir a proporção de pares duplicados corretamente identificados em relação ao total de pares de registros a serem comparados [Papadakis et al. 2016b, Papadakis et al. 2020]. No entanto, a métrica PQ apresenta limitações, pois não leva em conta comparações redundantes, o que pode levar a conclusões imprecisas sobre a precisão do esquema de blocagem. Para mitigar essa limitação, propomos a métrica Precisão Acurada, denotada por PQ^*, que aprimora o cálculo da precisão da blocagem ao desconsiderar as comparações redundantes. Diferentemente de PQ, a métrica acurada proposta (PQ^*) proporciona uma visão mais realista da qualidade do resultado da blocagem, tendo em vista que, em um cenário real de RE, as comparações redundantes não são de fato computadas, evitando processamento desnecessário na etapa de comparação [Christen and Christen 2012].

Neste artigo, objetivamos propor diferentes abordagens para calcular a precisão acurada (PQ^*) do resultado da blocagem produzida em tarefas de RE. Mais especificamente, pretende-se investigar três questões principais:

Questão 1: Como otimizar o cálculo de PQ^*?

Questão 2: Como estimar o valor de PQ^* no contexto de grandes volumes de dados?

Questão 3: Qual a diferença de complexidade entre abordagens empregadas para calcular e estimar PQ^*?

Considerando as questões supracitadas, propusemos o algoritmo PQ^*C, otimizado para calcular PQ^* de maneira eficiente; e o algoritmo $\widehat{PQ^*}E$, projetado para estimar PQ^* de forma escalável. Os resultados experimentais obtidos demonstram que, na prática, as métricas PQ e PQ^* produziram resultados distintos em todas as bases de dados avaliadas. Além disso, a diferença entre os resultados das duas métricas aumenta na medida em que mais chaves de blocagem são empregadas no processo de indexação e, consequentemente, produzem mais comparações redundantes.

Em suma, são apresentadas as seguintes contribuições neste artigo:

- Proposição de uma nova métrica para o cálculo da precisão acurada do resultado de blocagem na tarefa de RE;
- Proposição de algoritmos para calcular de maneira exata e estimada a métrica proposta, juntamente com a análise assintótica teórica de suas complexidades;
- Avaliação experimental dos resultados das métricas propostas considerando bases

de dados com diferentes características.

O restante deste artigo está estruturado da seguinte forma: A seção 2 apresenta a fundamentação teórica do artigo. A seção 3 apresenta a formalização da métrica proposta. Na seção 4, são propostos algoritmos para calcular PQ, assim como para calcular e estimar o valor da métrica proposta (PQ^*). Na seção 5, detalhamos os experimentos realizados e analisamos os resultados obtidos. Na seção 6, apresentamos os principais trabalhos relacionados. Por fim, na seção 7, discutimos as conclusões do estudo e sugerimos direções para trabalhos futuros.

2. Fundamentação Teórica

Nesta seção, são explicados conceitos fundamentais para o entendimento do artigo.

2.1. Resolução de Entidades

A *Resolução de Entidades* (RE) é uma tarefa fundamental em integração de dados, cujo objetivo é identificar registros que se referem à mesma entidade no mundo real [Christen and Christen 2012]. Esse problema é especialmente desafiador quando lidamos com grandes volumes de dados heterogêneos, nos quais os registros podem conter erros tipográficos, variações de formatação e informações incompletas. Métodos tradicionais de RE operam comparando pares de registros para determinar sua similaridade, o que pode ser inviável computacionalmente para grandes bases de dados devido ao crescimento quadrático do número de comparações [Papadakis et al. 2013]. Para mitigar esse problema, abordagens baseadas em *blocagem* [Li et al. 2020, Papadakis et al. 2020] têm sido amplamente empregadas.

2.2. Blocagem

A *Blocagem* (*blocking*) é uma técnica usada para reduzir a complexidade computacional da RE, agrupando registros em blocos antes da comparação [Papadakis et al. 2016a]. Em vez de realizar comparações exaustivas entre todos os pares de registros, a blocagem restringe a comparação apenas aos registros que pertencem ao mesmo bloco, reduzindo assim a quantidade de pares considerados. Os métodos de blocagem variam desde abordagens baseadas em chaves fonéticas até técnicas avançadas de *meta-blocking* [Papadakis et al. 2013], que refinam a estrutura dos blocos para otimizar a eficiência e a qualidade das comparações.

2.3. PC e PQ

Para avaliar a eficiência e a eficácia das abordagens de blocagem, são utilizadas métricas como *Pair Completeness* (PC) e *Pairs Quality* (PQ). A métrica *PC* mede a proporção de pares de entidades correspondentes que foram incluídos nos blocos em relação ao total de pares verdadeiros existentes. Formalmente, o *PC* é definido como:

$$PC = \frac{|DP|}{|TP|} \tag{1}$$

onde $|DP|$ representa o número de pares duplicados detectados e $|TP|$ o total de pares duplicados existentes na(s) base(s) de dados.

Por sua vez, a métrica ***PQ*** mede a qualidade das comparações realizadas, ou seja, a fração de comparações que realmente correspondem a pares duplicados. Sua definição formal é dada por:

$$PQ = \frac{|DP|}{|CP|} \tag{2}$$

onde $|CP|$ representa o total de comparações realizadas. Um alto valor de ***PC*** indica que a abordagem conseguiu recuperar a maior parte das entidades duplicadas, enquanto um alto valor de ***PQ*** sugere que a maioria das comparações realizadas foi relevante.

2.4. Comparações Redundantes e Comparações Supérfluas

No contexto da Resolução de Entidades, as comparações realizadas durante o processo de blocagem podem ser classificadas em diferentes categorias. Comparações redundantes ocorrem quando o mesmo par de registros é avaliado múltiplas vezes, aumentando o custo computacional sem melhorar a qualidade da resolução [Papadakis et al. 2016a]. Esse fenômeno é particularmente comum em esquemas de blocagem que combinam múltiplas chaves por meio de disjunções, nos quais um mesmo par de registros pode ser alocado a diferentes blocos devido a diferentes critérios de agrupamento. Além disso, ocorre em técnicas de blocagem que geram blocos sobrepostos, como aquelas baseadas em janelas deslizantes, que fazem com que registros sejam incluídos em múltiplos blocos, resultando em comparações repetitivas e desnecessárias.

Por outro lado, comparações *supérfluas* correspondem àquelas que envolvem pares de registros que não representam a mesma entidade no mundo real. O objetivo de abordagens eficientes de blocagem, como o *meta-blocking*, é reduzir tanto as comparações redundantes quanto as comparações supérfluas, com o intuito de que os blocos contenham mais pares relevantes para o processo de RE.

3. Métrica PQ^*: Precisão Acurada de Blocagem

Nesta seção, é proposta uma nova métrica para calcular a precisão dos resultados de blocagem em tarefas de RE.

Embora PQ forneça uma estimativa útil da precisão das comparações, esta métrica é sensível a comparações redundantes. Se um mesmo par (e, e') for repetidamente comparado em diferentes blocos, este fato irá afetar a avaliação a qualidade da blocagem, pois PQ trata todas as instâncias de comparação entre registros como independentes, ainda que estejam relacionadas ao mesmo par de registros.

A métrica PQ^* (Precisão Acurada de Blocagem) é uma variação da métrica PQ (Pairs Quality) que busca medir a precisão das comparações geradas pela estratégia de blocagem, eliminando o impacto das comparações redundantes. Enquanto PQ é definido como a proporção de pares duplicados em relação ao total de pares comparados, PQ^* aprimora essa medida ao excluir as redundâncias artificiais introduzidas pelo particionamento dos dados.

3.1. Definição Formal

Seja $\mathcal{B} = \{B_1, B_2, \ldots, B_n\}$ um conjunto de blocos gerado por um método de blocagem, onde cada bloco B_i contém um subconjunto de registros. O número total de comparações

realizadas dentro desses blocos é dado por:

$$C = \sum_{B \in \mathcal{B}} |B|(|B| - 1)/2$$

No entanto, devido à possível sobreposição entre blocos, um mesmo par de registros pode ser comparado múltiplas vezes. Seja $\mathcal{U}$ o conjunto de pares únicos efetivamente comparados pelo esquema de blocagem, ou seja, sem considerar duplicatas de comparação. O número total de comparações únicas pode ser expresso como $|\mathcal{U}|$, e o número de pares duplicados corretamente identificados dentro dessas comparações é denotado por $|DupSet|$. Assim, a métrica PQ^* é definida como:

$$PQ^* = \frac{|DupSet|}{|\mathcal{U}|}$$

,tal que $|DupSet|$ representa o número de pares duplicados corretamente identificados e $|\mathcal{U}|$ corresponde ao número de pares únicos comparados, removendo múltiplas instâncias do mesmo par em blocos diferentes.

4. Abordagens para Cálculo de PQ e PQ^*

Nesta seção, são apresentadas abordagens para realizar o cálculo de PQ e PQ^*, assim como uma abordagem heurística para estimar o cálculo de PQ^* no intuito de processar grandes volumes de dados de maneira escalável.

Para facilitar a proposição dos algoritmos, utilizamos a noção de k-combinações de um conjunto, conforme apresentado na Definição 1.

Definition 1 (k-combinações de um Conjunto)**.** *Seja S um conjunto. Denotamos por $\binom{S}{k}$ o conjunto de todas as k-combinações do conjunto S, ou seja, $\binom{S}{k} = \{s \in 2^S \mid |s| = k\}$, tal que 2^S é o conjunto potência de S e $|S|$ representa a cardinalidade do conjunto S.*

4.1. Algoritmo PQC

O Algoritmo 1 (*PQC - PQ Calculator*) calcula a métrica *PQ* (pairs quality), que mede a proporção de pares duplicados em um conjunto de blocos $\mathcal{B}$. O algoritmo recebe como entrada um conjunto de blocos $\mathcal{B}$ e retorna um valor numérico representando a qualidade dos pares. Inicialmente, são criadas as estruturas `DupSet`, um conjunto vazio que armazenará os pares duplicados encontrados, e `Cont`, uma variável contadora inicializada com zero (linhas 2 e 3).

A partir da linha 3, o algoritmo percorre cada bloco $B \in \mathcal{B}$. Para cada bloco, a variável `Cont` é atualizada na linha 4 para armazenar o número total de pares que podem ser formados dentro do bloco. Esse cálculo é dado pela fórmula combinatória, que representa o número de pares de registros (e, e') dentro do bloco B. Em seguida, o algoritmo percorre todos os pares possíveis dentro de B (linha 5), utilizando a notação $\binom{B}{2}$, que representa todas as combinações de dois elementos do bloco. Para cada par

(e, e'), é verificado na linha 6 se os elementos são equivalentes ($e_1 \equiv e_2$). Caso essa condição seja verdadeira, o par é adicionado ao conjunto `DupSet` (linha 7).

Após a iteração sobre todos os blocos, o algoritmo retorna, na linha 8, a métrica ***PQ***. Essa equação representa a proporção de pares duplicados encontrados em relação ao total de pares produzidos pela blocagem.

Algorithm 1: PQC - PQ Calculator

Input: $\mathcal{B}$ conjunto de blocos
Output: PQ pairs quality

```
1 DupSet ← ∅
2 Cont ← 0
3 foreach b ∈ B do
4 |   Cont ⟵ 2^-1 * |B| * (|B| − 1))
5 |   foreach (e, e') ∈ (B 2) do
6 |   |   if e1 ≡ e2 then
7 |   |   |   DupSet.add(ordered(e, e'))
8 return |DupSet|/Cont
```

Análise Assintótica. Seja n o número total de blocos e m o tamanho médio dos blocos. O algoritmo percorre todos os blocos (linha 3), o que contribui com um fator $O(n)$. Para cada bloco B, ele gera todos os pares (e, e'), o que resulta em um custo de $O(m^2)$ (linha 5). A verificação de duplicatas (linha 6) ocorre em tempo $O(1)$, assumindo um modelo eficiente de verificação. Portanto, o custo total do algoritmo pode ser expresso como $O(nm^2)$, sendo dominado pela geração de pares dentro de cada bloco.

4.2. Algoritmo PQ^*C

O Algoritmo 2 (PQ^***C**) segue um princípio semelhante ao **PQC**, porém, busca eliminar comparações redundantes. Para isso, além do conjunto `DupSet`, é introduzida a estrutura `comparisonsSet`, um conjunto utilizado para armazenar os pares já comparados, garantindo que cada par seja avaliado apenas uma vez. O algoritmo percorre cada bloco B no conjunto $\mathcal{B}$ e, para cada par (e, e') gerado pelas combinações dentro do bloco, a representação ordenada do par é adicionada ao `comparisonsSet`. Esse armazenamento evita que um mesmo par seja avaliado repetidamente em diferentes blocos, reduzindo a redundância das comparações. Se os elementos do par forem considerados equivalentes, ele é adicionado ao conjunto `DupSet`. Ao final, a métrica ***PQ*** é calculada como a razão entre o número de pares duplicados armazenados em `DupSet` e o número total de pares de registros em `comparisonsSet`. Dessa forma, o algoritmo PQ^*C produz uma métrica mais acurada em relação à precisão do resultado da blocagem.

Algorithm 2: PQ^*C - PQ^* Calculator

Input: $\mathcal{B}$ conjunto de blocos
Output: PQ pairs quality

```
1 DupSet ← ∅
2 comparisonsSet ← ∅
3 foreach B ∈ 𝓑 do
4     foreach (e, e') ∈ (B 2) do
5         comparisonsSet.add(ordered(e, e'))
6         if e ≡ e' then
7             DupSet.add(ordered(e, e'))
8 return |DupSet|/|comparisonsSet|
```

Análise assintótica. O algoritmo percorre todos os blocos e, para cada bloco, gera todas as combinações de dois elementos. Sejam n o número de blocos e m o tamanho médio dos blocos. O loop externo percorre n blocos, resultando em $O(n)$. O loop interno gera todas as combinações de dois elementos dentro de um bloco de tamanho médio m, o que resulta em complexidade $O(m^2)$, considerando que a inserção dos pares de registros no conjunto $ComparisonsSet$ pode ser realizada em $O(1)$, empregando uma estrutura de dados baseada em hash. Assim, a complexidade total é aproximadamente $O(nm^2)$.

4.3. Comparação entre PQC e PQ^*C

Embora o algoritmo PQC apresente a mesma complexidade assintótica do PQ^*C, ele reduz significativamente o uso de memória. No algoritmo PQ^*C, o conjunto `comparisonsSet` é utilizado para armazenar pares comparados, garantindo que cada comparação seja realizada apenas uma vez. No entanto, essa abordagem consome memória proporcional ao número de pares únicos, o que pode ser proibitivo para grandes volumes de dados. O PQC, por outro lado, elimina a necessidade de armazenar essas comparações, tornando-se mais eficiente em termos de uso de memória.

Entretanto, o PQC tem a desvantagem de não desconsiderar comparações redundantes no cálculo da métrica *PQ* (*pairs quality*). Como resultado, o valor calculado pode ser distorcido pela inclusão de pares repetidos, o que pode comprometer a o nível de acurácia da métrica, dependendo da distribuição dos dados.

4.4. Algoritmo $\widehat{PQ^*}E$

Nesta seção, é proposto um algoritmo para estimar o valor da métrica PQ^* com base em uma técnica de amostragem. O objetivo consiste em desenvolver um algoritmo denominado $\widehat{PQ^*}E$ que seja muito mais eficiente do que a estratégia adotada pelo algoritmo 2 e que produza uma boa aproximação da métrica PQ^*. Formalmente, almeja-se que $T_{exec}(\widehat{PQ^*}E(\mathcal{B})) \ll T_{exec}(PQ^*C(\mathcal{B}))$ e $\widehat{PQ^*}E(\mathcal{B})) \simeq PQ^*C(\mathcal{B})$, onde $\mathcal{B}$ representa um conjunto de blocos recebido como entrada.

O Algoritmo 3 ($\widehat{PQ^*}E$) estima o valor da métrica PQ^*E, considerando um conjunto de blocos $\mathcal{B}$ e utilizando uma amostragem aleatória de proporção α. O algoritmo recebe como entrada um conjunto de blocos $\mathcal{B}$ e um valor α que define a fração dos blocos a serem considerados na estimativa. A saída do algoritmo é o valor estimado de $\widehat{PQ}^*$.

Inicialmente, são criadas as estruturas `DupSet`, um conjunto para armazenar pares duplicados identificados, e `comparisonsSet`, um conjunto para registrar as comparações realizadas (linhas 1-2).

A amostragem ocorre na linha 3, onde é selecionado um subconjunto $\mathcal{B}^-$ contendo $\alpha \cdot |\mathcal{B}|$ blocos de $\mathcal{B}$. Em seguida, o algoritmo percorre cada bloco $B \in \mathcal{B}^-$ (linha 4). Para cada bloco, são gerados todos os pares possíveis (e, e') dentro do bloco B, utilizando a combinação $\binom{B}{2}$ (linha 5). Cada par é então adicionado ao conjunto `comparisonsSet`. Para cada par de registros no bloco, caso os registros sejam duplicados, ou seja, $e \equiv e'$, então esse par é adicionado ao conjunto `DupSet` (linhas 7-8). Essa verificação permite identificar os pares que representam duplicatas dentro da amostra selecionada. Após o término da iteração sobre os blocos, o estimador $\widehat{PQ}^*$ é calculado e retornado na linha 9.

Algorithm 3: $\widehat{PQ}^*$E - PQ^* Estimator

Input: $\mathcal{B}$ conjunto de blocos, $\alpha \in [0, 1]$ porcentagem amostral dos blocos
Output: $\widehat{PQ}^*$ estimated accurate pairs quality

```
1 DupSet ← ∅
2 comparisonsSet ← ∅
3 B⁻ ⟵ select (|B| * α) blocks from B
4 foreach B ∈ B⁻ do
5     foreach (e, e') ∈ (B 2) do
6         comparisonsSet.add(ordered(e, e'))
7         if e ≡ e' then
8             DupSet.add(ordered(e, e'))
9 return (|DupSet|)/|comparisonsSet|
```

Análise Assintótica. Seja n o número total de blocos e m o tamanho médio dos blocos. O custo da seleção da amostra de tamanho αn na linha 6 é $O(\alpha n)$. Para cada bloco selecionado na amostra (linha 4), são gerados $O(m^2)$ pares (linha 5), resultando em um custo total de $O(\alpha n m^2)$. A inserção dos pares de registros nos conjuntos ocorre em complexidade $O(1)$, se utilizarmos uma estrutura de dados eficiente, como um hash map ou hash set. O cálculo final do estimador na linha 9 é $O(1)$. Dessa forma, a complexidade geral do algoritmo é $O(\alpha n m^2)$, sendo dominada pela geração de pares dentro dos blocos.

4.5. Discussão das Questões Investigadas

As abordagens apresentadas nesta seção foram concebidas para responder diretamente às três questões formuladas na Seção 4. A seguir, discutimos como cada uma das questões definidas na Seção 4:

- **Questão 1: Como otimizar o cálculo de PQ^*?**
  Essa questão é tratada pela proposição do algoritmo PQ^*C, que busca eliminar comparações redundantes por meio do uso de uma estrutura auxiliar de memória (`comparisonsSet`). Embora essa abordagem aumente o consumo de memória, ela melhora significativamente a acurácia do cálculo ao evitar contagens repetidas de pares, produzindo uma métrica mais acurada do do que o algoritmo PQC.
- **Questão 2: Como estimar o valor de PQ^* no contexto de grandes volumes de dados?**

Para responder a essa questão, foi desenvolvido o algoritmo $\widehat{PQ^*}E$, que realiza uma estimativa eficiente da métrica PQ^* utilizando uma técnica de amostragem. Ao processar apenas uma fração dos blocos disponíveis, o algoritmo reduz drasticamente o tempo de execução e o uso de memória, oferecendo uma alternativa viável para cenários que necessitem o processamento de grandes bases de dados.

- **Questão 3: Qual a diferença de complexidade entre abordagens empregadas para calcular e estimar PQ^*?**
  A análise assintótica das abordagens evidencia que o algoritmo PQC possui complexidade $O(nm^2)$, sendo mais eficiente em memória. O algoritmo PQ^*C, apesar de manter complexidade semelhante, tem maior uso de memória devido ao armazenamento explícito das comparações realizadas. Por fim, o estimador $\widehat{PQ^*}E$ apresenta complexidade reduzida $O(\alpha nm^2)$, o que o torna adequado para estimativas rápidas com razoável nível de acurácia.

5. Avaliação Experimental

Nesta seção, são conduzidos experimentos para avaliar empiricamente: i) os resultados da métrica proposta (PQ^*); e ii) como os resultados de (PQ^*) diferem da métrica clássica (PQ) para avaliação dos resultados de precisão produzidos por técnicas de blocagem no contexto de RE. Os experimentos foram conduzidos para responder às seguintes questões:

- **Q1:** Existem diferenças nos resultados produzidos pelas métricas (PQ^*) e (PQ)?
- **Q2:** Os resultados produzidos pelas métricas (PQ^*) e (PQ) variam de acordo com a quantidade de chaves de bloco empregadas?

5.1. Ambiente Experimental

Todos os experimentos foram conduzidos em um computador com processador Intel(R) Core(TM) Ultra 7 155H 3.80 GHz, 328GB de memória RAM e sistema operacional Windows 11. Os algoritmos foram implementados em Java, usando a biblioteca commons-codec para a indexação baseada em soundex.

5.2. Conjuntos de Dados

Selecionamos conjuntos de dados sintéticos e do mundo real para avaliar a abordagem proposta, considerando diferentes características e tamanhos. Os seguintes conjuntos de dados foram utilizados na avaliação experimental:

- **North Carolina Voters**[1]: conjunto de dados reais de eleitores do estado da Carolina do Norte;
- **DS3**: conjunto de dados contendo registros de publicações científicas [Mestre et al. 2017b];
- **Company Names**: conjunto de dados com nomes de empresas (utilizado para avaliar abordagens de RE com classificação coletiva em [Hassanzadeh et al. 2009]);
- **Cora**: conjunto de dados de registros de publicações;
- **DBLPM4**: conjunto de títulos de publicações (também utilizado em [Hassanzadeh et al. 2009]);

Esses conjuntos foram escolhidos por apresentarem diferentes tamanhos, proporções de duplicatas e número de atributos, o que permite uma avaliação abrangente dos algoritmos propostos.

[1] https://dbs.uni-leipzig.de/research/projects/benchmark-datasets-for-entity-resolution

Base de Dados	número de chaves de blocagem	PQ	PQ^*
Cora	1	0.45	0.45
	2	0.20	0.35
	3	0.13	0.25
DBLPM4	1	0.22	0.22
	2	0.14	0.22
	3	0.062	0.095
Company Names	1	0.13	0.13
	2	0.09	0.13
	3	0.07	0.13
DS3	1	0.12	0.12
	2	0.03	0.04
	3	0.028	0.033
NCV	1	0.0016	0.0016
	2	0.00083	0.0016
	3	0.00079	0.0014

Tabela 1. Resultados de PQ e PQ^* considerando uma, duas e três chaves de blocagem combinadas na forma de disjunção.

5.3. Design dos Experimentos

Para cada conjunto de dados, aplicamos a técnica de blocagem padrão [Christen and Christen 2012], incluindo chaves de blocagem baseada em chaves fonéticas (Soundex) e blocagem por prefixos ou sufixos. Para cada base de dados, foram avaliados os resultados produzidos pelas métricas PQ e PQ^*, considerando 1, 2 e 3 chaves de blocagem combinadas na forma de disjunção.

Para o cálculo de PQ, foi empregado o algoritmo PQC (Algoritmo 1). Por sua vez, para o cálculo de PQ^*, utilizamos o algoritmo PQ^*C (Algoritmo 2).

No cenário em que apenas uma chave de blocagem é empregada para indexar a base de dados, não são produzidas comparações redundantes, tendo em vista que a blocagem padrão produz blocos disjuntos. Por sua vez, ao empregar um número $n > 1$ de chaves de blocagem, podem ser produzidas comparações redundantes considerando os blocos produzidos pelas diferentes chaves de blocagem e, consequentemente, é possível obter resultados distintos para os resultados produzidos pelas métricas PQ^* e PQ. No entanto, decidimos avaliar os resultados das métricas considerando apenas uma chave de blocagem para garantir que a implementação realizada estava correta. Neste sentido, devem ser verificadas a seguintes condições nos resultados obtidos:

$$\big(n = 1\big) \implies \big(PQ^* = PQ\big) \tag{3}$$

$$\diamondsuit(n > 1 \ \wedge PQ^* > PQ) \tag{4}$$

, tal que $\diamondsuit$ denota o símbolo de possibilidade da lógica modal.

5.4. Resultados Experimentais

Os resultados apresentados na Tabela 1 permitem responder às questões Q1 e Q2 formuladas no início desta seção. Com relação à **Q1**, observa-se que, para todos os conjuntos de dados avaliados, as métricas PQ e PQ^* produzem resultados idênticos quando apenas uma chave de blocagem é utilizada. Este comportamento era esperado, pois nesse cenário os blocos gerados são disjuntos e, portanto, não há comparações redundantes entre pares de registros, o que garante que ambas as métricas computem o mesmo conjunto de comparações no denominador das respectivas métricas.

Ao considerar múltiplas chaves de blocagem combinadas por disjunção (isto é, $n = 2$ ou $n = 3$), as métricas passam a produzir resultados distintos. Em todos os casos, os valores de PQ^* superam os de PQ. Esta diferença é explicada pelo fato de que o cálculo clássico de PQ considera as comparações redundantes, ao passo que PQ^* é projetada para eliminar tais redundâncias da contagem total de comparações. Dessa forma, a métrica PQ^* reflete de forma mais precisa o impacto da blocagem sobre a qualidade da RE, especialmente em cenários em que há sobreposição entre os blocos ou múltiplas chaves de bloco são empregadas para realizar a indexação dos registros.

Com relação à **Q2**, os resultados confirmam que o uso de múltiplas chaves de blocagem tende a acentuar a diferença entre as métricas. À medida que se aumenta o número de chaves, observa-se uma redução dos valores de PQ, enquanto os valores de PQ^* permanecem estáveis ou decrescem em menor intensidade. Isso reforça a capacidade de PQ^* em atenuar o impacto da redundância, capturando de forma mais justa a contribuição dos blocos adicionais para a identificação de pares corretos.

6. Trabalhos Relacionados

A tarefa de resolução de entidades foi bastante investigada em pesquisas realizadas nas últimas décadas [Nascimento et al. 2016, Papadakis et al. 2020, Li et al. 2020], dada sua importância em contextos como integração de bases de dados, aplicação de censos populacionais e detecção de fraudes.

Uma das abordagens mais utilizadas para avaliar a qualidade dos algoritmos de vinculação é baseada em medidas tradicionais de classificação, como precisão, revocação e F-medida. No entanto, essas métricas podem não ser adequadas em contextos com grande desbalanceamento entre pares correspondentes e não correspondentes, como discutido em [Hand and Christen 2018], no qual os autores argumentam que o uso da F-medida pode não ser adequado quando aplicada em problemas de RE. Isso ocorre principalmente devido à grande assimetria entre o número de pares verdadeiros positivos (correspondências) e pares verdadeiros negativos (não correspondências), o que pode levar a uma superestimação da qualidade dos classificadores. Para resolver tal problema, os métodos precisam classificar a mesma quantidade de pares de registros, caso contrário a avaliação da classificação deixa de ser justa.

Diversos trabalhos anteriores já exploraram os desafios associados à RE e suas métricas de avaliação. Referências clássicas da área de resolução de entidades [Elmagarmid et al. 2007, Christen and Christen 2012] discutem os principais algoritmos, métricas e desafios da área. Por sua vez, técnicas de blocagem mais robustas, baseadas na poda de grafos de blocagem, foram propostas em [Gagliardelli et al. 2022,

Papadakis et al. 2014, Papadakis et al. 2013], nos quais são empregadas as métricas de blocagem PQ (Pairs Quality), PC (Pairs Completeness) e Reduction Ration (RR) e cardinalidade agregada dos blocos (i.e., a quantidade de comparações produzidas por todos os blocos gerados na etapa de indexação).

Mais recentemente, outros trabalhos [Zeakis et al. 2023, Li et al. 2024] focaram na avaliação de modelos pré-treinados, e.g. baseados em BERT, no contexto de RE. Para tal, são exploradas as métricas de avaliação *Blocking Recall* (equivalente a PC) e *scalability*. Como pode ser observado, a literatura mais recente realizou pouco foco na proposição de novas métricas para avaliação de blocagem no contexto de RE. Diferentemente dos trabalhos discutidos nesta seção, este artigo apresenta um esforço direcionado na proposição e avaliação inicial de uma nova métrica para avaliação de resultados de blocagem no processo de RE.

7. Conclusões e Trabalhos Futuros

Neste trabalho, propomos abordagens para calcular e estimar a métrica PQ^*, que avalia a precisão acurada produzida por blocos gerados em tarefas de resolução de entidades. Os experimentos realizados demonstraram que, para todos os conjuntos de dados avaliados, as métricas PQ e PQ^* produzem resultados distintos em todos os cenários em que mais de uma chave de blocagem é utilizada. Outrossim, os resultados experimentais obtidos demonstram que o uso de mais chaves de blocagem combinadas como disjunção tende a aumentar a diferença entre as métricas PQ e PQ^*.

Com base nessas conclusões, sugerem-se algumas direções para trabalhos futuros. Primeiramente, uma linha de pesquisa interessante envolve a adaptação dos algoritmos propostos para cenários de *streaming*, onde os dados são processados de forma contínua e não podem ser armazenados integralmente na memória. Além disso, pretendemos estender o presente trabalho ao realizar uma análise experimental mais robusta para avaliar a influência do parâmetro α sobre o valor estimado de PQ^*, considerando bases de dados mais volumosas.

Referências

Araújo, T. B., Pires, C. E. S., Mestre, D. G., Nóbrega, T. P. d., Nascimento, D. C. d., and Stefanidis, K. (2019). A noise tolerant and schema-agnostic blocking technique for entity resolution. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 422–430.

Christen, P. and Christen, P. (2012). *The data matching process*. Springer.

Elmagarmid, A. K., Ipeirotis, P. G., and Verykios, V. S. (2007). Duplicate record detection: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 19(1):1–16.

Gagliardelli, L., Papadakis, G., Simonini, G., Bergamaschi, S., Palpanas, T., et al. (2022). Generalized supervised meta-blocking. *Proceedings of the VLDB Endowment*, 15(9):1902–1910.

Getoor, L. and Machanavajjhala, A. (2012). Entity resolution: theory, practice & open challenges. *Proceedings of the VLDB Endowment*, 5(12):2018–2019.

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

Hassanzadeh, O., Chiang, F., Lee, H. C., and Miller, R. J. (2009). Framework for evaluating clustering algorithms in duplicate detection. *Proceedings of the VLDB Endowment*, 2(1):1282–1293.

Li, B.-H., Liu, Y., Zhang, A.-M., Wang, W.-H., and Wan, S. (2020). A survey on blocking technology of entity resolution. *Journal of Computer Science and Technology*, 35:769–793.

Li, H., Li, S., Hao, F., Zhang, C. J., Song, Y., and Chen, L. (2024). Booster: leveraging large language models for enhancing entity resolution. In *Companion Proceedings of the ACM Web Conference 2024*, pages 1043–1046.

Mestre, D. G., Pires, C. E. S., and Nascimento, D. C. (2017a). Towards the efficient parallelization of multi-pass adaptive blocking for entity matching. *Journal of Parallel and Distributed Computing*, 101:27–40.

Mestre, D. G., Pires, C. E. S., Nascimento, D. C., de Queiroz, A. R. M., Santos, V. B., and Araujo, T. B. (2017b). An efficient spark-based adaptive windowing for entity matching. *Journal of Systems and Software*, 128:1–10.

Nascimento, D. C., Pires, C. E., and Mestre, D. (2016). Data quality monitoring of cloud databases based on data quality slas. In *Big-Data Analytics and Cloud Computing: Theory, Algorithms and Applications*, pages 3–20. Springer.

Nascimento, D. C., Pires, C. E. S., and Mestre, D. G. (2020). Exploiting block co-occurrence to control block sizes for entity resolution. *Knowledge and Information Systems*, 62(1):359–400.

Papadakis, G., Koutrika, G., Palpanas, T., and Nejdl, W. (2013). Meta-blocking: Taking entity resolutionto the next level. *IEEE Transactions on Knowledge and Data Engineering*, 26(8):1946–1960.

Papadakis, G., Papastefanatos, G., and Koutrika, G. (2014). Supervised meta-blocking. *Proceedings of the VLDB Endowment*, 7(14):1929–1940.

Papadakis, G., Papastefanatos, G., Palpanas, T., and Koubarakis, M. (2016a). Scaling entity resolution to large, heterogeneous data with enhanced meta-blocking. In *EDBT*, pages 221–232.

Papadakis, G., Skoutas, D., Thanos, E., and Palpanas, T. (2020). Blocking and filtering techniques for entity resolution: A survey. *ACM Computing Surveys (CSUR)*, 53(2):1–42.

Papadakis, G., Svirsky, J., Gal, A., and Palpanas, T. (2016b). Comparative analysis of approximate blocking techniques for entity resolution. *Proceedings of the VLDB Endowment*, 9(9):684–695.

Zeakis, A., Papadakis, G., Skoutas, D., and Koubarakis, M. (2023). Pre-trained embeddings for entity resolution: an experimental analysis. *Proceedings of the VLDB Endowment*, 16(9):2225–2238.