Melhorias no Processo de Blocagem para Resolução de Entidades Baseadas na Relevância dos Termos

Laís Soares Caldeira[1], **Anderson Almeida Ferreira**[1]

[1]Departamento de Computação – Universidade Federal de Ouro Preto (UFOP)
Ouro Preto, MG – Brasil

laissoarescaldeira@hotmail.com, anderson.ferreira@ufop.edu.br

Abstract. *Entity Resolution is a task commonly faced in data integration process. Due to quadratic number of comparisons to decide those instances belonging to the same entity, we need another way for performing such comparisons. In order to mitigate such a problem, techniques of blocking and block processing have been applied aiming the efficiency. In this work, we propose options to choose terms in the blocking step based on their relevance to the dataset in the phases of blocking and processing of blocks. We assess our proposal comparing it against relevant works available in the literature. The results show that our proposal decrease the run time by half, increasing the efficiency.*

Resumo. *Resolução de Entidades é uma tarefa comumente enfrentada no processo de integração de dados. Por necessitar de um número de comparações de ordem quadrática, torna-se inviável aplica-lá em grandes conjuntos de dados. Técnicas de blocagem e de processamentos de blocos têm sido propostas, visando amenizar esse problema. Neste trabalho, é proposta uma forma de escolher termos para serem usados na etapa de blocagem e no processamento de blocos, com base em sua relevância na coleção de dados. A proposta é avaliada comparando-a com trabalhos relevantes publicados na literatura. Os resultados mostram que a proposta deste trabalho reduz o tempo de processamento pela metade e melhora a qualidade dos blocos gerados.*

1. Introdução

A Web é um universo em crescimento e vem sendo dominada por conteúdo semi-estruturado e não estruturado. Além do aumento do montante de dados na *Web*, há uma grande diversidade entre as estruturas desses dados, dando origem a um gigantesco volume de dados heterogêneos. Tal questão representa um dos maiores desafios para busca na Web, levando a utilização de técnicas de integração de dados para melhorar os resultados retornados por essas buscas [Madhavan et al. 2007]. No processo de integração de dados, informações de diversas fontes devem ser comparadas e combinadas para que os usuários possam acessá-las e manipulá-las de forma unificada [Halevy et al. 2006].

Uma questão central do processo de integração de dados em larga escala é a Resolução de Entidades (ER - *Entity Resolution*), ou seja, a tarefa de identificar diferentes instâncias que pertencem a mesma entidade do mundo real [Christen 2012]. Uma entidade pode ser uma pessoa, uma empresa ou qualquer outro objeto com significado bem definido. Tipicamente a ER compara cada instância de uma coleção de dados com todas as outras, ou seja, a quantidade de comparações é de ordem quadrática com relação a quantidade de instâncias, tornando-se impraticável em grandes coleções de dados.

Para a ER se tornar escalável, normalmente são utilizadas técnicas de blocagem (conhecidas como técnicas de *blocking* - em inglês) [Christen 2012]. As técnicas de blocagem podem ser usadas para aprimorar o tempo de processamento em ER dividindo as instâncias em blocos, para comparar apenas as instâncias dentro do mesmo bloco. Com isso, o ganho em usar blocos está na redução de comparações entre as instâncias.

Os blocos podem ser reprocessados por técnicas de processamentos de blocos. Tais técnicas tentam reduzir comparações redundantes (instâncias comparadas várias vezes) e supérfluas (entre instâncias pertencentes a entidades distintas). Meta-blocagem (*meta-blocking* - em inglês) é uma técnica de reestruturação de blocos que descarta drasticamente comparações redundantes e supérfluas, por meio da transformação do conjunto de blocos em um grafo, onde os vértices correspondem às instâncias e as arestas conectam vértices que representam instâncias que coocorrem em um bloco, e objetiva manter apenas as arestas mais promissoras de correspondência [Papadakis et al. 2014]. Melhorias significativas na eficiência são encontradas com a aplicação de meta-blocagem. Porém, há muito o que se investigar para que a precisão das técnicas relacionadas ao processo de manipulação de blocos seja melhorada e que o tempo de processamento seja reduzido.

O trabalho tem como objetivo principal melhorar a eficiência (tempo de construção dos blocos), sem diminuir (podendo melhorar) a eficácia, em termos de correspondência entre instâncias encontradas, de técnicas de blocagem usadas em processos de ER, evitando comparações desnecessárias entre instâncias de uma coleção. Assim, o foco do trabalho é o processo de blocagem e não a tarefa inteira de Resolução de Entidades.

Contribuições: Inspirado nas abordagens baseadas em *tokens* (ou seja, termos) para formação dos blocos, a hipótese principal deste trabalho é que, características dos termos presentes nas instâncias da coleção de dados podem ser úteis para gerar blocos mais indicados para a ER. Assim, a originalidade do trabalho está no fato de avaliar e usar características específicas dos termos para alcançar melhorias na eficiência e na eficácia do processo de blocagem e seu processamento. Diversas características extraídas de termos foram analisadas e experimentadas, sendo que as características que obtiveram melhores resultados são apresentadas neste trabalho. Assim, este trabalho apresenta melhorias no processo de blocagem por meio do PBBRT (Processo de Blocagem Baseado na Relevância de Termos), que se divide em dois passos. Primeiramente, foi desenvolvida uma forma de escolher os termos a serem usados para blocagem por meio da entropia dos termos na coleção de dados. Com os blocos gerados no passo anterior, uma técnica de processamento de blocos que se baseia em meta-blocagem é adaptada utilizando a frequência dos termos na coleção de dados. Em conjunto, os dois passos são usados para produzir conjuntos de blocos de alta qualidade. O PBBRT foi avaliado em 3 coleções de dados semi-estruturados do mundo real (podendo ser aplicado a dados estruturados), mostrando resultados satisfatórios em relação a precisão e ao tempo de processamento, comparados aos da técnica de meta-blocagem proposta em Papadakis et al. [2016].

O restante deste trabalho está estruturado da seguinte forma: A Seção 2 descreve alguns conceitos importantes para o trabalho. A Seção 3 apresenta os trabalhos relacionados. A Seção 4 descreve a proposta deste trabalho. A Seção 5 descreve os experimentos e analisa os resultados. Finalmente, a Seção 6 apresenta conclusões trabalhos futuros.

2. Fundamentação Teórica

Nesta seção, são discutidos alguns conceitos fundamentais para o trabalho.

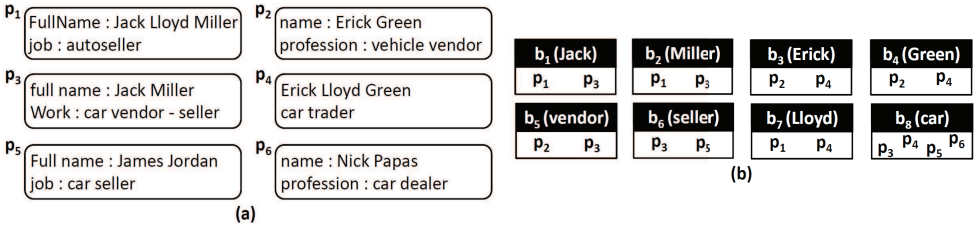

Figura 1. (a) Um conjunto de instâncias, e (b) os blocos resultantes da aplicação da técnica *Token Blocking*. Figura extraída de Papadakis et al. (2016).

2.1. Resolução de Entidades

No contexto de Resolução de Entidades (ER), uma instância p de um conjunto de dados P se refere a uma entidade e é descrita por meio de uma lista de atributos. Duas instâncias, p_i e p_j, são consideradas *duplicatas* quando p_i e p_j descrevem ou se referem à mesma entidade ($p_i \equiv p_j$). O objetivo da resolução de entidades é identificar todas as instâncias que são duplicatas no conjunto P, onde $D(P)$ representa esse conjunto de duplicatas e $|D(P)|$ a quantidade de duplicatas em $D(P)$ [Papadakis et al. 2016].

Há dois tipos de tarefas de ER: *Clean-Clean ER* e *Dirty ER*. A tarefa *Clean-Clean ER* recebe dois conjuntos sem duplicatas, mas que se sobrepõem, e identifica as instâncias que correspondem entre os conjuntos, enquanto a tarefa *Dirty ER* recebe como entrada um único conjunto com duplicatas e produz como saída um conjunto de grupos contendo cada um instâncias que são correspondentes [Papadakis et al. 2014].

2.2. Blocagem

Técnicas de blocagem aprimoram o tempo de processamento da ER. A maioria das técnicas de blocagem lidam com alto nível de heterogeneidade, tanto nos valores quanto nos nomes dos atributos. Isso é contornado normalmente ignorando as informações sobre o esquema e a semântica. Por exemplo, a técnica *Token Blocking* [Papadakis et al. 2013] é uma técnica de blocagem que lida com essa questão, colocando em um mesmo bloco instâncias que compartilham pelo menos um termo nos valores de seus atributos.

O exemplo da Figura 1 ilustra a técnica *Token Blocking*. A Figura 1 (a) contém as instâncias p_1, p_2, p_3, p_4, p_5 e p_6, em que p_1 corresponde a p_3 e p_2 corresponde a p_4, ou seja, pares que correspondem a mesma entidade (duplicatas). *Token Blocking* agrupa as instâncias nos blocos mostrados na Figura 1 (b). É possível notar que ambos os pares de duplicatas coocorrem em pelo menos um bloco, gerando um total de 13 comparações (par a par em cada bloco). A abordagem de força bruta, aquela que compara uma instância com todas as outras sem formar os blocos, teria 15 comparações.

No exemplo, os blocos b_1 e b_3 contêm uma comparação redundante e b_2 e b_4 também. Todos os outros blocos possuem comparações supérfluas entre instâncias não correspondentes, exceto para a comparação redundante p_3-p_5 em b_8, que se repete em b_6. No total, os blocos da Figura 1 (b) envolvem 2 comparações necessárias, 3 redundantes e 8 supérfluas, dentre as 13 comparações. Isso pode ser considerado uma proporção elevada. Neste trabalho, um conjunto de blocos será representado por B, com $|B|$ denotando seu tamanho (número de blocos) e $||B||$ sua cardinalidade (número total de comparações).

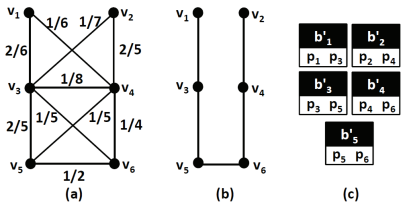

Figura 2. (a) Grafo de blocagem extraído dos blocos da Figura 1 (b), (b) um possível grafo de blocagem com arestas podadas, e (c) os novos blocos derivados. Figura extraída de Papadakis et al. (2016).

2.3. Meta-blocagem

Os blocos resultantes da blocagem podem ser reestruturados pela meta-blocagem [Papadakis et al. 2014]. Ela transforma um conjunto de blocos B em um grafo de blocagem GB, que contém um vértice para cada instância da coleção de dados remanescente da blocagem e uma aresta para cada par de instâncias de um bloco.

A Figura 2 (a) mostra o grafo para os blocos na Figura 1 (b) para uma estratégia de meta-blocagem. Cada aresta no grafo é ponderada com um peso análogo à probabilidade de que as instâncias ligadas pela aresta se referem a mesma entidade. Quanto maior o peso de uma aresta, mais provável é que as instâncias representadas nos vértices conectados sejam correspondentes. Como exemplo para a poda das arestas, pode-se definir como limiar a média dos pesos de todas as arestas do grafo. O grafo podado é mostrado na Figura 2 (b). O conjunto de blocos reestruturados B' é formado por meio da criação de um novo bloco para cada aresta mantida. Note que na Figura 2 (c) existem 5 blocos referentes às 5 arestas do grafo da Figura 2 (b). No entanto, o conjunto de blocos B' reduz as comparações de 13 para apenas 5, mantendo originalmente o número de possíveis duplicatas encontradas. Observe que a meta-blocagem tenta podar as arestas do grafo de blocagem deixando os vértices de instâncias correspondentes conectados.

A meta-blocagem descarta parte das arestas do grafo de blocagem utilizando um algoritmo de poda centrado nas arestas ou um algoritmo de poda centrado nos vértices do grafo. Em [Papadakis et al. 2014], os autores propuseram 4 opções de poda: *Cardinality Edge Pruning* (CEP), que ordena as arestas em ordem decrescente de peso e mantém somente as k primeiras, sendo $k = \lfloor \sum_{b \in B} |b|/2 \rfloor$ e $|b|$ o tamanho de cada bloco pertencente ao conjunto B; *Cardinality Node Pruning* (CNP), que mantém as *top-k* arestas da vizinhança de um vértice, sendo $k = \lfloor \sum_{b \in B} |b|/|P| - 1 \rfloor$ e $|P|$ a quantidade de instâncias no conjunto P; *Weighted Edge Pruning* (WEP), que descarta todas as arestas com peso menor que um limiar; e *Weighted Node Pruning* (WNP), que considera cada vértice do grafo e suas arestas adjacentes, podando as arestas que são inferiores a um limiar local.

O exemplo da Figura 2 utiliza um algoritmo de poda centrado nas arestas do grafo de blocagem. Um exemplo de poda centrada nos vértices do grafo é apresentado na Figura 3 (a). Para cada vértice na Figura 2 (a), foram mantidas as arestas incidentes que excedam o peso médio dos vértices vizinhos (vizinhança). Para maior clareza, as arestas mantidas são dirigidas, uma vez que podem ser mantidas na vizinhança de ambas as instâncias incidentes. Novamente, cada aresta mantida forma um novo bloco, obtendo o conjunto de blocos reestruturados B' da Figura 3 (b). Neste caso, o conjunto de blocos B' reduz as comparações de 13 para 9 em relação ao conjunto de blocos da Figura 1 (b).

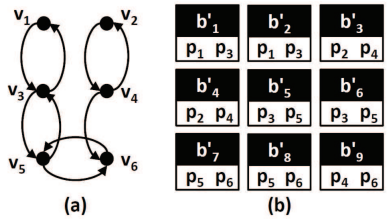

Figura 3. (a) Um possível grafo de blocagem com poda centrada em vértice, para o grafo da Figura 2 (a). (b) Os novos blocos derivados do grafo podado. Figura extraída de Papadakis et al. (2016).

3. Trabalhos Relacionados

Técnicas de blocagem e técnicas de processamento de blocos são frequentemente utilizadas em ER. A primeira técnica de blocagem encontrada na literatura é a *Standard Blocking* [Fellegi and Sunter 1969]. Nessa técnica, cada instância é representada por apenas uma chave de blocagem predefinida, agrupando as instâncias em blocos que compartilham exatamente a mesma chave. Desta forma, *Standard Blocking* é considerada uma técnica livre de redundância, pois produz blocos que não se sobrepõem. As técnicas de blocagem, em sua maioria, produzem blocos que se sobrepõe. Em Christen [2012], são apresentadas análises comparativas de uma grande parte das técnicas de blocagem. De acordo com [Papadakis et al. 2014], dependendo da interpretação de redundância, técnicas de blocagem podem ser classificadas em três categorias: técnicas de redundância positiva, técnicas de redundância negativa e técnicas de redundância neutra.

Técnicas de redundância positiva garantem que, se existem dois ou mais blocos com pares de instâncias, é provável que elas sejam correspondentes. Dentro desta categoria se encaixam as técnicas *Token Blocking* (descrita na Seção 2.2) e *Attribute Clustering* [Papadakis et al. 2013]. A técnica *Attribute Clustering* explora padrões nos nomes dos atributos na coleção de dados, e utiliza tais informações para construção dos blocos. As técnicas de redundância negativa garantem que as instâncias correferentes compartilham apenas um bloco, por exemplo, *Canopy Clustering* [McCallum et al. 2000] utiliza uma métrica de distância para colocar instâncias em blocos sobrepostos. As técnicas de redundância neutra produzem blocos sobrepostos, mas o número de blocos comuns entre duas instâncias é irrelevante para a probabilidade de correferência. *Sorted Neighborhood* [Hernández and Stolfo 1995] é um exemplo de técnica para essa categoria, que, por meio de uma janela deslizante de tamanho fixo, passa gradualmente sobre todas as instâncias da coleção de dados, criando os blocos dinamicamente.

As técnicas de processamento de blocos destinam-se a processar um conjunto de blocos já existentes, visando diminuir a quantidade de comparações entre as instâncias. São exemplos: *Iterative Blocking* [Whang et al. 2009] e *Comparison Propagation* [Papadakis et al. 2013]. *Iterative Blocking* distribui as duplicatas encontradas em um bloco a outros blocos que irão ser posteriormente processados, gerando resultados de novas instâncias coreferentes em outros blocos, tornando todo o processo iterativo. *Comparison Propagation* utiliza uma estrutura de tabela *hash* que possibilita descartar todas as comparações redundantes de um conjunto de blocos, mantendo as duplicatas encontradas. Meta-Blocagem, descrita na Seção 2.3, é considerada também uma técnica de processamento de blocos. As contribuições apresentadas em [Papadakis et al. 2016] para técnicas de meta-blocagem relacionadas a tarefa *Dirty ER* são comparadas a deste trabalho.

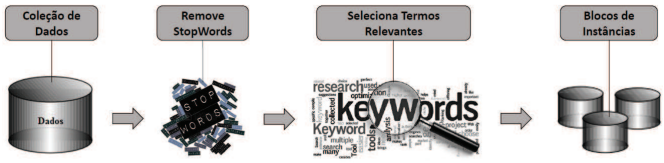

Figura 4. Etapas da construção de blocos do PBBRT

Simonini et al. [2016] propõem o BLAST. BLAST utiliza uma estratégia baseada em LSH (*Locality-Sensitive Hashing*) que coleta informações estatísticas diretamente dos dados. Com base nessas informações, os atributos são particionados de acordo com a semelhança de seus valores e, em seguida, é aplicada a técnica *Token Blocking* explorando os atributos de particionamento. Assim, apenas as instâncias cujos *tokens* pertencem a atributos na mesma partição serão comparadas. Posteriormente, usa-se essa informação na aplicação da técnica de meta-blocagem WNP, produzindo conjunto de blocos de alta qualidade, direcionados para a tarefa *Clean-Clean ER*.

A principal diferença da proposta deste trabalho para os demais está no fato de usar características de termos para a obtenção de blocos iniciais e a aplicação de características de termos no processamento dos blocos. Além disso, a proposta deste trabalho tem como foco a tarefa *Dirty ER*, diferentemente do trabalho de Simonini et al. [2016], que foca na tarefa *Clean-Clean ER*. Em [Papadakis et al. 2016], ambas as tarefas ER são abordadas.

4. PBBRT - Processo de Blocagem Baseado na Relevância dos Termos

Nesta seção, são apresentadas as melhorias no processo de blocagem por meio do PBBRT (Processo de Blocagem Baseado na Relevância dos Termos). O PBBRT é composto por duas técnicas que são utilizadas em conjunto: uma técnica de blocagem e uma técnica de processamentos de blocos. A seguir, as técnicas são retratadas em mais detalhes.

4.1. Blocagem do PBBRT

Para lidar com a complexidade do espaço de informações altamente heterogêneo, a técnica de blocagem do PBBRT se baseia na redundância positiva, onde cada instância pode ser colocada em vários blocos, de forma que quanto mais blocos duas instâncias estiverem, maior será a probabilidade das instâncias serem correspondentes. Isso é feito para reduzir o número de correspondências perdidas e é praticamente indispensável no contexto de dados heterogêneos. Para realizar a blocagem de um conjunto de instâncias, PBBRT verifica a relevância dos termos dessas instâncias e cria blocos onde os termos relevantes são as chaves. Assim, cada bloco contém as instâncias em que sua chave correspondente está presente nos valores dos atributos.

A Figura 4 ilustra as etapas da técnica de construção de blocos do PBBRT. A primeira etapa efetua a remoção de *stopwords*. *stopwords* são termos não representativos em uma coleção de dados. Geralmente esses termos são: preposições, artigos, advérbios, números, pronomes e pontuação [Wilbur and Sirotkin 1992].

Após a remoção de *stopwords*, é realizada a seleção de termos relevantes. Para identificar os termos relevantes foram analisadas e experimentadas diversas características relacionadas aos termos, tais como, entropia, quantidade de caracteres nos termos e frequência do termo na coleção. Dentre elas, a entropia mostrou melhores resultados e é a usada para blocagem neste trabalho. O conceito de entropia foi transformado numa

medida de quantidade de informação por [Shannon 2001]. Neste trabalho, considere entropia (H) calculada como H = - $\sum prob_i \times \log_2 prob_i$, sendo $prob_i$ a probabilidade do termo i estimada pela sua frequência na coleção de dados. O cálculo da probabilidade dos termos, $prob_i$, seria f_i dividido pela frequência total de todos os termos, $FreqTotal$. Portanto $prob_i = \frac{f_i}{FreqTotal}$. Dado que as probabilidades para todos os valores de frequência existentes foram estimadas, é realizado o somatório para encontrar H.

A ideia neste trabalho é fazer a blocagem com os termos com o maior ganho de informação, com o intuito de construir blocos representativos. Quanto maior o valor de H, maior é o ganho. Para saber a importância que uma determinado termo tem para uma coleção de dados, foram realizados cálculos de entropia desconsiderando os termos com uma determinada frequência em análise. Visto que cada cálculo de entropia desconsidera um valor de frequência dos termos, quanto menor for o resultado do cálculo, maior é a importância dos termos com a frequência excluída. Dessa forma, consideraram-se os termos excluídos quando a entropia fica menor que um limiar para serem chaves na geração de blocos. O limiar foi definido como a média de todos os valores de entropia encontrados.

Por fim, a criação dos blocos segue a técnica *Token Blocking*, descrita na Seção 2.2, considerando apenas os termos selecionados.

4.2. Processamento de Blocos do PBBRT

Os quatro esquemas de poda para a meta-blocagem, apresentados na Seção 2.3 (CEP, CNP, WEP e WNP), foram investigados em [Papadakis et al. 2016], onde mostrou-se que CNP e WNP geralmente são mais eficientes. Dado que o CNP supera WNP em termos de precisão, um algoritmo alternativo para o CNP, chamado *Reciprocal Node-centric Pruning*, abreviado como *Reciprocal* CNP, foi apresentado em [Papadakis et al. 2016] para aplicações ER com o objetivo de melhorar a eficiência do processo. Considerando que o presente trabalho tem foco em melhorar também a precisão, uma adaptação foi feita no *Reciprocal* CNP para receber como entrada o conjunto de blocos criados por meio da técnica de blocagem, apresentada na subseção anterior, visando atingir melhores resultados relacionados à eficiência. Na subseção a seguir é apresentado o funcionamento do *Reciprocal* CNP e na Subseção 4.2.2 é mostrada a adaptação realizada neste trabalho.

4.2.1. Reciprocal CNP

Na Subseção 2.3, foi descrita que a meta-blocagem transforma o conjunto de blocos resultante da técnica de blocagem em um grafo. No entanto, é muito custoso materializar o grafo de blocagem em memória, visto que, para grandes coleções de dados, o grafo pode conter milhares de vértices e arestas. Uma solução foi implementar o grafo implicitamente, integrando ao *Reciprocal* CNP o *Comparison Propagation* [Papadakis et al. 2013]. Nessa técnica, um índice de instâncias é construído. Esse índice constitui uma estrutura de tabela *hash*, cujas chaves são os identificadores das instâncias da coleção de dados remanescentes da blocagem e seus valores são listas com índices dos blocos que contêm as instâncias correspondentes.

Dessa forma, ao invés de iterar sobre todas as comparações nos blocos B de entrada, *Reciprocal* CNP itera sobre todas as instâncias, dado que o índice de instâncias foi criado. Para cada instância p_i (correspondente a um vértice no grafo), é identificada todas as outras instâncias que coocorrem com p_i nos blocos associados (vizinhança) e o

valor de frequência de cada um desses pares de instâncias é registrado em um vetor. Ao final, tem-se o número de blocos compartilhados por p_i e p_j e essa informação é utilizada para estimar o peso da aresta que liga as instâncias p_i e p_j. Dado que a vizinhança armazena cada par único de instâncias que coocorrem, a redundância pode ser eliminada. *Reciprocal* CNP trata as comparações redundantes como pares de instâncias com grandes chances de correspondência. Essas comparações correspondem a ligações recíprocas no grafo de blocagem. Por exemplo, as arestas $a_{1,3}$ e $a_{3,1}$ na Figura 3 (a) indicam que p_1 tem alta probabilidade de correspondência com p_3 e vice-versa, reforçando assim, a probabilidade de que as duas instâncias são duplicatas. Com base neste raciocínio, *Reciprocal* CNP mantém uma comparação para cada par de instâncias que são mutuamente ligadas no grafo de blocagem. Instâncias que estão conectadas por uma única aresta não são comparadas no conjunto de blocos reestruturado.

Outra funcionalidade do *Reciprocal* CNP está relacionada aos vértices do grafo, em que um limiar de cardinalidade k é derivado ($k = \lfloor \sum_{b \in B} |b|/|P| - 1 \rfloor$). Este limiar determina o número máximo de arestas que serão mantidas e é utilizado como critério de poda da vizinhança de cada vértice. Uma estrutura de dados armazena os *top-k* vizinhos do vértice, considerando o peso da aresta que interliga o vértice ao vizinho analisado. Cinco esquemas foram propostos para a ponderação das arestas do grafo de blocagem: ARCS, CBS, ECBS, JS e EJS (mais detalhes em [Papadakis et al. 2014]). Todos normalizam os pesos entre [0, 1], de modo que os valores mais altos se referem às arestas que são mais propensas a conectar instâncias que se correspondem. A média de todos os esquemas de ponderação é utilizada para ponderar as arestas do grafo de blocagem.

4.2.2. Adaptação do Reciprocal CNP

A adaptação proposta neste trabalho para o *Reciprocal* CNP acontece exatamente na fase de ponderação. Segundo Shannon [2001], à medida que a ocorrência de um grupo de símbolos (termos) se torna mais frequente, a quantidade de informação decresce. Desta forma, foi utilizada para ponderar as arestas essa intuição, implementada por meio de uma função logarítmica, ou seja, se o par de instâncias (vértices interligados a aresta em análise) vier de um bloco cuja chave de blocagem for um termo que ocorre pouco na coleção, esse termo pode ser mais informativo (maior peso) do que outros, levando a manter no conjunto de blocos finais pares com maior probabilidade de correspondência.

A função $\frac{1}{log(x)}$ expressa bem essa questão, pois quanto menor o valor de x para valores positivos, maior o valor do resultado atribuído pela função. Assim, a ponderação das arestas é feita da seguinte forma: Dado uma frequência x de um termo (chave do bloco onde se extraiu a comparação entre p_i e p_j), é obtido o peso referente a aresta a_{ij}, sendo $peso(x) = \frac{1}{log(x)}$. Para cada aresta, é somado o peso encontrado anteriormente com o esquema de ponderação de arestas ECBS (*Enhanced Common Blocks*). A ponderação com o ECBS se dá da seguinte forma: $ECBS(p_i, p_j, B) = |B_{ij}| \times log\frac{|B|}{|B_i|} \times log\frac{|B|}{|B_j|}$, sendo $|B_{ij}|$ a quantidade de blocos comuns entre p_i e p_j, $|B_i|$ a quantidade de blocos que contém p_i, $|B_j|$ a quantidade de blocos que contém p_j e $|B|$ a quantidade total de blocos. Portanto, a equação final da ponderação de arestas é dada por: $pesoAresta(x, p_i, p_j, B) = \frac{1}{log(x)} + (|B_{ij}| \times log\frac{|B|}{|B_i|} \times log\frac{|B|}{|B_j|})$. Dessa forma, a técnica de processamento de blocos do PBBRT descarta comparações supérfluas, pela poda das arestas com peso menores, utilizando a frequência dos termos na coleção de dados para ponderar as arestas.

Tabela 1. Características técnicas das coleções de dados

	C1	C2	C3
|P|	63.869	50.797	3.354.773
|AV|	208.065	971.445	19.064.747
|D(P)|	2.308	22.863	892.579
||F||	2.580.284.412	1.290.142.206	5.627.249.263.378

5. Avaliação Experimental

A implementação do PBBRT foi feita em Java 8 como uma extensão do *framework* de código aberto apresentado em [Papadakis et al. 2014]. Todas as técnicas comparadas ao PBBRT neste trabalho foram implementadas no mesmo *framework*. Os experimentos foram realizados em um computador com processador Intel Xeon(R) E5620 2.40 GHz, 47GB de RAM e sistema operacional CentOS Linux 7. As medições de tempo de processamento de todas as técnicas foram repetidas 10 vezes e a média dos tempos é apresentada como resultado, com uma confiança de 95%, de modo a minimizar efeitos

5.1. Coleções de Dados

Para a avaliação experimental, foram utilizadas três coleções de dados semi-estruturados reais, que variam de tamanho e características. As coleções de dados referem-se a *Dirty ER*, ou seja, coleções de instâncias com duplicatas, disponíveis publicamente[1].

A Tabela 1 mostra características das coleções de dados para *Dirty ER*. A coleção C1 contém dados bibliográficos originados da DBLP (http://dblp.org) e Google Scholar (https://scholar.google.gr). A coleção C2 contém dados de filmes originados da IMDB (http://www.imdb.com) e da DBPedia (http://dbpedia.org). A coleção C3 contém instâncias de dois *snapshots* diferentes da *Wikipedia* em inglês (http://en.wikipedia.org). A seguinte notação é utilizada na apresentação das características técnicas das coleções de dados: $|P|$ representa o número de instâncias na coleção, $|AV|$ o número total de pares atributo-valor, $|D(P)|$ o número de duplicatas existentes, $||F||$ o número de comparações executadas pela abordagem força bruta, que compara cada instância com todas as outras.

5.2. Baseline

Os resultados deste trabalho são comparados com os apresentados por Papadakis et al. [2016]. Para a blocagem, Papadakis et al. [2016] utilizam a técnica *Token Blocking*. Em seguida, aplicam *Block Purging* [Papadakis et al. 2013] e *Block Filtering* [Papadakis et al. 2016]. *Block Purging* descarta os blocos que contêm mais da metade das instâncias da coleção, que são as chaves de blocagem altamente frequentes. *Block Filtering* visa reestruturar o conjunto de blocos eliminando as instâncias que são desnecessárias nos blocos. Essas tarefas são consideradas pré-processamento para a meta-blocagem, descartando mais da metade das arestas desnecessárias do grafo, em média.

Para o processamento de blocos, é utilizada para comparação a técnica de meta-blocagem *Reciprocal* CNP descrita na Subseção 4.2.1. Segundo Papadakis et al. [2016], as técnicas de meta-blocagem superam as demais técnicas de processamento de blocos.

5.3. Métricas de Avaliação

Para avaliar a qualidade de um conjunto de blocos B criado a partir do conjunto de instâncias da entrada P, são utilizadas as métricas *Pair Completeness (PC)* e *Pair Qua-*

[1]https://sourceforge.net/projects/erframework/files/DirtyERDatasets/RealDatasets/

lity (PQ) [Christen 2012]. Dado que $||B||$ é o número total de comparações nos blocos, $D(B)$ o conjunto de instâncias que coocorrem, $|D(B)|$ o seu tamanho (número de duplicatas possíveis de serem encontradas) e $|D(P)|$ o número de duplicatas existentes na coleção de dados (utiliza-se um gabarito onde as duplicatas estão identificadas), tem-se:

- *Pairs Completeness (PC)* é similar a revocação. Mede quão eficaz é a técnica em agrupar as duplicatas existentes. PC está definido no intervalo [0, 1], com valores mais altos indicando maior completude. Fórmula: $PC = \frac{|D(B)|}{|D(P)|}$.
- *Pairs Quality (PQ)* é similar a precisão. Mede quão eficiente é a técnica na obtenção dos blocos. PQ toma valores no intervalo [0, 1], com valores mais altos indicando maior qualidade para B. Fórmula: $PQ = \frac{|D(B)|}{||B||}$.

O desempenho da Resolução de Entidades pode ser distinguido em duas categorias: eficiência intensiva e eficácia intensiva. A eficiência intensiva tem como objetivo minimizar o tempo de processamento, sem deixar de detectar a maioria das duplicatas existentes. Mais formalmente, o seu objetivo é maximizar a qualidade dos pares (PQ) para uma completude (PC) que exceda 0,80. A eficácia intensiva permite um tempo de processamento mais elevado, desde que a completude (PC) seja maximizada, onde o seu valor não deve estar abaixo de 0,95 [Papadakis et al. 2016]. Juntamente com PC e PQ, outras duas métricas são utilizadas para a avaliação do processo de blocagem: $||B||$ indica o número total de comparações verdadeiras nos blocos; e o *Tempo de Processamento*, que mede o tempo necessário para extrair o conjunto de blocos finais. Para ambos, quanto menor o valor, melhor é o resultado. O intervalo de confiança (IC) para a média dos tempos de processamento serão apresentados usando uma distribuição *t de Student*.

5.4. Resultados

Dentre as características dos termos presentes em uma coleção avaliadas para identificar termos promissores para obter blocos, foram experimentadas a entropia (PC = 0,983 e PQ = 3,42E-04, coleção C1), frequência do termo na coleção (PC = 0,999 e PQ = 5,95E-05, coleção C1) e quantidade de caracteres do termo (PC = 0,999 e PQ = 3,23E-05, coleção C1). Entropia teve resultados melhores para C1 e também para C2 e C3, em termos de PQ, levando a sua escolha. Por brevidade, são apresentados os resultados apenas das características que obtiveram os melhores resultados.

A Tabela 2 mostra o desempenho da blocagem do PBBRT (representado por PBBRT P1, primeira parte do processo do PBBRT), em comparação com a técnica de blocagem usada em [Papadakis et al. 2016] (representado por T), aplicados às coleções de dados C1, C2 e C3. Observa-se que o PBBRT P1 obteve uma redução no número de comparações nos blocos de 68% em média, considerando as três coleções de dados, ao custo de uma redução de PC em torno de 1,7%, mantendo, ainda assim, o PC acima de 0,95 para todas as coleções. Dessa forma, PQ aumenta em média 4 vezes, diminuindo pela metade o tempo de execução nas coleções C1 e C2 e em 64% para coleção C3. Com a técnica de blocagem do PBBRT escolheu-se os termos relevantes, criando, assim, um número menor de blocos e tornando o processo de blocagem mais rápido e preciso.

No entanto, resultados mais satisfatórios para a precisão foram encontrados com a aplicação da técnica de processamento de blocos. A Tabela 3 mostra os resultados encontrados com a primeira e a segunda parte do PBBRT (blocagem + processamento de blocos), comparados ao *Reciprocal* CNP aplicado aos blocos criados, apresentados em Papadakis et al. [2016]. Os resultados mostrados para o *Reciprocal* CNP utilizam para

Tabela 2. Comparações das técnicas de blocagem

	C1		C2		C3	
	T	**PBBRT P1**	**T**	**PBBRT P1**	**T**	**PBBRT P1**
PC	0,994	0,983	0,976	0,951	0,997	0,982
PQ	9,62E-05	3,42E-04	1,62E-04	1,08E-03	3,86E-05	7,29E-05
||B||	2,38E+07	6,64E+06	1,37E+08	2,02E+07	2,31E+10	1,20E+10
Tempo	4,3 s	2,2 s	8,4 s	4,2 s	13 min	4,7 min
IC Tempo	± 0,05 s	± 0,1 s	± 0,2 s	± 0,1 s	± 11,7 s	± 5,6 s

ponderação de arestas a média dos cinco esquemas propostos em Papadakis et al. [2014] (ARCS, CBS, ECBS, JS e EJS). Porém, para a técnica de processamento de blocos do PBBRT somente o ECBS foi utilizado. Assim, são comparados ao PBBRT, o *Reciprocal* CNP usando a média dos cinco esquemas de ponderação de arestas (representado por M1) e o *Reciprocal* CNP com o ECBS (representado por M2).

Tabela 3. Comparações das técnicas de processamento de blocos

	C1			C2			C3		
	M1	**M2**	**PBBRT**	**M1**	**M2**	**PBBRT**	**M1**	**M2**	**PBBRT**
PC	0,846	0,867	0,855	0,650	0,736	0,760	0,868	0,882	0,871
PQ	0,017	0,016	0,024	0,057	0,063	0,078	0,111	0,102	0,132
||B||	1,19E+05	1,25E+05	8,35E+04	2,86E+05	2,65E+05	2,23E+05	7,12E+06	7,73E+06	5,87E+06
Tempo	22,8 s	21,9 s	10,8 s	8,1 min	5,0 min	57,8 s	13,9 h	10,6 h	7,9 h
IC Tempo	± 0,5 s	± 0,5 s	± 0,4 s	± 12,4 s	± 11,2 s	± 0,9 s	± 1,8 min	± 1,7 min	± 12,9 s

Comparado à técnica M1, o PBBRT tem melhores resultados sob todas as métricas de avaliação em todas as coleções de dados. O número de comparações diminui em média 23,4%, com um aumento no PC de 16,9% na coleção C2 e em torno de 1% nas coleções C1 e C3. Em média, PQ aumenta em torno de 32,3% e o tempo de processamento é reduzido em torno de 61,2%. A técnica M2 comparada ao PBBRT só ganha em relação ao PC nas coleções C1 e C3, em torno de 1,3%. Vale ressaltar, que o PC desejado deve estar acima de 0,80 para eficiência intensiva. Na coleção C2, esse valor para a métrica PC não é atingido por nenhuma das técnicas em comparação. Porém, o PBBRT melhora o resultado para PC em relação a M1 e M2.

Apesar da revocação (PC) variar muito pouco comparando as técnicas, há melhorias na precisão (PQ) e ganhos expressivos em relação ao tempo total de processamento. Os tempos de processamento das técnicas mostrados na Tabela 2 e na Tabela 3 foram analisados utilizando o teste de hipótese *t de Student*, avaliando se havia diferença significativa entre as médias dos tempos de processamento das técnicas. A hipótese nula, que afirma que as duas médias de tempos são iguais, foi rejeitada para todas as técnicas em todas as coleções de dados com uma confiança de 95%, comprovando o ganho em relação ao tempo de processamento do PBBRT.

6. Conclusões

Neste trabalho, foram apresentadas melhorias no processo de blocagem por meio do PBBRT. O PBBRT verifica a relevância dos termos presentes em coleções de dados e utiliza tais informações com o objetivo de aumentar a qualidade dos blocos, diminuindo o número de comparações em uma tarefa de Resolução de Entidades, por exemplo. O PBBRT foi avaliado experimentalmente em coleções de dados reais e os resultados mostram que o PBBRT supera uma técnica representativa de meta-blocagem em até 16,9% de

completude e em 32,3%, em média, na qualidade dos blocos gerados, reduzindo o tempo de processamento aproximadamente pela metade. Assim, foi demonstrado que o PBBRT pode processar eficientemente grandes coleções de dados altamente heterogêneas.

Como trabalhos futuros, pretende-se adaptar o PBBRT para a tarefa *Clean-Clean ER*, avaliar outras características baseada em termos e outros meios de ponderação de arestas, visando melhorar ainda mais os resultados em termos de eficiência e eficácia.

Agradecimentos. Este trabalho foi apoiado e financidao pela Universidade Federal de Ouro Preto (UFOP), Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq (grant 312395/2017-5).

Referências

Christen, P. (2012). A survey of indexing techniques for scalable record linkage and deduplication. *IEEE TKDE*, 24(9):1537–1555.

Fellegi, I. P. and Sunter, A. B. (1969). A theory for record linkage. volume 64, pages 1183–1210.

Halevy, A., Rajaraman, A., and Ordille, J. (2006). Data integration: the teenage years. In *VLDB*, pages 9–16.

Hernández, M. A. and Stolfo, S. J. (1995). The merge/purge problem for large databases. *ACM SIGMOD Rec.*, 24(2):127–138.

Madhavan, J., Jeffery, S. R., Cohen, S., Dong, X., Ko, D., Yu, C., and Halevy, A. (2007). Web-scale data integration: You can only afford to pay as you go. In *CIDR*, pages 342–350.

McCallum, A., Nigam, K., and Ungar, L. H. (2000). Efficient clustering of high-dimensional data sets with application to reference matching. In *ACM SIGKDD*, pages 169–178.

Papadakis, G., Ioannou, E., Palpanas, T., Niederee, C., and Nejdl, W. (2013). A blocking framework for entity resolution in highly heterogeneous information spaces. *IEEE TKDE*, 25(12):2665–2682.

Papadakis, G., Koutrika, G., Palpanas, T., and Nejdl, W. (2014). Meta-blocking: Taking entity resolutionto the next level. *IEEE TKDEFherna*, 26(8):1946–1960.

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

Shannon, C. E. (2001). A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55.

Simonini, G., Bergamaschi, S., and Jagadish, H. (2016). Blast: a loosely schema-aware meta-blocking approach for entity resolution. *VLDB*, 9(12):1173–1184.

Whang, S. E., Menestrina, D., Koutrika, G., Theobald, M., and Garcia-Molina, H. (2009). Entity resolution with iterative blocking. In *ACM SIGMOD*, pages 219–232.

Wilbur, W. J. and Sirotkin, K. (1992). The automatic identification of stop words. *Journal of information science*, 18(1):45–55.