Avaliando o Processo de Seleção de Características na Tarefa de Junção de Similaridade

Lucas Romeiro Silva[1], **Dimas Cassimiro Nascimento**[1,2]

[1]Universidade Federal do Agreste de Pernambuco
[2]Universidade Federal de Campina Grande

lucas_romeiro@outlook.com,dimas.cassimiro@ufape.edu.br

Abstract. *Similarity join is the process of identifying pairs of similar records in one or more datasets. Since this task usually produces a significant amount of comparisons between records, it is important to employ filters that aim to limit the amount of comparisons produced. For doing so, it is necessary to determine which attributes will be explored by the filters. This work aims to propose and evaluate an incremental feature selection approach for the similarity join problem. The experimental results indicate that the investigated technique is promising, since specific combinations of attributes used in the similarity join resulted in a greater identification of pairs of similar records.*

Resumo. *A junção de similaridade consiste no processo de identificar pares de registros semelhantes em uma ou mais bases de dados. Uma vez que esta tarefa usualmente produz uma quantidade significativa de comparações entre registros, é importante empregar filtros que visem limitar a quantidade de comparações produzidas. Para tal, é necessário determinar quais atributos serão explorados pelos filtros. Este trabalho visa propor e avaliar uma técnica de seleção incremental de características para a tarefa de junção de similaridade. Os resultados experimentais obtidos indicam que a técnica investigada se mostra promissora, uma vez que combinações específicas de atributos na junção resultaram em uma maior identificação de pares de registros similares.*

1. Introdução

No mundo real os dados são bastante sujos. Inconsistências, tais como erros tipográficos valores ausentes, degradam a qualidade dos dados, o que pode acarretar em análises imprecisas e decisões de negócio erradas [Chu and Ilyas 2016]. A integração de dados é um passo essencial para a manutenção de data warehouses e repositórios de dados centralizados. Esta é uma das tarefas mais complexas do fluxo de trabalho de um cientista de dados. Sendo assim, agilizar essa tarefa é crucial para entregar resultados de análises em tempo hábil [Ribeiro et al. 2020].

Neste contexto, a busca e a junção de similaridade são tarefas importantes nos processos de limpeza e integração de dados. Derivadas das operações de busca e junção tradicionais, estes procedimentos estendem essas operações por tolerarem erros e inconsistências. Dado um objeto de consulta e uma coleção de dados, a busca de similaridade encontrará objetos similares à consulta, enquanto que na junção de similaridade, dadas duas coleções de dados, serão retornados todos os pares similares destas coleções. Existem diversos desafios [Yu et al. 2016] relacionados a estas tarefas, por exemplo: como executar estas tarefas visando atingir alta eficácia e eficiência?

Um algoritmo ingênuo (*naïve)* compara cada par de registros, produzindo uma quantidade de comparações usualmente proibitiva. Visando otimizar as tarefas de busca e junção por similaridade, métodos existentes usualmente empregam técnicas de filtragem. Dessa forma, busca-se, inicialmente, gerar um conjunto de pares de registros candidatos ao filtrar, de forma eficiente, o maior número de objetos dissemelhantes. Para tal, é empregado um filtro leve, com baixo custo computacional, para em seguida computar a similaridade entre os pares de registros candidatos. Este trabalho objetivou investigar o problema de selecionar quais combinações de atributos são mais relevantes para serem exploradas por filtros de tamanho e prefixo na tarefa de junção de similaridade, considerando bases de dados contendo múltiplos atributos.

Este artigo apresenta as seguintes contribuições: i) a formalização do problema de selecionar uma combinação de atributos para otimizar o processo de junção de similaridade; ii) uma abordagem para seleção de atributos para junção de similaridade inspirada em *incremental feature selection* [Yang et al. 2022]; e iii) uma avaliação experimental inicial que evidencia a aplicabilidade da abordagem proposta.

1.1. Trabalhos Relacionados

Nas últimas duas décadas, várias técnicas [Deng et al. 2014, Li et al. 2015] foram propostas com vista a agilizar o processo de junção de similaridade. Essas técnicas, no entanto, consideram objetos representados por apenas um atributo, enquanto que dados reais, em sua maioria, são compostos por múltiplos atributos. Nesse cenário, algumas abordagens podem ser adotadas, como, por exemplo, selecionar um atributo para computação de pares ou concatenar os valores de vários atributos para representar um objeto, o que, por sua parte, pode produzir resultados insatisfatórios. Para solucionar este problema, alguns autores têm proposto filtros de múltiplos atributos, visando melhorar o processo de identificação de pares similares via junção de similaridade [Ribeiro et al. 2020, do Carmo Oliveira et al. 2018].

A seleção de atributos, nessa circunstância, surge como uma maneira de eliminar atributos irrelevantes e/ou reduzir ruídos, selecionando, assim, um subconjunto de atributos capazes de aprimorar o reconhecimento de pares [Chandrashekar and Sahin 2014]. Ainda nessa circunstância, poderiam ser exploradas várias técnicas disponíveis, visto a possibilidade de aumentar o poder de configuração de tais algoritmos para cenários de atributos múltiplos, uma vez que, a depender do filtro e/ou função de similaridade empregada, pode-se encontrar soluções mais aptas a diferentes bases de dados [Jiang et al. 2014].

2. Definição do Problema

Nesta seção, são formalizados os conceitos de junção de similaridade e o problema investigado neste trabalho. Uma base de dados $\mathcal{D}$ é composta por um conjunto de registros, i.e., $\mathcal{D} = \{r_1, r_2, \cdots, r_n\}$. Cada registro $r_j \in \mathcal{D}$ é definido como um conjunto de pares $\langle a_i, v_i^j \rangle$, tal que a_i é um atributo pertencente ao esquema $A(\mathcal{D})$ e v_i^j é o valor associado ao atributo a_i do registro r_j. Sejam $\mathcal{D}_1$ e $\mathcal{D}_2$ duas bases de dados, sim uma função de similaridade do tipo $sim : \mathcal{D}_1 \times \mathcal{D}_2 \rightarrow [0, 1]$ e $\tau \in [0, 1]$ um limiar de similaridade, o objetivo da junção de similaridade é identificar todos os pares de registros $(r, r') \subseteq \mathcal{D}_1 \times \mathcal{D}_2$, tal que $sim(r, r') > \tau$. Na Definição 1, é formalizado o problema investigado neste artigo.

Definição 1 (Seleção de Características para Junção de Similaridade)**.** *Sejam $\mathcal{D}_1$ e $\mathcal{D}_2$ duas bases de dados, sim uma função de similaridade, $\tau \in [0, 1]$ um limiar de similaridade e $\mathcal{F}$ um filtro (de tamanho ou prefixo). O problema consiste em:*

Selecionar	$A^* \subset 2^{A(\mathcal{D}_1) \cap A(\mathcal{D}_2)}$
Executar	*Junção de Similaridade sobre $\mathcal{D}_1$, $\mathcal{D}_2$ usando $\langle \mathcal{F}, A^*, sim, \tau \rangle$*
Gerar	*$\mathcal{S} \subseteq \mathcal{D}_1 \times \mathcal{D}_2$, tal que $\forall (r, r') \in S : sim(r, r') > \tau$*
Mazimizar	$\lvert S \rvert$
Minimizar	*número de comparações realizadas entre registros*

3. Abordagem Proposta

A abordagem proposta para seleção de características para junção de similaridade é inspirada na ideia de seleção de características de forma incremental [Yang et al. 2022], a qual consiste em buscar por uma combinação de características otimizada de forma iterativa, expandindo gradativamente a quantidade de características consideradas na busca. A solução proposta é formalizada no Algoritmo 1. O algoritmo recebe como parâmetros de entrada duas bases de dados, o tamanho da amostra a ser avaliada, um limiar de similaridade, uma função de similaridade, um filtro (de tamanho ou prefixo) a ser empregado na junção de similaridade e uma função objetivo o. Inicialmente, são declaradas duas variáveis para armazenar a melhor qualidade e solução (combinação de atributos) encontradas, respectivamente (linhas 2 e 3). Em seguida, é selecionada uma amostra das bases de dados de tamanho s (linha 4) e calculado o conjunto dos atributos disponíveis em ambas as bases de dados (linha 5).

Então, na linha 6, para cada valor de i entre 1 e a quantidade de atributos disponíveis ($|\mathcal{A}|$), o algoritmo realiza as seguintes operações: i) gera uma combinação de atributos (A') contendo i atributos (linha 7); ii) executa a junção de similaridade utilizando a amostra $sample$ e a combinação de atributos A' e retorna a quantidade de comparações realizadas entre registros e o conjunto de pares similares, os quais são armazenados nas variáveis C e S, respectivamente (linha 8); iii) se a qualidade da solução obtida ($o(S, C)$) é melhor do que a solução obtida até então (linha 10), então o algoritmo armazena a combinação de atributos A' em A^* (linha 11) e a qualidade da solução obtida em $BestQuality$ (linha 12). Na linha 13, é então verificado se a combinação de atributos que produziu a melhor qualidade (A^*) engloba i atributos, ou seja, a quantidade de atributos considerada na iteração atual. Se não for o caso, significa que não foi possível encontrar uma combinação de atributos melhor expandindo a melhor solução encontrada na iteração anterior e a busca é encerrada (linha 14). Finalmente, a combinação de atributos otimizada A^* é retornada na linha 15.

Para avaliar a qualidade da combinação de atributos no Algoritmo 1, a função objetivo o deve explorar a relação entre o custo (número de comparações realizadas entre registros - C) e o benefício (quantidade de pares similares identificados - S) obtidos a partir da execução da junção de similaridade utilizando a amostra selecionada e a combinação de atributos considerada (A'). Uma implementação básica da função objetivo consiste em calcular $o(S, C) = \frac{|S|}{C}$. Possíveis variações podem ser também exploradas ao atribuir um peso maior para algum dos parâmetros, por exemplo: $o(S, C) = \frac{\beta * |S|}{C}$, tal que $\beta > 1$, ou $o(S, C) = \frac{|S|^2}{C}$.

A configuração ideal da função objetivo depende do contexto no qual o processo de junção de similaridade é empregado. Em outras palavras, caso o benefício seja mais importante, o fator $|S|$ deve ser ponderado de maneira mais relevante. Caso contrário, se o custo representar o fator mais crítico, então C deve ser ponderado de maneira mais relevante.

Algorithm 1: Algoritmo para Seleção de Características na Tarefa de Junção de Similaridade

input : $\mathcal{D}_1, \mathcal{D}_2$: bases de dados, s: tamanho da amostra, τ: limiar de similaridade, sim: junção de similaridade, $\mathcal{F}$: filtro (de tamanho ou prefixo), o: função objetivo
output: A^*: subconjunto de atributos

```
1  begin
2      BestQuality ⟵ 0
3      A* ⟵ ∅
4      sample ⟵ sampling(D1, D2, s)
5      A ⟵ A(D1) ∩ A(D2)
6      for i = 1 to |A| do
7          foreach A' ∈ 2^A , s.t. |A| = i do
               // retorna quantidade de comparações entre registros realizadas e
                  o conjunto de pares selecionados
8              C, S ⟵SimJoin⟨sample, F, A', sim, τ⟩
               // calcula a qualidade produzida pela combinação de atributos A'
9              quality ⟵ o(S, C)
10             if quality > BestQuality then
11                 A* = A'
12                 BestQuality = quality
13         if |A*| < i then
14             break
15     return A*
```

4. Avaliação

Nesta seção, é apresentada uma avaliação experimental inicial no contexto da contribuição do trabalho. A avaliação inicial consiste em investigar a hipótese de que, em bases de dados reais, a utilização da combinação de atributos A', tal que $|A'| > 1$, permite a tarefa de junção de similaridade detectar mais pares de registros similares do que a utilização dos atributos individuais pertencentes ao conjunto A'. Foram utilizadas duas bases de dados reais: i) Amazon Prime-Netflix, englobando dados dos catálogos de filmes e séries disponíveis nos EUA em março/2023; e ii) *San Francisco*, que representa dados acerca do estágio dos vários projetos na cidade de San Francisco - EUA.

Para a configuração dos experimentos, foram considerados dois filtros: filtro de prefixo ($prefix\_size = 2$) e filtro de tamanho. Como *baseline*, foi empregada a técnica *naïve* para junção de similaridade, a qual compara todos os pares de registros provenientes do produto cartesiano das bases de dados recebidas como entrada. A função $Jaccard$ foi empregada para calcular a similaridade entre os pares de registros, verificando se a média de similaridade entre os valores dos atributos disponíveis em $\mathcal{A}'$ superam o valor de $\tau = .8$. Os algoritmos foram implementados em Python empregando a tecnologia *Jupyter Notebook* e a implementação pode ser acessada em: *ommited for blind review*.

Os resultados experimentais são apresentados na Figura 1. Como mostrado na Figura 1(a), a quantidade de pares similares identificados empregando os dados da base *Amazon Prime-Netflix* tende a ser superior quando mútiplos atributos são empregados, em especial quando os atributos específicos são considerados (e.g., *genre* + *ti-*

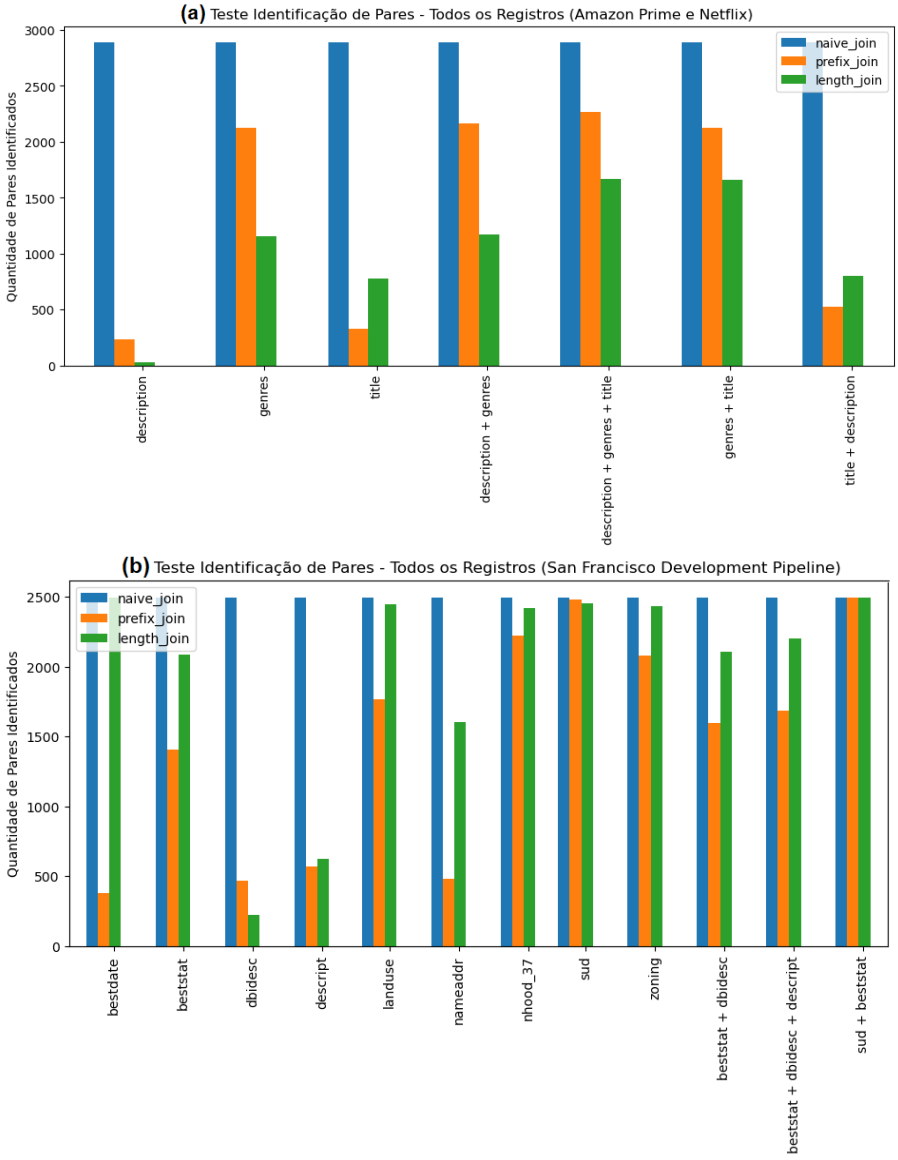

Figure 1. Resultado de eficácia da tarefa de junção de similaridade, considerando as seguintes bases de dados: (a) *netflix-amazon*; e (b) *San Francisco*.

tle). Combinações específicas de atributos corroboraram para um aumento na quantidade de pares similares identificados, especialmente quando o filtro de tamanho é considerado. Em relação à eficiência, a utilização dos filtros foi capaz de reduzir o tempo de execução da tarefa de junção de similaridade entre 60%-80% (filtro de tamanho) e 50%-88% (filtro de prefixo), a depender da combinação de atributos empregada. Os resultados obtidos a partir da base de dados *San Fracisto* (Figuras 1-b) também apontam para um comportamento semelhante, no qual combinações específicas de atributos (por ex., *beststat+dbidesc+descript*) permitiram a identificação de uma quantidade maior de pares de registros similares (tanto na amostra quanto processando um conjunto maior dos dados). Em relação à eficiência, os filtros de prefixo e de tamanho permitiram a execução da junção de similaridade de maneira muito mais eficiente, melhorando o tempo de execução em $25\% - 68\%$, dependendo do tipo de filtro e da combinação de atributos. Os resultados

experimentais obtidos indicam a ocorrência de resultados práticos em que combinações específicas de atributos melhoram a quantidade de pares de registros similares identificados, o que evidencia a relevância e grau de aplicabilidade da abordagem proposta.

5. Conclusões e Trabalhos Futuros

Neste trabalho, realizamos um esforço inicial no sentido de investigar o problema de seleção de características relevantes para a tarefa de junção de similaridade. Inicialmente, o problema investigado foi definido formalmente e foi proposta uma solução para o problema inspirada nas técnicas de *incremental feature selection*. Então, uma avaliação experimental inicial é apresentada, a qual buscou identificar se, em bases de dados reais, existem combinações de atributos otimizadas que produzem melhores resultados na junção de similaridade do que a utilização de atributos individuais. Os resultados experimentais apontam que, em muitos casos, a utilização de combinações de atributos aumenta a quantidade de pares de registros similares detectados.

Como trabalhos futuros, pretende-se avaliar a abordagem para seleção de características proposta considerando uma quantidade maior de bases de dados e explorando diversas variações de funções objetivo e funções de similaridade.

References

Chandrashekar, G. and Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28.

Chu, X. and Ilyas, I. F. (2016). Qualitative data cleaning. *Proceedings of the VLDB Endowment*, 9(13):1605–1608.

Deng, D., Li, G., and Feng, J. (2014). A pivotal prefix based filtering algorithm for string similarity search. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 673–684.

do Carmo Oliveira, D. J., Borges, F. F., Ribeiro, L. A., and Cuzzocrea, A. (2018). Set similarity joins with complex expressions on distributed platforms. In *Advances in Databases and Information Systems: 22nd European Conference, ADBIS 2018, Budapest, Hungary, September 2–5, 2018, Proceedings 22*, pages 216–230. Springer.

Jiang, Y., Li, G., Feng, J., and Li, W.-S. (2014). String similarity joins: An experimental evaluation. *Proceedings of the VLDB Endowment*, 7(8):625–636.

Li, G., He, J., Deng, D., and Li, J. (2015). Efficient similarity join and search on multi-attribute data. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1137–1151.

Ribeiro, L. A., Borges, F. F., and do Carmo Oliveira, D. J. (2020). A framework for set similarity join on multi-attribute data. In *SBBD*, pages 61–72.

Yang, Y., Chen, D., Zhang, e., Ji, Z., and Zhang, Y. (2022). Incremental feature selection by sample selection and feature-based accelerator. *Applied Soft Computing*, 121:108800.

Yu, M., Li, G., Deng, D., and Feng, J. (2016). String similarity search and join: a survey. *Frontiers of Computer Science*, 10:399–417.