Proceeding Series of the Brazilian Society of Computational and Applied Mathematics, v. 11, n. 1, 2025.

Trabalho apresentado no XLIII CNMAC, Centro de Convenções do Armação Resort - Porto de Galinhas - PE, 2024

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

Medidas de Similaridade Aplicadas à Vinculação Automática de Dados com Base em Nomes: Um Estudo de Caso na Área da Saúde

Ricardo da S. Santos[1]
IMECC/UNICAMP, Campinas, SP
Murilo G. Gazzola[2]
FCI/UPM, São Paulo, SP
Renato T. Souza[3], Rodolfo de C. Pacagnella[4]
FCM/UNICAMP, Campinas, SP
Cristiano Torezzan[5]
FCA/UNICAMP, Limeira, SP

Resumo. A integração de bases de dados é um desafio comum em diversas aplicações de ciência de dados. Frequentemente, a vinculação é realizada por meio de campos de texto livre, como nomes de pessoas, que costumam apresentar inconsistências devido a erros de grafia, abreviações e outras variações. Essas inconsistências podem resultar na representação de um mesmo indivíduo por diferentes registros, dificultando a identificação e análise precisa das informações. Dada a relevância desse problema, várias técnicas têm sido propostas para a vinculação de bases de dados. No entanto, esses estudos têm alto custo, em virtude da necessidade de validação manual qualificada. Este estudo tem como objetivo principal investigar a viabilidade de métodos automáticos de vinculação de dados aplicados no idioma português brasileiro. Os resultados da vinculação automática foram validados manualmente por uma equipe de especialistas, os quais atestaram a viabilidade técnica. Todas as alternativas testadas obtiveram um índice F1-Score superior a 0,90.

Palavras-chave. Distância de Levenshtein, Distância de Jaro, Distância de Jaro-Winkler, Similaridade entre Nomes

1 Introdução

Integrar bases de dados sem a existência de um identificador único é um desafio recorrente em várias aplicações de ciência de dados. Em muitos casos, essa integração depende de campos abertos, como os nomes de pessoas, para estabelecer os relacionamentos. Com o crescente avanço de métodos de aprendizado de máquina, essa tarefa tem se tornado cada vez mais importante, especialmente em áreas com vasta disponibilidade de dados não estruturados, como a medicina.

Dada a relevância deste problema, diversos métodos têm sido propostos para integração (*linkage*) automático, ou probabilístico, de dados. Essas abordagens buscam evitar o alto custo envolvido na integração manual de dados, que usualmente requer a intervenção de especialistas, o que também limita a escalabilidade das soluções [1].

[1]r263150@dac.unicamp.br
[2]gazzola@alumni.usp.br
[3]rtsouza@g.unicamp.br
[4]rodolfop@unicamp.br
[5]torezzan@unicamp.br

Uma característica fundamental dos métodos automáticos para vinculação de dados é a definição de medidas de similaridade entre dois campos de dados, ou de *strings*. Embora exista na literatura algumas medidas usualmente utilizadas para essas aplicações, há poucos trabalhos que investigam a influência na escolha das medidas de similaridade na acurácia de métodos de vinculação de dados, especialmente no contexto do idioma português brasileiro.

Nesse contexto, o objetivo deste trabalho é avaliar o desempenho de três métodos para calcular similaridade entre *strings* que são amplamente reconhecidos na literatura: similaridade de Levenshtein [2], similaridade de Jaro [3] e similaridade de Jaro-Winkler [4] numa tarefa específica de relacionamento de bases de dados de saúde em língua portuguesa, tendo como campo-chave o nome da mãe.

2 Definição do Problema

Este estudo está inserido no âmbito de uma pesquisa conduzida por uma equipe de médicos especialistas em saúde materna, cujo objetivo é avaliar o desempenho no gerenciamento oportuno de gestações com fetos pequenos para a idade gestacional e/ou com restrição do crescimento fetal. Para tanto, é necessário estabelecer uma relação precisa entre o acompanhamento da gestação e o nascimento do bebê.

Embora essa identificação possa parecer simples à primeira vista, esses dois conjuntos de dados são gerados de forma independente por fontes distintas de informação. Enquanto os relatórios de nascidos vivos são de responsabilidade da maternidade, os laudos de ultrassom são gerados por clínicas de acompanhamento do pré-natal, não necessariamente integradas à base de dados da maternidade.

O estudo é conduzido com base em dados reais, de um município do interior de São Paulo e a única variável em comum entre das duas fontes de dados é nome da mãe. Diante dessa necessidade, é crucial recorrer a métodos probabilísticos para a comparação de nomes, ou *strings*. Uma forma de abordar essa tarefa consiste em estabelecer uma medida de similaridade para cada par de nomes, um de cada banco de dados. Para isso, calcula-se um índice de similaridade que varia de 0 (totalmente diferente) a 1 (totalmente igual). Se o valor calculado ultrapassar um limiar estabelecido, δ, presume-se que os dados correspondem entre si.

Neste trabalho investigamos a influência de três medidas de similaridade usualmente utilizadas para esse tipo de aplicação, no contexto de integração de dados no idioma português brasileiro. As medidas que serão utilizadas neste trabalho estão descritas na próxima seção.

3 Similaridade entre *Strings*

A correspondência entre *strings* é uma área fundamental em ciência da computação e campos relacionados, como a teoria de códigos e matemática discreta, onde a comparação e análise de sequências numéricas ou de caracteres são frequentemente necessárias. Formalmente podemos definir:

Definição 1. *Sejam* $\mathbf{a}$ *e* $\mathbf{b}$ *duas strings, ou seja, sequências de símbolos de um alfabeto finito* Σ*. Dizemos que* $\mathbf{a}$ *corresponde a* $\mathbf{b}$ *quando* $f(\mathbf{a}, \mathbf{b}) > \delta$*, onde* $f : \Sigma \times \Sigma \to [0, 1]$ *é uma função de distância, que quantifica a similaridade entre* $\mathbf{a}$ *e* $\mathbf{b}$*, e* δ *é um limiar de similaridade.*

3.1 Similaridade de Levenshtein

A distância de Levenshtein [2], é uma medida que quantifica a diferença entre duas *strings*. Ela representa o número mínimo de operações necessárias para transformar uma *string* na outra.

As operações permitidas são inserção, exclusão ou substituição de um caractere. A distância de Levenshtein entre duas *strings* a e b, com comprimentos $|a|$ e $|b|$ é dada por $lev_{a,b}(|a|,|b|)$, e pode ser encontrada através da seguinte recursão:

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j), & \text{se } \min(i,j) = 0; \\ \min(lev_{a,b}(i-1,j)+1, lev_{a,b}(i,j-1)+1, lev_{a,b}(i-1,j-1)+\delta), & \text{se } \min(i,j) > 0. \end{cases}$$

onde $\delta = 1$ se $(a_i \neq b_j)$ e $\delta = 0$ se $(a_i = b_j)$, e $\text{lev}_{a,b}(i,j)$ é a distância entre os i primeiros caracteres de a e os primeiros j caracteres de b.

A distância de Levenshtein permite estabelecer a Similaridade de Levenshtein, que avalia a semelhança entre duas *strings*, considerando não apenas as diferenças específicas de caracteres, mas também o comprimento integral das *strings*. A a Similaridade de Levenshtein é dada por:

$$Lev(a,b) = 1 - \frac{2.lev_{a,b}(|a|,|b|) - ||a| - |b||}{|a| + |b|}$$

3.2 Similaridade de Jaro

A similaridade de Jaro é uma medida estatística de similaridade entre duas *strings*, que considera a proporção de caracteres coincidentes e a similaridade das posições desses caracteres, dada por:

$$J(a,b) = \frac{1}{3}\left(\frac{m}{|a|} + \frac{m}{|b|} + \frac{m-t}{m}\right),$$

onde $|a|$ e $|b|$ são os comprimentos das *strings* a e b, respectivamente. m é o número de caracteres que coincidem entre as duas *strings*. t é a metade do número de transposições.

3.3 Similaridade de Jaro-Winkler

A similaridade de Jaro-Winkler é uma extensão da similaridade de Jaro, que enfatiza correspondências nos primeiros caracteres das *strings* e é calculada pela fórmula:

$$J_W(a,b) = J(a,b) + l \cdot p \cdot (1 - J(a,b)),$$

onde l é um fator de escala, geralmente entre 0 e 0.25 e p é o comprimento do prefixo comum das *strings*, representando o número de caracteres iniciais idênticos. Essencialmente, quanto maior o valor de p, maior a probabilidade de igualdade entre as palavras, particularmente se compartilham um início semelhante.

3.4 Exemplo

A título de exemplo, considere as *strings* a = 'TATIANE' e b = 'TATYANNA'. Vamos determinar a similaridade entre as duas *strings* usando as três abordagens apresentadas acima.

Para determinar a similaridade de Levenshtein, precisamos inicialmente calcular a distância de Levenshtein, ou seja, precisamos determinar o número de edições mínimas (inserção, exclusão ou substituição) para transformar 'TATIANE' em 'TATYANNA'. Como são necessárias 3 alterações para transformar uma *string* na outra, temos que a distância de Levenshtein entre "TATIANE" e 'TATYANNA' é 3, logo podemos determinar a similaridade de Levenshtein: $Lev(a,b) = 1 - \frac{2.3-|7-8|}{7+8} = 1 - \frac{5}{15} = 0,6667$.

Na similaridade de Jaro, primeiro precisamos determinar os caracteres coincidentes entre as *strings*. Observe que nas duas *strings* temos os caracteres em comum 'T','A','T','A','N', logo $m = 5$, além

disso, esses caracteres aparecem na mesma ordem em ambas as *strings*, portanto, $t = 0$. Diante disso, temos que: $J(a,b) = \frac{1}{3}\left(\frac{5}{7} + \frac{5}{8} + \frac{5-0}{5}\right) = 0,7798$.

A similaridade de Jaro-Winkler, além de considerar a Similaridade de Jaro, atribui um 'bônus"pela quantidade inicial de caracteres iguais, no nosso caso, as duas palavras iniciam com o prefixo 'TAT', ou seja, temos $p = 3$, e considerando $l = 0,10$, temos que $J_W(a,b) = 0,7798 + 0,1 \cdot 3 \cdot (1 - 0,7798)) = 0,8458$.

4 Metodologia

4.1 Tratamento dos Dados

Os registros de Nascidos Vivos (RNV) utilizados neste estudo foram obtidos em um hospital do interior de São Paulo que é responsável por realizar todos os partos do Sistema Único de Saúde (SUS) do município. Os laudos ultrassonográficos foram obtidos de uma clínica situada na mesma localidade, a qual conduz todos os exames de ultrassom do Sistema Único de Saúde (SUS).

Para este experimento, foi selecionado aleatoriamente um mês que continha 277 laudos ultrassonográficos, sendo 276 com nome da mãe completo. Todos os laudos de ultrassonografias foram então confrontados com os registros de Nascidos Vivos do hospital nos nove meses subsequentes, que continha 2897 registros, sendo 121 registros com nomes de mãe repetido (casos de gêmeos) e 3 registros inválidos. Portanto foram, analisados 2773 registros válidos.

Como parte do pré-processamento dos dados, realizamos a remoção de acentos e espaços em branco dos nomes. Além disso, todas as letras foram convertidas para minúsculas, visando padronizar a representação dos nomes.

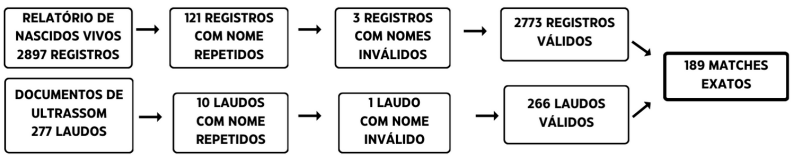

Figura 1: Organograma dos Bancos de Dados. Fonte: dos autores

4.2 Métricas Avaliadas

Inicialmente, uma equipe de especialistas realizou uma integração manual meticulosa que foi considerada como o padrão-ouro para a comparação dos métodos automáticos. A figura abaixo representa o que seria esperado como o padrão-ouro, baseado na análise conduzida pelos especialistas.

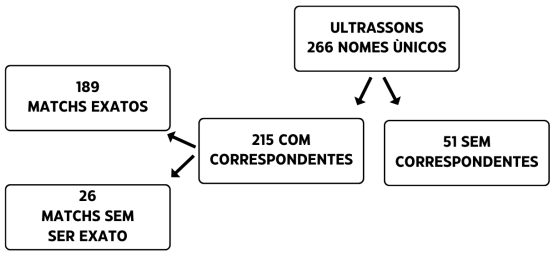

Figura 2: Padrão-ouro. Fonte: dos autores

Considerando que neste experimento removemos duplicidades de nomes tanto dos laudos de ultrassom quanto do Relatório de Nascidos Vivos, o esperado no padrão-ouro seria encontrar apenas uma correspondência correta entre os laudos de ultrassom e o Relatório de Nascidos Vivos. No entanto, é comum que problemas de correspondência retornem dois ou mais nomes como correspondentes, mesmo acima do limiar de similaridade. Diante dessa situação, é prática comum avaliar o desempenho dos algoritmos de similaridade por meio de métricas como Precisão, *Recall* e F1-Score [2]. Essas métricas podem ser definidas como: **Precisão**: A precisão é a proporção de verdadeiros positivos (capturas corretas) em relação ao número total de capturas feitas pelo algoritmo. É uma medida de quão precisas são as capturas feitas pelo algoritmo.

$$\text{Precisão} = \frac{\text{Número de capturas corretas}}{\text{Número total de capturas}}$$

Recall: O *recall* é a proporção de verdadeiros positivos em relação ao número total de casos verdadeiros na base de validação manual. É uma medida de quão completa é a captura de todos os casos relevantes.

$$\textit{Recall} = \frac{\text{Número de capturas corretas}}{\text{Total de casos verdadeiros na base de validação manual}}$$

F1-score: É a média harmônica entre precisão e *recall*. Ele fornece uma única medida que leva em consideração tanto a precisão quanto o *recall*.

$$\text{F1-score} = 2 \times \frac{\text{Precisão} \times \textit{Recall}}{\text{Precisão} + \textit{Recall}}$$

5 Resultados

Na Tabela 1, realizamos uma análise do desempenho dos algoritmos que utilizaram as medidas de similaridade de Levenshtein, Jaro e Jaro-Winkler, todos testados com limiares de similaridade de $0,90$ e $0,95$.

Os testes foram conduzidos utilizando as bibliotecas **fuzzywuzzy** e **python-Levenshtein** em Python. Para a medida de Jaro-Winkler, empregamos o fator de escala padrão $l = 0,10$ da biblioteca **python-Levenshtein**.

Adicionalmente, foram realizados testes que incluíram a comparação não apenas dos nomes completos, mas também dos primeiros nomes. Entre esses testes, o método que combinou uma similaridade de $\delta = 0,90$ pelo algoritmo de Levenshtein para os nomes completos e um $\delta = 0,80$ também pelo algoritmo de Levenshtein para os primeiros nomes obteviram o melhor desempenho, alcançando um F-1 Score de $0,9786$. Em outras palavras, dos 215 nomes de ultrassom que deveriam

DOI: 10.5540/03.2025.011.01.0426

ser capturados, esse método conseguiu identificar corretamente 206 nomes, sem cometer nenhum erro ao atribuir falsas correspondências.

Todos os testes de comparação dos primeiros nomes foram realizados exclusivamente com a medida de similaridade de Levenshtein. Essa escolha se deve ao fato de que os outros métodos são menos sensíveis a pequenas variações em nomes curtos, como é o caso dos primeiros nomes.

Tabela 1: Métricas de Desempenho - Conjunto Completo de Dados.

Método	**Precisão**	***Recall***	**F-1 Score**
LEVENSHTEIN 95%	1,0000	0,9163	0,9563
LEVENSHTEIN 90%	0,9809	0,9581	0,9694
JARO 95%	1,000	0,9023	0,9486
JARO 90%	0,9585	0,9674	0,9630
JARO-WINKLER 95%	0,9950	0,9395	0,9665
JARO-WINKLER 90%	0,8452	0,9907	0,9122
LEVENSHTEIN 90% e PRIMEIRO NOME 80%	1,0000	0,9581	0,9786
LEVENSHTEIN 80% e PRIMEIRO NOME 80%	0,9106	0,9953	0,9511
JARO WINKLER 90% e PRIMEIRO NOME 80%	0,8838	0,9907	0,9342

Embora seja comum na literatura a utilização dos dados integrais conforme apresentado na Tabela 1, essa abordagem não reflete totalmente a eficácia dos métodos de similaridade entre *strings*. Em nosso caso específico, dos 189 casos em que houve uma correspondência total, todos os métodos de similaridade proporcionam uma captura precisa. No entanto, a análise apresentada na Tabela 2 se concentra exclusivamente nos casos em que não há correspondência total. Em outras palavras, os dados da Tabela 2 buscam avaliar a eficiência dos métodos em identificar os 26 nomes que não são idênticos entre o relatório de Nascidos Vivos e os laudos de ultrassom.

Tabela 2: Métricas de Desempenho - Desconsiderando os Matches Exatos.

Método	**Precisão**	***Recall***	**F-1 Score**
LEVENSHTEIN 95%	1,0000	0,3077	0,4706
LEVENSHTEIN 90%	0,8095	0,6538	0,7233
JARO 95%	1,0000	0,1923	0,3226
JARO 90%	0,6786	0,7308	0,7037
JARO WINKLER 95%	0,9286	0,5000	0,6500
JARO WINKLER 90%	0,3809	0,9231	0,5392
LEVENSHTEIN 90% e PRIMEIRO NOME 80%	1,0000	0,6538	0,7907
LEVENSHTEIN 80% e PRIMEIRO NOME 80%	0,5435	0,9615	0,6944
JARO WINKLER 90% e PRIMEIRO NOME 80%	0,4615	0,9231	0,6153

Ao analisar a Tabela 2, observamos que o melhor método continua sendo aquele que combina uma similaridade de $\delta = 0,90$ pelo algoritmo de Levenshtein para os nomes completos e um $\delta = 0,80$ também pelo algoritmo de Levenshtein, alcançando um F1-Score de $0,7907$. Em outras palavras, dos 26 nomes do padrão-ouro, esse método encontrou 17 sem cometer nenhum erro de falsa correspondência.

No entanto, ao examinar a Tabela 2, métodos como Levenshtein com similaridade $\delta = 0,95$ não se destacam tanto. Embora não cometam nenhum erro de correspondência, conseguem capturar apenas 8 dos 28 nomes. Nesse contexto, o método de Jaro-Winkler com $\delta = 0,95$ apresenta um F1-Score de $0,7037$, com uma precisão de $0,8566$ (capturando corretamente 13 de 26 nomes) e apenas uma correspondência errada.

DOI: 10.5540/03.2025.011.01.0426

Os métodos que demonstraram maior *recall*, ou seja, conseguiram capturar mais dados, foram aqueles que utilizaram Levenshtein com $\delta = 0,80$ tanto para o primeiro nome quanto para o nome completo. Esse método conseguiu capturar 25 dos 26 nomes, atingindo um *recall* de 0,9615. No entanto, é importante ressaltar que esse método cometeu 19 falsas correspondências.

6 Conclusões

Neste trabalho realizamos a comparação entre o desempenho de diferentes métodos para cálculo de similaridade entre *strings* com o objetivo de fazer a vinculação de duas bases de dados para uma aplicação em saúde da mulher. A comparação entre os resultados computacionais e uma análise manual por especialistas mostrou que a técnica pode ser eficiente para essa aplicação. Nossos resultados mostram que, para os dados analisados, a combinação de uma similaridade de Levenshtein com $\delta = 0,90$ para os nomes completos e um $\delta = 0,80$ para a similaridade de Levenshtein para os primeiros nomes apresentou o melhor desempenho, alcançando um F1-Score de $0,9786$ nos dados completos e de $0,7907$ nos dados que não possuem correspondência exata.

Além do F1-Score, que é uma medida importante, dependendo do contexto e dos objetivos específicos da aplicação outras métricas como o *recall* e a precisão podem ser igualmente cruciais. Além disso, a validação humana agrega um valor significativo à pesquisa, mostrando que os métodos para determinação que os métodos de similaridades entre *strings* podem ser viáveis para comparação do nome de pessoas em banco de dados brasileiros.

Em estudos futuros, podemos explorar o uso de métodos de aprendizado de máquina para ajustar os parâmetros de limiar tanto do índice de similaridade, quanto do fator l no método de Jaro-Winkler, obtendo índices de correspondência ainda melhores. Além disso, podemos explorar testes em bases de dados diferentes ou em períodos distintos; no entanto, um empecilho é a necessidade de mais validação humana.

Referências

[1] A. K. Gupta, S. N. Kasthurirathne, H Xu, X. Li, M. M. Ruppert, C. A. Harle e S. J. Grannis. "A framework for a consistent and reproducible evaluation of manual review for patient matching algorithms". Em: **Journal of the American Medical Informatics Association**. Vol. 29. Oxford University Press, 2022, pp. 2105–2109. DOI: 10.1093/jamia/ocac175. URL: https://doi.org/10.1093/jamia/ocac175.

[2] M. Hadwan, M. A. Al-Hagery, A. M. Sanabani e S. Al-Hagree. "Soft Bigram Distance for Names Matching". Em: **PeerJ Computer Science**. Vol. Vol. 7. 2021. DOI: 10.7717/peerj-cs.465.

[3] M. A. Jaro. "Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida". Em: **American Statistical Association, Taylor Francis, Ltd.** Vol. 84. 1989, pp. 414–420. DOI: https://doi.org/10.2307/2289924.

[4] W. E. Winkler. "String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage." Em: **Proceedings of the Section on Survey Research**. 1990, pp. 354–359.

