# Comparação de Performance entre o Coeficiente de Jaccard e a Distância de Levenshtein na Detecção de Similaridade em Textos

**Arthur Rodrigues Batista**[1]

[1]Departamento de Informática – Universidade Estadual de Maringá (UEM)
Maringá – PR – Brasil

ra105422@uem.br[1]

***Abstract.*** *This paper contrasts two strategies, Jaccard Coefficient and Levenshtein Distance, in order to analyze a case study to verify the efficiency of both when calculating the similarity rate between texts. The objective was to observe and compare the performance of the techniques using previously stipulated measures for four pairs of texts, and to infer which one had the best outcome according to the established metrics. Throughout this work, the functioning of each technique is defined as well as the fundamental concepts necessary for full understanding. At the end, there is a discussion about the analyzed results and it is evident why Levenshtein's strategy stood out in the analyzed parameters.*

***Resumo.*** *Este trabalho contrapõe duas estratégias, Coeficiente de Jaccard e Distância de Levenshtein, a fim de analisar um estudo de caso para verificar a eficiência de ambas ao calcular a taxa de similaridade entre textos. O Objetivo foi observar e comparar o desempenho das técnicas utilizando medidas previamente estipuladas para quatro pares de textos, e inferir qual teve maior aproveitamento de acordo com as métricas estabelecidas. Ao decorrer deste trabalho, define-se o funcionamento de cada técnica bem como os conceitos fundamentais necessários para o entendimento pleno. Ao final, faz-se uma discussão acerca dos resultados analisados e evidência-se por que a estratégia de Jaccard sobressaiu-se nos parâmetros analisados.*

## 1. Introdução

A taxa de similaridade textual é utilizada em inúmeras aplicações no âmbito da teoria da informação, em que se busca reconhecer o contexto, fazer uma análise sintática, semântica, léxica e morfológica, sob a perspectiva de uma ou mais estruturas textuais. Denota-se essa área como sendo processamento natural de linguagens, subárea da Inteligência Artificial. Uma das razões pelos quais se busca otimizar métodos que computam taxas de similaridade é minimizar o tempo de consultar informações (*search engines*). A consulta visa classificar uma coleção de documentados de acordo com suas semelhanças com a entrada especificada por um usuário. Mesmo que os *search engines* sejam um enfoque inerente da área de similaridade entre textos, outros domínios também se fazem presentes, tais como *feedback* de relevância, classificação de texto, métodos para avaliação automática de tradução de idiomas e coerência do textual [Lv and Zhai 2009].

Como visto, há grande estimulo na área de classificação de semelhanças textual no âmbito da tecnologia da informação, tendo isso em vista, este trabalho visa realizar uma

análise quantitativa comparando duas metodologias distintas, a fim de inferir qual método teve maior aproveitamento com base nas métricas apresentadas na seção 2.1. A primeira técnica (Distância Levenshtein) utiliza uma abordagem tradicional baseado em caracteres, em que se compara a sequência de letras em dois textos. A segunda (Coeficiente de Jaccard) é baseada em termos, neste caso, a comparação é realizada mediante uma relação da quantidade de palavras comuns entre ambos os textos.

As duas estratégias foram sujeitas ao mesmos casos testes (4 pares de textos), com medidas previamente calculadas pelo Plagiarism Checker[1], e executadas na mesma linguagem de programação, Racket. Após a efetivação dos testes, foram comparadas as saídas de ambas juntamente às medidas calculadas antemão e, ao final, foi constatado que o Coeficiente de Jaccard, sobressaiu-se com base nos parâmetros analisados.

## 2. Metodologia

Esta Seção será subdividida em duas subseções. Na primeira, será discutido acerca das peculiaridades de cada estratégia, isto é, Distância Levenshtein e Coeficiente de Jaccard. Em seguida, estabelecer-se-á as métricas de avaliação dos métodos, para que na Seção 4 seja feita uma análise dos resultados das execuções.

### 2.1. Estratégias

A taxa de similaridade de dois ou mais textos pode ser calculada de duas maneiras: sintaticamente, quando compartilham a mesma sequência de caracteres e semanticamente, quando carregam o mesmo significado (por exemplo, sinônimos). Neste trabalho, ter-se-á como enfoque o aspecto léxico.

#### 2.1.1. Distância Levenshtein

A distância editada (ou Distância de Levenshtein) entre dois textos é o número de inserções, exclusões e substituições necessárias para transformar um texto no outro[Levenshtein 1966]. Essa distância é de fundamental importância em vários campos, como biologia computacional e processamento/pesquisa de texto e, consequentemente, problemas envolvendo a Distância de Levenshtein foram estudados extensivamente. Na biologia computacional, por exemplo, a distância de edição e suas variantes são as medidas mais elementares de dissimilaridade para dados genômicos [Andoni and Onak 2011].

A Distância de Levenshtein entre dois pares de texto **a** e **b** (de comprimento **a** e **b** respectivamente) pode ser representado como $lev_{a,b}(\|a\|, \|b\|)$[2]:

$$lev_{a,b}(i,j) = \begin{cases} max(i,j), & \text{se } min(i,j) = 0 \quad (1) \\ min = \begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1_{(\mathrm{a}_i \neq b_j)} \end{cases} & \text{caso contrário.} \quad (2) \end{cases}$$

[1]https://copyleaks.com/

[2]https://dzone.com/articles/the-levenshtein-algorithm-1

em que $1_(a_i \neq b_j)$ é a função indicativa que assume o valor de 0 quando $(a_i = b_j)$ e 1 caso contrário.

A Distância de Levenshtein entre "FLOMAX"e "VOLMAX"é 3, uma vez que há três edições consecutivas: F por V, L por O e O por L e não se consegue transformar uma palavra em outra com menos edições.

### 2.1.2. Coeficiente de Jaccard

O índice de similaridade Jaccard[3] (às vezes chamado de coeficiente de similaridade Jaccard) compara elementos entre dois conjuntos para verificar quais destes são compartilhados e quais são distintos. É uma medida que simboliza uma taxa entre 0% e 100%, quanto maior a porcentagem, mais semelhantes os dois conjuntos.

Neste trabalho, o coeficiente de similaridade de Jaccard foi utilizado para mensurar a semelhança entre textos, pois como é concluído em [Niwattanakul et al. 2013], o índice de Jaccard é adequado o suficiente para ser empregado na medição de similaridade entre palavras. Com isso posto, o coeficiente pode ser expresso como $jac(A, B)$ em que **A** e **B** são o conjunto de palavras dos textos A e B, respectivamente:

$$jac(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{\|A\| + \|B\| - |A \cap B|} \tag{3}$$

## 2.2. Métricas de Avaliação

A primeira métrica estabelecida neste trabalho é referente à taxa de correlação das saídas dos métodos com as medidas mensuradas previamente. Na estatística, correlação ou associação é qualquer relação estatística, causal ou não, entre duas variáveis. No sentido mais amplo, refere-se ao grau em que um par de variáveis está linearmente relacionado. O coeficiente de correlação varia de -1 a 1. Se for igual a 1 isso implica que uma equação linear descreve perfeitamente a relação entre X e Y, ou seja, Y aumenta à medida que X aumenta. Caso o valor seja igual a -1, à medida que Y aumenta, X diminui. Em ambos os casos, os valores são fortemente relacionados, porém, se a correlação se aproximar de 0 isso significa que não há correlação linear entre as variáveis.

Essa medida foi adotada pelo fato de indicar um relacionamento preditivo que pode ser explorado na execução dos testes. A maneira mais tipicamente utilizada para calcular a correlação entre duas variáveis é o coeficiente de correlação de Pearson[Derrick et al. 1994], sendo expressa como:

$$r_{xy} = \frac{\sum_{n=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{n=1}^{n}(x_i - \bar{x})^2(y_i - \bar{y})^2}} \tag{4}$$

em que *n* é o tamanho da amostra, $x_i, y_i$ são os elementos contidos nas amostras **X** e **Y** e $\bar{x}, \bar{y}$ são as médias respectivas desses conjuntos.

Embora a taxa de correlação seja um parâmetro inicial de comparação, se analisada de maneira singular, não seria suficiente para inferir qual dos métodos têm maior

[3]https://www.statisticshowto.datasciencecentral.com/jaccard-index/

eficiência. Dessa forma, a próxima métrica definida é o tempo de execução de cada teste com o respectivo método. O motivo pelo qual se optou por utilizá-la se dá pelo fato de que, para solucionar os problemas que foram destacados anteriormente, deve-se pensar técnicas que resolvam em tempo hábil de processamento. Além disso, já que foi proposto inferir qual técnica demonstra maior desempenho, é imprescindível validar o tempo de execução, haja vista que, segundo os autores [Hennessy and Patterson 2002], a única medida consistente e confiável de desempenho é o tempo de execução de programas. Outras alternativas, embora sejam relevantes durante a análise de performance, podem levar a constatações enganosas ou até mesmo erros em projetos de softwares.

## 3. Resultados e Discussão

Nesta Seção será realizada uma análise referente à execução da gama de testes nas duas estratégias, de modo que seja possível explicar as causas dos resultados obtidos referentes a cada técnica, e categorizar qual destas possui um maior desempenho com base nas métricas já estabelecidas.

Como aspecto inicial de análise, discutir-se-á acerca da taxa de correlação entre a medida previamente calculada com as técnicas. A Figura 1 representa um comparativo da taxa de similaridade de Jaccard e Levenshtein nos pares de textos examinados. Por meio dela, é possível identificar que a similaridade dos pares 1 e 2 são mais próximos do esperado quando observado as saídas de Levenshtein, 0.809 e 0.800, 0.689 e 0.680, respectivamente. Embora que nos outros dois testes a taxa tenha sida mais favorável ao lado de Jaccard, a taxa de correlação com o esperado foi de 89% em contrate à 97% de Levenshtein. Este comportamento pode ser explicado pela maneira na qual estão dispostos os caracteres nos textos. Os pares 1 e 2 compartilham muitas posições de caracteres em comum, já que ambos foram exemplos de plágios diretos, ou seja, foram copiados trechos idênticos de um autor original. Os textos 3 e 4 foram de plágios mosaicos, ou seja, foram textos parafraseados, e sendo assim, a técnica de Levenshtein não se mostrou tão eficiente quanto o de Jaccard, mesmo que este não leve em consideração palavras que possuem significados iguais e escritas diferentes.

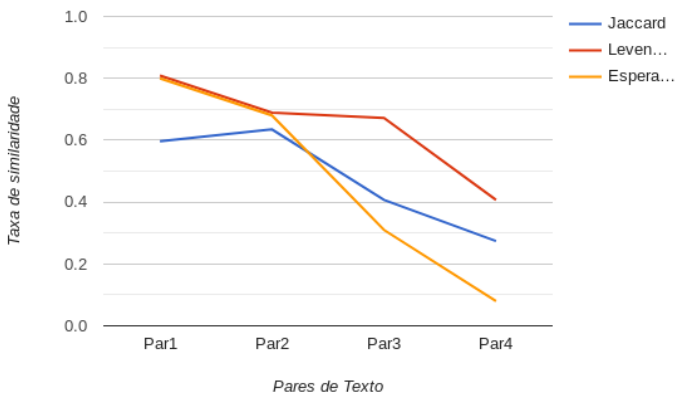


**Figura 1. Taxa de similaridade de Jaccard e Levenshtein com o resultado esperado**

Como já mencionado, se a taxa de correlação for analisada de maneira isolada, não é suficiente para concluir qual estratégia teve melhor desempenho. Atentando-se a isso, a próxima métrica que será analisada diz respeito ao tempo de execução de cada teste. A Figura 2 representa essa relação, isto é, dos testes executados (relacionado com a técnica respectiva) com seu tempo de execução.


**Figura 2. Tempo de execução de Jaccard e Levenshtein com os pares de testes**

Nota-se, conforme a Figura acima, que a estratégia de Levenshtein teve um tempo de processamento médio aproximadamente 4500 vezes pior que o Jaccard, isso pode ser explicado por meio da complexidade de computação necessária para executá-lo. O custo computacional do algoritmo de Levenshtein, utilizando programação dinâmica, é $O(m \times n)$, em que *m,n* são a quantidade de caracteres nos textos **M,N**, respectivamente [Backurs and Indyk 2014]. Supondo que ambos tenham o mesmo tamanho de comprimento, o tempo de complexidade no pior caso desse algoritmo é $O(n^2)$. Segundo os autores [Backurs and Indyk 2014], para entradas grandes essa técnica é inviável e soluções aproximativas ou heurísticas podem ser utilizadas para seu aprimoramento.

Isso posto, a complexidade de Jaccard pode ser expressa como sendo a quantidade de esforço computacional necessário para realizar uma operação de intersecção de dois conjuntos **A** e **B**, já que a soma interna dos conjuntos pode ser calculada em tempo linear. Assumindo que ambos têm uma quantidade *m* de elementos, aplica-se um algoritmo de ordenação de ordem $O(m \log(m)$ e é possível realizar a interseção em tempo $O(m \log(m)$. Assim, evidência-se ô porque o algoritmo de Jaccard teve um tempo significativamente mais rápido.

## 4. Conclusão

As estratégias para detecção de similaridade entre textos estão amplamente difundidas no âmbito da teoria da informação. A otimização desse processo contribui para a classificação de textos e mecanismos de buscas. Por essa razão, uma gama de testes e comparações faz-se necessária para comparar e aprimorar as diversas técnicas já existentes.

Neste trabalho foi contraposta duas estratégias, Coeficente de Jaccard e Distância Levenshtein. Ambas foram sujeitas aos mesmos casos teste e excetuadas na linguagem de programação, Racket. Conforme demonstrado na seção 3, a Distância de Levenshtein teve resultados surpreendentes quando exposta aos pares de textos que tiveram cópia direta de uma fonte original, beirando uma taxa de 100% de correlação com o resultado esperado, contraposto à 89% do Coeficiente de Jaccard. Embora o Coeficiente de Jaccard não tenha tido funcionamento quase perfeito nesse quesito, mostrou-se extremamente mais rápido durante sua execução e exibiu uma ótima taxa de correlação. Por conseguinte, para calcular a taxa de similaridade de pequenos textos, mostrou-se mais aconselhável que a Distância Levenshtein, pois retorna um resultado aceitável em um período de tempo muito menor.

Para futuros trabalhos, seria necessário executar uma gama mais alta de testes com mais variedade de conteúdos, já que neste trabalho só foi avaliado a questão sintática e com textos com cópia direta e parafraseados. Além disso, outras estratégias poderiam ser comparadas a fim de, de fato, averiguar uma melhor opção e realizar modificações de tal forma a otimizá-la.

## Referências


Andoni, A. and Onak, K. (2011). Approximating edit distance in near-linear time. *CoRR*, abs/1109.5635.

Backurs, A. and Indyk, P. (2014). Edit distance cannot be computed in strongly subquadratic time (unless SETH is false). *CoRR*, abs/1412.0348.

Derrick, T., Bates, B., and Dufek, J. (1994). Evaluation of time-series data sets using the pearson product-moment correlation coefficient. *Medicine and science in sports and exercise*, 26(7):919—928.

Hennessy, J. L. and Patterson, D. A. (2002). *Computer Architecture: A Quantitative Approach 3rd edition*. Morgan Kaufmann, ISBN 1-55860-724-2.

Levenshtein, V. I. (1966). Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707.

Lv, Y. and Zhai, C. (2009). Adaptive relevance feedback in information retrieval. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09, pages 255–264, New York, NY, USA. ACM.

Niwattanakul, S., Singthongchai, J., Naenudorn, E., and Wanapu, S. (2013). Using of jaccard coefficient for keywords similarity. In *Proceedings of the international multiconference of engineers and computer scientists*, volume 1, pages 380–384.