Implementação dos algoritmos Longest Common SubString (LCS) e Jaccard Similarity Coeficient para verificação de similaridade de textos

Beatriz de Jesus Costa[1]

[1]Departamento de Informática – Universidade Estadual de Maringá (UEM)
87.020-900 – Maringá – PR – Brasil

ra104361@uem.br

Resumo. *O presente trabalho visa a verificação de similaridade textual dentre oito textos, utilizando os métodos Longest Common SubString, que é Character-based, e Jaccard Similarity Coeficient, Term-based. O objetivo é avaliar a eficiência destes algoritmos na verificação de semelhança entre dois textos, tendo como parâmetro um objetivo esperado. Os algoritmos foram implementados utilizando a linguagem Racket e, após execução e analise dos resultados, observou-se que o algoritmo Jaccard obtivera uma maior correlação com o resultado esperado.*

Abstract. *The present work aims to verify the textual similarity between the two texts, using the Longest Common SubString method, which is based on characters and Jaccard Similarity Coeficient, Term-based. The objective is to evaluate the performance of these algorithms in the verification of similarity between two texts, having as parameter an expected objective. The algorithms were implemented using the Racket language, after execution and analysis of results, the Jaccard algorithm obtained a higher estimate than the expected result.*

1. Introdução

Na atualidade, com a publicação de documentos e trabalhos na internet, o reuso, referenciamento ou até cópia destas informações se tornou algo recorrente. Existem várias maneiras de se fazer o reuso de um texto, seja retirando um fragmento do texto e escrevendo-o exatamente igual, reformulando-o por meio da utilização de sinônimos e adição de informações ou escrevendo a mesma informação de maneira totalmente diferente [Bendersky and Croft 2009]. Para que possamos identificar o quanto dois textos são iguais, são utilizados algoritmos de verificação de similaridade textual. A similaridade textual consiste na comparação e medição do quanto dois ou mais textos são semelhantes.

A verificação de similaridade textual também é muito utilizada em outras coisas como recuperação de informações, classificação de texto, agrupamento de documentos, detecção de tópicos, rastreamento de tópicos, geração de perguntas, resposta à perguntas, pontuação de redação, pontuação de respostas curtas, tradução automática, resumo de texto e outros.

Um conjunto de palavras pode ser considerado similar tanto semânticamente como lexicalmente. Quando analisamos a semântica, vemos se foram usados sinônimos das palavras, opostos, se estão no mesmo contexto, entre outros. Ao analisarmos lexicalmente,

vemos se eles têm uma sequência de caracteres semelhante. Existem diversos métodos para realizar a verificação de similaridade textual, eles podem ser separados em léxicos e semânticos, sendo os léxicos Character-Based e Term-based, e os semânticos Corpus-Based e Knowledge-Based. [Gomaa and Fahmy 2013]

Neste trabalho serão implementados os algoritmos dos métodos Longest Common SubString (Character-Based) e Jaccard (Term-based), utilizando a linguagem Racket. Será feita uma avaliação entre os dois métodos e a verificação de similaridade e correlação de acordo com um resultado esperado.

2. Longest Common Substring

O método Longest Common Substring (LCS) é um método bastante popular para verificação de similaridade textual, ele consiste em, dadas duas Strings, encontrar a maior substring similar entre elas. Ele as compara e encontra a similaridade baseada na maior cadeia de caracteres iguais.

3. Jaccard Similarity Coeficient

O método Jaccard Similarity Coeficient, ou Jaccard Index, computa o número de termos sobre todos os termos diferentes nas duas strings, ou seja, é o tamanho da interseção das string dividido pelo tamanho da união destas.

4. Metodologia

A metodologia utilizada para verificação e avaliação da similaridade de textos foram os algoritmos Longest Common Substring e Jaccard Similarity Coeficient, implementados na linguagem funcional Racket. Para a realização da avaliação dos algoritmos, primeiramente foram realizados testes unitários em ambos, para assegurar seu correto funcionamento. Em seguida, foram estipulados as medidas do melhor resultado aproximado de similaridade entre as strings e, por fim, ambos algoritmos foram comparados com o resultado estipulado por meio de oito textos. Estes oito textos foram divididos em pares a serem comparados, gerando assim quatro resultados.

5. Métricas

Para a especulação do resultado esperado foi utilizado um resultado previamente calculado. Ambos algoritmos retornam um valor entre 0 e 1, sendo 0 para totalmente diferentes e 1 para totalmente iguais. Para a análise dos resultados dos algoritmos em comparação com o resultado esperado foi utilizada a função ”correlation”da linguagem Racket, que faz a correlação dos resultados dos algoritmos com os resultados esperados.

6. Resultados e Discussão

Ao executar ambos algoritmos e fazer a correlação com o resultado esperado, observou-se que o algoritmo Jaccard apresentou uma correlação maior com o resultado esperado, enquanto o Longest Common Substring permaneceu abaixo do esperado.

Como pode-se observar na figura 1, o algoritmo LCS possui oscilações, enquanto o Jaccard se mantém constante e ”seguindo”o resultado esperado. Na figura 2 podemos

observar melhor a diferença entre as correlações, sendo a correlação do LCS com o esperado igual a 0.287711404231489, e a correlação do Jaccard com o esperado igual a 0.704241640358447.

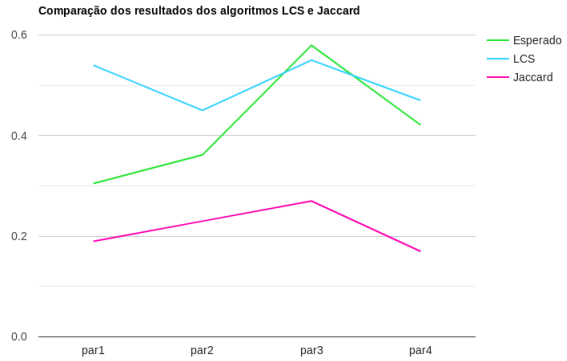

Figura 1. Comparação dos resultados dos algoritmos LCS, Jaccard e Esperado

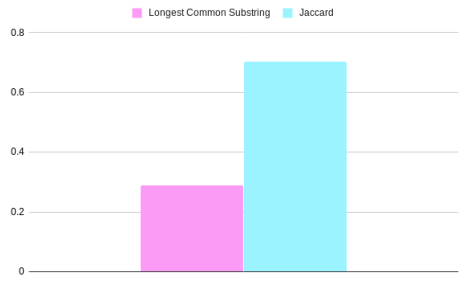

Figura 2. Correlação dos resultados dos algoritmos LCS e Jaccard

7. Conclusão

Após a análise dos resultados, pode-se concluir que ambos os métodos de verificação de similaridade são bons, porém dependem do tipo de texto que estamos analisando e do resultado esperado. Neste trabalho foram utilizados textos relacionados a um mesmo assunto, porém diferentes. A partir disso, o algoritmo Term-based demonstrou maior correlação com o resultado esperado.Conclui-se também que estes métodos são muito eficientes para a identificação da similaridade textual.

Referências

Bendersky, M. and Croft, W. B. (2009). Finding text reuse on the web. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, pages 262–271. ACM.

Gomaa, W. H. and Fahmy, A. A. (2013). A survey of text similarity approaches. *International Journal of Computer Applications*, 68(13):13–18.