Universidade Federal de Ouro Preto

Instituto de Ciências Exatas e Biológicas

Departamento de Computação

BCC241 – Projeto e Análise de Algoritmo

Trabalho Prático I

Desambiguação de nomes de autores

André Ribeiro de Brito

Mateus Freire

Gustavo Quintão

Introdução

Neste contexto, a ambiguidade de nomes de autores é um problema que prejudica a qualidade dos serviços de recuperação de informação bibliográfica fornecidos pelas bibliotecas digitais.

Este problema ocorre quando dois ou mais autores compartilham um mesmo nome (nomes homônimos) ou quando um autor utiliza diferentes nomes em suas referências bibliográficas (nomes sinônimos).

Os desafios ao lidar com este problema têm levado ao desenvolvimento de inúmeros métodos de desambiguação, baseando-se no proposito do trabalho uniremos três tipos de método de programação para desambiguação incremental baseando-se esses métodos de divisão e conquista, algoritmos gulosos e programação dinâmica, sendo se capaz de criar subconjunto de autores de cada registro.

Na avaliação experimental, foram obtidos resultado de baseados em métricas de precisão, revogação e f1.

Objetivo do trabalho

Dado um conjunto D de n registros com autores ambíguos (autores distintos que publicaram trabalhos com o mesmo nome ou o mesmo autor que publicou trabalhos com nomes diferentes), a tarefa de desambiguação consiste em dividir (particionar) o conjunto D em m subconjuntos, de tal forma que, cada subconjunto contenha (idealmente) todos e somente todos os registros publicados por um autor ambíguo.
O programa deve receber como entrada o arquivo “CChen.txt”, que contém publicações dos autores ambíguos “C. Chen”, e fornecer como saída os subconjuntos gerados e os resultados das métricas precisão, revogação e F1.

Técnicas de programação

As técnicas de programação utilizada são baseadas em:

Divisão e conquista: dado um problema é dividida em duas ou mais instâncias menores, cada instância menor e combinam as sobrepostas, as soluções das instâncias menores são combinadas para produzir uma solução da instância original. Sendo dividido em três partes, sendo elas:

*Dividir: o problema em determinado número de subproblemas.

*Conquistar os subproblemas, resolvendo-os recursivamente.
Se o tamanho do subproblema for pequeno o bastante, então a solução é direta.

*Combinar as soluções fornecidas pelos subproblemas, a fim de produzir a solução par a o problema original.

Algoritmos gulosos: é uma técnica de algoritmos para resolver problemas de otimização, sempre realizando a escolha que parece ser a mais promissora naquele instante, fazendo uma escolha ótima local, na esperança de que esta escolha leve até a solução ótima global.

Programação dinâmica: procura resolver o problema de otimização através da análise de uma sequência de problemas mais simples do que o problema original. A resolução do problema original de N variáveis é caracterizada pela determinação de uma variável e pela resolução de um problema que possua uma variável a menos (N-1). Este por sua vez é resolvido pela determinação de uma variável e pela resolução de um problema de N-2 variáveis e assim por diante

Analise assintótica

Analisando cada função do algoritmo, sendo essas funções: criar grafo, poda, distancia , processa artigos, mínimo e compare nocase. A complexidade em cada parte varia entre O(1) + O(n) +O(n²), sendo que O(1) atribuições, com isso analisamos sua complexidade assintótica e pegamos o de maior grau sendo um algoritmo linear de O(n²).

Resultados

Os resultados são baseados em cálculos de métricas de precisão, revogação e f1par a par. Sendo que todas as métricas são representadas por equações, como:

$$F1 = \frac{2 * precisão * revocação}{precisão + revocação}$$

$$precisão = \frac{qtde\_total\_de\_pares\_de\_registros\_do\_mesmo\_autor\_nos\_subconjuntos\_gerados}{qtde\_total\_de\_pares\_de\_registros\_nos\_subconjuntos\_gerados}$$

$$revocação = \frac{qtde\_total\_de\_pares\_de\_registros\_do\_mesmo\_autor\_nos\_subconjuntos\_gerados}{qtde\_total\_de\_pares\_de\_registros\_dos\_subconjuntos\_que\_deveriam\_ser\_gerados}$$

Conclusão

O trabalho foi bastante complicado de se entender quando se tratava juntar os três métodos de programação, tendo pouquíssimos materiais de apoio na web para complementar o conhecimento sobre o assunto tratado.

Referência

http://www.ime.usp.br/~pf/analise_de_algoritmos/aulas/divide-and-conquer.html

https://www.dcc.ufmg.br/pos/cursos/defesas/1796M.PDF

http://www.each.usp.br/digiampietri/ACH2002/notasdeaula/7-divisaoEConquista.pdf