Relatorio EP 3
Localização de Palavras I

Fellipe Souto Sampaio[1]

MAC 0323 Estrutura de Dados
Prof. Dr. Yoshiharu Kohayakawa

Instituto de Matemática e Estatística - IME USP
Rua do Matão 1010
05311-970 Cidade Universitária, São Paulo - SP

[1]e-mail: fellipe.sampaio@usp.com

1 Introdução

Este relatório serve como descrição para o exercício programa 3 e a forma como sua solução foi implementada. O algorítmo utilizado no EP3 para implementação das árvores rubro-negras esquerdistas foi baseado nas notas de aula do professor Yoshiharu e nas notas online do professor Sedgewick.

2 Client.c

Este é o arquivo principal do programa, o cliente. Através dele são feitas as chamadas para os outros módulos que constroem as ARNEs e buscam as palavras no texto.

2.1 int main

Parâmetros de entrada : -farquivodeentrada.txt.
Saída: 0 se a execução ocorrer sem falha.

O programa recebe como entrada o nome do livro processado pelo CoreNLP e o abre. No main é inicializada as árvores rubro-negras que implementam o funcionamento do programa. Após a leitura de todas palavras o usuário pode entrar com palavras para realizar buscas, as seguintes opções foram implementadas

-e Imprime todas as sentenças que ocorrem a palavra λ.

-ev Imprime todas as sentenças que ocorrem a palavra λ, o número da sentença e a quantidade de tokens.

-a Imprime todas as sentenças que ocorrem o lema λ e todas suas variações.

-av Imprime todas as sentenças que ocorrem o lema λ e todas suas variações, juntamente com o número da sentença e a quantidade de tokens.

-aV Imprime as sentenças notadas de um determinado lema λ.

3 interface.c

Esta interface funciona como ponte entre o cliente, as árvores de busca e o livro processado, por meio dele o cliente pode processar um livro e procurar palavras nas árvores, recebendo como produto final a localização de palavras e lemas ao longo do texto.

4 readBook.c

Interface que processa o livro fornecido pelo usuário. Cada sentença anotada é separada e processada individualmente, como se fosse uma linha individual. Dentro dessa linha separa-se a senteça, os tokens e o número da senteças, que são armazenados na estrutura de dados bookSentence. O restante da senteça anotada (informação gerada pelo coreNLP) é filtrado pela função do C *strtok* que transforma cada item fechado por chaves da sentença em um token individual, fornecendo como saída um par palavra e lema. Esse par é enviado para ambas as árvores para que sejam inseridos. Ao fim da extração do lema, cada sentença notada é salva para futuras impressões.

5 lemmaTree e wordTree

Interfaces de criação, edição e gerenciamento das árvores rubro-negras. O funcionamento é análogo ao explicado em sala de aula, pequenas resalvas são feitas quanto os itens que cada nó carrega e as chaves. Na árvore de palavras(wordTree) temos como chaves as próprias palavras e na árvores de lemas temos estes como chaves. Ambas árvores partilham da estrutura de dados EnglishWord, com uma unica diferença de uma para outra.

5.1 Englishword

Existem três campos dentro desta estrutura:

- word - A palavra ou lema, dependendo da árvore
- sentenceArray - Em wordTree guarda uma copia do lema da palavra word, em lemaTree guarda um array de strings nos quais todas são derivações do lema.
- sentenceLocal - Em wordTree guarda o número da sentença em que word aparece e na primeira posição a quantidade de elementos que existem no vetor. Em lemaTree guarda a quantidade de elementros que existem em sentenceArray

O comportamento de cada estrutura em sua respectiva árvore não diferencia-se muito de uma para outra, as principais diferenças são as descritas acima.

6 itemWord e itemLemma

São as funções que manipulam certos tipos de dados dentro das árvores de busca. Ambas funções são exatamente iguais, decidi não unilas em uma referência única por problemas que estavam ocorrendo quando as funções eram requisitadas.