Faculdade de Engenharia de Universidade do Porto

FEUP

Tema 4 – Classificação de Notícias

João Pedro Milano Silva Cardoso
200900579 - ee09063@fe.up.pt

Rui Filipe Laranjeira da Costa
199600952 – ei12150@fe.up.pt

Joel Alexandre Ezequiel Dinis
120509064 – ei12064@fe.up.pt

Relatório realizado no âmbito do Mestrado Integrado de Informática e Computação
Concepção e Análise de Algoritmos

2 de Junho de 2014

Índice de Conteúdos

Índice de Ilustrações

Introdução

O relatório apresentado foi realizado no contexto do 1º trabalho prático da Unidade Curricular de Concepção e Análise de Algoritmos.

Neste trabalho, pretende-se efectuar a análise de um texto correspondente a uma notícia, classificando-a de acordo com o seu conteúdo. As notícias são classificadas em diferentes categorias: política, desporto, economia, entretenimento.

Na determinação da categoria de uma notícia, é medida a frequência de ocorrência de termos relacionados com cada categoria (não incluindo nestes cálculos palavras irrelevantes: artigos, conjunções,…).

Princípio de Funcionamento

Dados de Entrada

O programa utiliza como dados de entrada ficheiros com extensão txt. Um desses ficheiros é o artigo a ser analizado. Os restantes são os dicionários com termos relacionados a um determinado tema.

Funcionamento do Programa

O programa começa por carregar o artigo para uma string e cada dicionário para um vector de strings. Na string do artigo são retirados caractéres especias como vírgulas, pontos finais, parênteses, aspas e todos os caractéres passam para lowercase.

Em seguida, para cada string de cada dicionário, o programa aplica o algoritmo Knuth–Morris–Pratt para encontrar a palavra ou expressão no artigo.

Este algoritmo devolve um vector de inteiros, a indicar em que posição se encontra a string. O tamanho deste vector indica quantas vezes a string aparece no artigo.

Limites da Aplicação/Restrições

Como o algoritmo KMP encontra padrões de caracteres em strings, é possível que existam palavras que sejam contadas incorrectamente. Por exemplo, se o padrão em questão for "prima" e o algoritmo encontrar a palavra "primavera", a palavra prima será contada uma vez.

Resultado Esperado

No final, é impresso, para cada categoria o número total de palavras ou expressões encontradas, as próprias palavras ou expressões e o número de vezes que cada uma aparece.

A categoria escolhida para o artigo é aquela em que aparecerem mais palavras. Por exemplo, se no fim aparecer uma palavra do tema de Política 10 vezes e 5 palavras do tema de Tecnologia, cada uma 1 vez, a categoria escolhida será Tecnologia.

Principal Algoritmo Aplicado

O principal algoritmo aplicado é o algoritmo Knuth–Morris–Pratt.

```
vector<int> T(K.size() + 1, -1);
vector<int> matches;

if(size(K) == 0)          //string vazia
        matches = matches ∪ {0}
        return matches;

for i = 1 to size(K)   do          //O(n) -> linear, tamanho de K, auxiliar
        int pos = T[i - 1]
        while(pos != -1  && K[pos] != K[i - 1])
                pos = T[pos]
        T[i] = pos + 1

while(sp < S.size()) do                    // O(n) -> linear, do tamanho de S
        while(kp != -1 && (kp == K.size() || K[kp] != S[sp])) do
                kp = T[kp]
        kp++
        sp++
        if(kp == K.size() && S[sp] == ' ' && S[sp-K.size()-1] == ' ') do
                matches = matches ∪ {sp - size(K)}

return matches;
```

Embora o algoritmo KMP procure padrões em strings, este foi modificado para procurar palavras inteiras.

Quando o padrão é encontrado, é também verificado se o caracter seguinte é um espaço, bem como o caractér anterior ao ínicio do padrão, de modo a encontar uma palavra inteira, em vez de uma substring.

Algoritmo Alternativo

O algoritmo KMP encontra padrões em strings. Se uma string que se procura for uma substring de outra palavra ou expressão, a contagem final irá ser errada.

Antes de se modificar o algoritmo KMP desenvolveu-se este algoritmo extra, o qual tem complexidade O(n*m), sendo "n" o número de palavras no artigo e "m" o número de palavras na expressão que se quer encontrar. O espaço auxiliar usado para inicialização é O(n+m).

```
vector< string > strs;
vector< string > strSearch;

istringstream stm(article) ;
string word ;
while( stm >> word ) //-> O(n), auxiliar
{
        strs.push_back(word);
}
string buf;
stringstream ss(myFrase);
int ncount = 0;

while (ss >> buf) // -> O(m), auxiliar
        strSearch.push_back(buf);

for(unsigned int i=0; i<strs.size(); i++) //-> O(n)
{
        unsigned int total=0;
        for(unsigned j=0;  j<strSearch.size(); j++) // -> O(m)
        {
                if((i+j)<strs.size())
                {
                        if(strs[i+j] == strSearch[j])
                                total++;
                }
        }
        if(total==strSearch.size())
                ncount++;
}
return ncount;
```

Usos da Aplicação

A aplicação é capaz de carregar um ficheiro de texto, e usando ficheiros adicionais, os dicionários, procura palavras ou expressões no ficheiro inicial. O programa classifica o texto com base nos dicionários de vários temas.

Diagrama UML

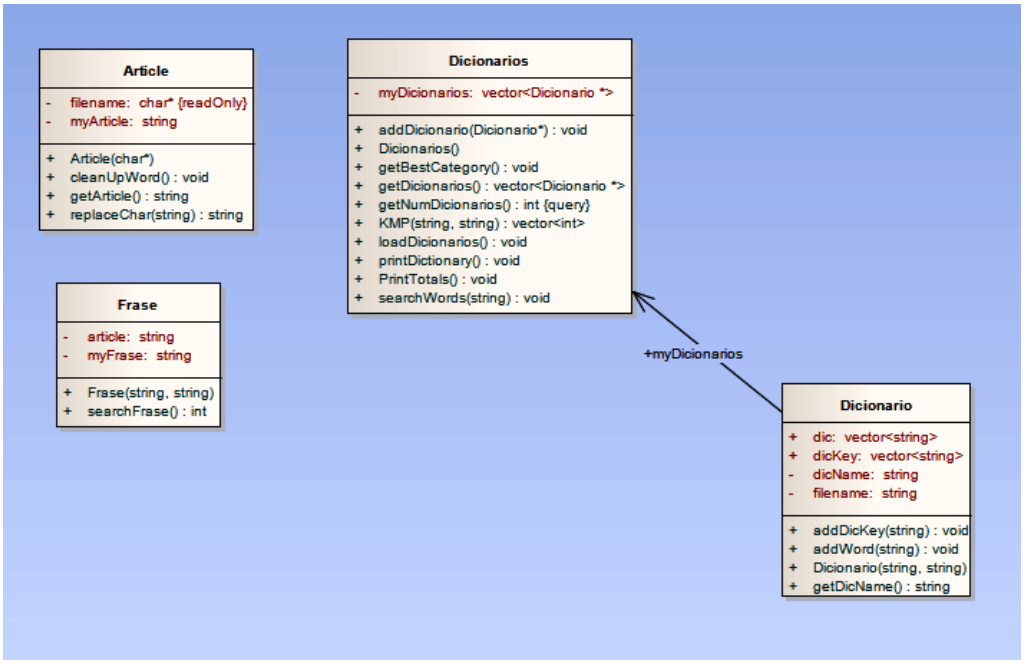

Ilustração 1 - Diagrama UML

Contribuição dos Elementos do Grupo

Elaboração do Programa sem Classes/Optimização – João Cardoso
Elaboração de Classes/Optimização – Rui Costa
Elaboração do Relatório – Joel Dinis

Conclusões

É possível concluir que a complexidade do algoritmo é O(|n|) e que a complexidade do espaço auxiliar, é também O(|n|).

O algoritmo percorre sempre todo o artigo.

Algumas melhorias a implementar poderiam ser mudar a estrutura do programa de modo a que cada palavra seria o nó de uma árvore binária. Deste modo, ao introduzir palavras na árvore, estas poderiam ser filtradas de modo a retirar artigos definidos e indefinidos, por exemplo.

Em cada nó poderia estar também o número de vezes que cada palavra aparece. Para procura numa árvore binária no pior caso a complexidade seria O(|h+1|) em que h é altura da árvore.

Outra melhoria poderia ser atribuir um peso a cada string de cada dicionário e na escolha de cada categoria encontrar um equilíbrio entre o peso total e o número de palavras.