Universidade de Aveiro
Engenharia de Computadores e Telemática
Informação e Codificação
Gabriel Saudade [89304], António Domingues [89007] e
Henrique Silva [88857]

universidade de aveiro

Assignment 3

01 Fevereiro 2022

Introdução

O presente relatório serve o propósito de explicar detalhadamente os métodos usados e raciocínios seguidos para a realização dos exercícios propostos no projeto 2 da cadeira de IC. Para cada exercício é feita uma descrição do código implementado e são apresentados os resultados obtidos.

O link para o repositório do github referente ao projeto é o seguinte: **https://github.com/SAUDADE1994/project3_ic**

Parte A

Com o intuito de recolher informação estatística de um ficheiro de texto, foi criado um programa (***fcm***), para recolher toda a informação necessária. Enquanto grupo, decidiu-se apenas ter em conta todos os caracteres que fazem parte do alfabeto, excluindo todos os espaços, vírgulas, acentos, e outros símbolos que não letras. Ao efetuar estas condições, numa fase seguinte, a deteção de similaridades entre textos, fica um pouco limitada a textos em que as vírgulas e os pontos, ..., são irrelevantes. Se o objetivo fosse identificar uma linguagem de programação (por exemplo), todos os símbolos delimitadores, contêm informação útil para determinar qual a linguagem do código em causa e assim seria proveitoso também ter em conta esses diversos símbolos para a construção do nosso modelo.

Como visto nas aulas, o valor de ***K***, define o tamanho de cada contexto que posteriormente vai ser armazenado num ***map***. Para se armazenar toda a informação e se construir o modelo, foi criado um **map<string, map<char, uint32_t>>**, que contém todos os contextos, e a contagem de ocorrências de cada char para esse determinado contexto. De seguida, criou-se um outro **map**, para armazenar o total de cada contexto, para numa fase

seguinte, determinar a entropia do ficheiro em análise. De seguida são apresentados alguns resultados da entropia, para vários valores de ***K***, para vários ficheiros de texto.

Parte B

Nesta parte do trabalho, é necessário efetuar a comparação entre dois textos, para se determinar quantos bits são necessários, para comprimir um texto ***t***, usando um texto de referência ***ri***. Este exercício foi implementado no mesmo ficheiro, que os exercícios seguintes. Muito rapidamente, é executada a função main, onde é passado um texto ***t***. O nosso código vai determinar o modelo para um determinado texto de referência(***ri***) e esse texto ***t***, vai ser comparado com esse texto ri, obtendo-se no final um número total de bits necessário para comprimir esse texto ***t***, com o modelo do texto ***ri.***

Este processo é efetuado para diversos textos de referência, onde vão sendo calculados o número de bits necessários para codificar o texto passado como argumento pelo terminal. O texto ***t***, é percorrido da mesma forma que o texto referência, só que em vez de se fazer a contagem de ocorrências do caracter ***k+1***, determina-se qual a probabilidade de aparecer o caracter ***k+1***, recorrendo ao modelo do texto de referência ***ri.*** Após determinar essa probabilidade, faz-se o *-log2* dessa probabilidade para se converter uma probabilidade em número de bits. Nota: quanto maior for a probabilidade, menos bits são necessários, quanto menor for a probabilidade, mais bits são necessários. Ao comparar o texto ***t***, com vários modelos de referência, esse vai ser mais semelhante ao texto de referência que necessitar de menos bits para o comprimir.

Conclusões

Relativamente aos resultados obtidos, para um texto em português algum dos resultados obtidos no que diz respeito às referências textuais para **k=3** e **alpha=0.9** foram os seguintes:

	Total Bits	Bits/char	ExecTime (seconds)
Czech	10330288.895	3.414	4.565
Spanish	8559664.214	2.829	4.634
Lithuanian	9753949.614	3.223	4.372

...	...	...	...
Slovak	10345148.939	3.419	4.610
French	9263504.423	3.061	4.472
Portuguese	7531525.275	2.489	4.804

A linguagem foi corretamente encontrada, pois a referência cujo número necessário de bits para comprimir é menor, é a linguagem portuguesa.

Para o mesmo exemplo, mudando o valor de **alpha** para 0.1 os resultados obtidos foram os seguintes:

	Total Bits	Bits/char	ExecTime (seconds)
Czech	10023508.300	3.312	4.529
Spanish	8358770.884	2.762	4.705
Lithuanian	9415215.564	3.111	4.473
...	...	...	...
Slovak	10059339.172	3.324	4.608
French	9027054.897	2.983	4.394
Portuguese	7291179.939	2.409	4.936

Como podemos ver, o número total de bits por caracter diminui. Ao aumentar o valor de **K**, o tempo de execução também vai ser maior, aumentando aproximadamente com um fator de 1,33.

Por fim, para certas linguagens cujo o alfabeto contém muitos caracteres cuja representação é mais de 1 byte (caracteres com acento, por exemplo), a dedução não foi correta, pelo facto de se estar a descartar todos esses assentos para a definição do nosso modelo.