**TAREFA1**

O presente vídeo tem como objetivo descrever a resolução do Projeto 3  
desenvolvido no âmbito da unidade curricular de Informação e Codificação.

Na primeira tarefa, o objetivo foi desenvolver um programa denominado fcm onde  
este fornece ferramentas que permitem recolher informações estatísticas sobre  
textos com base num modelo de contextos finito.

Dados um ficheiro de entrada, a ordem do modelo, k, e um smoothing parameter, α, este programa consegue calcular a entropia do texto. De seguida, iremos executar um ficheiro de teste para esta tarefa para observarmos os valores obtidos nesta implementação.

Executar ficheiro e Abrir imagem

../texto-models-bin/

Sabemos que, à medida que se aumenta a ordem do modelo o tamanho da  
palavra considerada será cada vez maior e daí a variedade de caracteres que  
podem aparecer de seguida será menor. Isto é, se pensarmos numa palavra  
de ordem 10 como, por exemplo, "referência" conseguimos facilmente concluir  
que o número de caracteres que poderão aparecer a seguir a esta palavra será  
reduzido. Na sua maioria, poderá aparecer um "s" ou algum caracter especial  
como, por exemplo, um ponto final.  
De igual forma, considerando o caso oposto, palavras de ordem pequena, como  
1, existe uma enorme variedade de caracteres seguintes possíveis.  
Ou seja, concluímos que o aumento da ordem de um modelo reduz a sua en-  
tropia como conseguimos observar no gráfico.

**TAREFA 2**

Passando à tarefa 2, este programa, vai receber 2 ficheiros: um com um texto representativo,  
ri, em qualquer linguagem e um outro ficheiro, ti, que vai ser analisado para  
ser calculado o número de bits precisos para comprimir este ficheiro usando o  
modelo de ri.

Dependendo do que o utilizador pretenda, pode ser criado um modelo a partir  
de um texto, como nos é pedido

Adicionalmente, acrescentámos também uma opção em que o ficheiro inse-  
rido pode ser já um modelo existente calculado anteriormente, e dessa forma o  
cliente insere uma flag, ’-l’, se pretender usar este modo.

Para verificarmos o funcionamento do nosso programa vamos testar-lo com  
diferentes modelos e para cada um foram usar vários ficheiros de texto em  
linguagens diferentes, sendo calculada a estimativa do número de bits necessários  
para os comprimir tal como o número de bits estimados para representar cada  
caractér. Usando k com o valor 5 e alfa com 0,1 temos:

tabela

Conseguímos observar que, para cada modelo o texto que obtém a menor  
distância calculada corresponde à linguagem representada pelo modelo, como  
pretendíamos demonstrar

**TAREFA 3**

de seguida mostramos o correto funcionamento do programa, onde este advinha a linguagem para qual um certo texto, t, foi escrito.

EXECUTAR ../texto-models-bin/findlang BOOKS/akumaJPN.txt

Para este teste, a linguagem fornecida, foi de facto, japonês.