

Universidade de Aveiro

Departamento de Eletrónica, Telecomunicações e Informática

Informação e Codificação (2021/22)

Projeto #03

Finite Context Models

Equipa:

João Moraes - 93288

Pedro Coutinho - 93278

Repositório:

github.com/antoniojoao10/Project3IC



universidade
de aveiro

Parte A

Ex1

Comandos:

```
$ g++ -o <ficheiro-output> ./ExA1.cpp  
$ ./<ficheiro-output> <ficheiro-de-entrada> <valor-k> <valor-a>
```

Descrição da implementação:

O programa recebe um ficheiro de entrada, a ordem do modelo 'k' e o parâmetro de smoothing 'a'. Com esses valores, ele utiliza as seguintes fórmulas para recolher a entropia do texto do ficheiro de entrada, o número de bits, o número de símbolos e o número estimado de bits por símbolo.

$$symbol_prob = \frac{symbol.num_occurences + alpha}{ctx.num_occurences + (alpha * alphabet.length)}$$

$$context_prob = \frac{ctx.num_occurrences}{total_ctx_count}$$

$$context_entropy = \sum symbol_prob_i * -\log_2(symbol_prob_i)$$

Neste caso, *symbol.num_occurrences* é o número de vezes que um determinado símbolo apareceu a seguir ao contexto ctx e

ctx.num_occurrences é o número de vezes que o contexto ctx aparece no ficheiro utilizado.

ctx.num_occurrences tem o mesmo significado do primeiro tópico. *total_ctx_count* é o número de contextos foram construídos a partir do ficheiro utilizado.

Demonstração:

```
Entropy --> 2.34072  
Numbers of bits --> 5.43642e+06  
Number of symbols --> 87  
Estimated number of bits per symbol: 62487.5
```

Parte B

Ex1

Comandos:

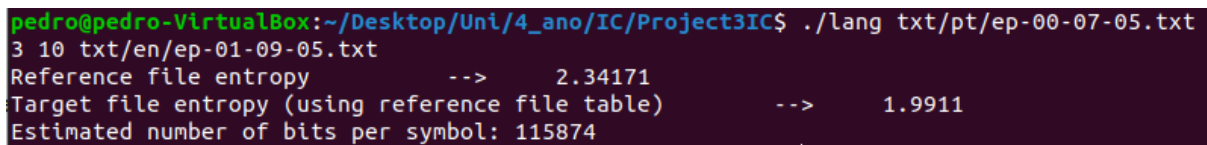
```
$ g++ -o <ficheiro-output> ./lang.cpp
$ ./<ficheiro-output> <ficheiro-referencia> <valor-k> <valor-a> <ficheiro-analise>
```

Descrição da implementação:

O programa vai utilizar o ficheiro de referência para encher as tabelas que vão ser usadas como base para a comparação com o ficheiro de análise. Quanto maior for a entropia do último ficheiro em comparação com o primeiro, maior é a probabilidade de ser a mesma linguagem.

Vai ser impresso a entropia do ficheiro de referência, a entropia do ficheiro de análise usando a tabela do ficheiro de referência e o número estimado de bits por símbolo.

Demonstração:



```
pedro@pedro-VirtualBox:~/Desktop/Unl/4_ano/IC/Project3IC$ ./lang txt/pt/ep-00-07-05.txt
3 10 txt/en/ep-01-09-05.txt
Reference file entropy      -->      2.34171
Target file entropy (using reference file table)      -->      1.9911
Estimated number of bits per symbol: 115874
```

Ex2, Ex3 & Ex4

Comandos:

```
$ g++ -o <ficheiro-output> ./findLang.cpp
$ ./<ficheiro-output>
```

Descrição da implementação:

Quando o utilizador corre o programa, este vai ser apresentado com quatro opções. A opção de verificar que linguagens; a de aprender uma linguagem nova, a de adivinhar a linguagem de um ficheiro e a de sair.

Se escolher a primeira opção vai ser apresentada uma lista de todas as linguagens já conhecidas, que se encontram na pasta 'savedLang'.

Na segunda opção é inicialmente perguntado o nome da linguagem que se pretende adicionar, seguido do nome do ficheiro e por último o valor do 'k' e do 'a'. A linguagem é guardada, sendo impresso o valor da entropia e o número estimado de bits por símbolo.

Na terceira opção é pedido o nome do ficheiro que se pretende adivinhar a língua, o valor de 'k' e de 'a'. A partir dessa informação o ficheiro vai comparar com todas as linguagens guardadas, imprimindo em cada teste a entropia utilizando a tabela de lang e o número estimado de bits por símbolo. No final o programa diz a linguagem que pensa que o ficheiro está escrito, com base nas linguagens que o programa aprendeu anteriormente.

O programa está implementado de maneira que consiga aprender com qualquer valor de 'k'. Além disso, também consegue descobrir a linguagem com qualquer valor de 'k' desde que este tenha sido previamente ensinado.

Demonstração:

```
Programs:
Press 1 : Check known languages
Press 2 : Learn new language
Press 3 : Guess file language
Press 0 : LEAVE
Enter ---->
```

```
Enter ---->1
```

```
pt_k3
pt_k2
bg_k3
en_k3
es_k3
it_k3
el_k3
fr_k3
hu_k3
```

```
Enter ---->2
```

```
Language(example:pt):lv
File:txt/lv/ep-08-10-21-007.txt
K value:3
a:10
Entropy      -->      2.79294
Estimated number of bits per symbol: 45971.9
```

Enter ---->3

File to guess language:txt/it/ep-00-07-05.txt

K value:3

a:10

Testing language: pt

Entropy using table from lang --> 1.78019

Estimated number of bits per symbol: 41640.6

Testing language: bg

Entropy using table from lang --> 0.0426678

Estimated number of bits per symbol: 30102.8

Testing language: en

Entropy using table from lang --> 1.84811

Estimated number of bits per symbol: 44357.1

Testing language: es

Entropy using table from lang --> 1.82095

Estimated number of bits per symbol: 42129.2

Testing language: it

Entropy using table from lang --> 2.0251

Estimated number of bits per symbol: 42899.4

Testing language: el

Entropy using table from lang --> 0.042532

Estimated number of bits per symbol: 30145.2

Testing language: fr

Entropy using table from lang --> 1.62521

Estimated number of bits per symbol: 41234.1

Testing language: hu

Entropy using table from lang --> 0.8956

Estimated number of bits per symbol: 40274.9

Testing language: lv

Entropy using table from lang --> 1.37103

Estimated number of bits per symbol: 40593.2

Language is: it