

Lab work n◦1

Teoria Algorítmica da Informação

Trabalho realizado por:

- Isaac dos Anjos nmec : 78191

- Lucas Barros nmec : 83895

- Pedro Cavadas nmec : 85090

Introdução

O principal objetivo deste trabalho é criar um gerador de texto automático utilizando um modelo estatístico. This modelo is generated through processing and analyzing input text. Throughout this report, there will also be studys of the behavior of the text generator sliding the parameter values.

(acho devemos introduzir k e alfa debaixo do Modelo Matematico)

Para além disto, pretendemos estudar o comportamento de 2 parâmetros k e alfa e observar o comportamento do gerador de texto ao variarmos estes parâmetros.

Modelo Matemático

Para gerar o texto, pensámos num sistema que tivesse algo semelhante a um histórico que nos permitisse gerar um texto com melhor precisão. Por exemplo, se no texto original ocorrer a palavra “pois” e caso estejamos a considerar um histórico de 3 letras (ou seja, k = 3),where k is length of the contexto in question, então se o texto gerado for “poi” então é de se esperar que o programa gere um ‘s’ como próxima letra.

Para um alfabeto de 0’s e 1’s, temos na seguinte tabela um exemplo do que pretendemos

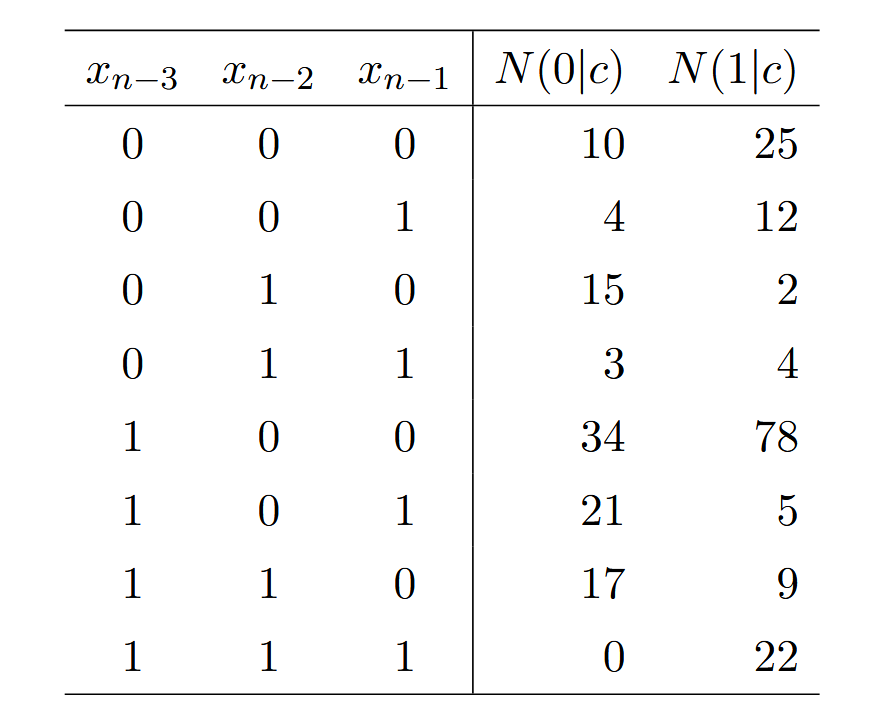
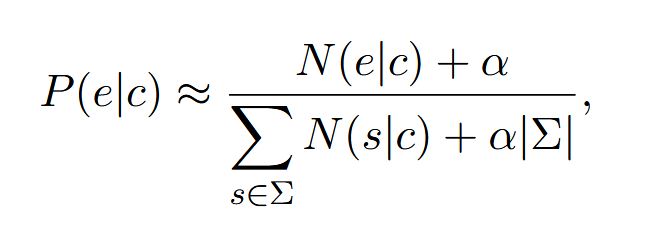


Figura - Tabela de estatísticas

em que na primeira linha N(0|c) é a contagem de vezes em que 0 apareceu seguido do contexto “000”. Assim, a probabilidade de se escrever um certo carater e seguido de uma sequência c é dado por

Although the model presented above seems straight forward, there is a problem when the program generates the ‘111’. The probability that the next character is ‘1’ is 1 or 100% meaning that the generator will enter an infinite loop which is not desirable. To solve this, alpha is added to the formula.



Looking back to the previous problem, if alpha = 1, the probability of generating ‘0’ knowing that our context is ‘111’ is 1/(22+2) thus eliminating any existing loop holes.

(should I talk about the side affects?)

Descrição da Solução

A nossa solução está partida em 2 componentes principais, os ficheiros “fcm.cpp” e “generator.cpp” em que ambos utilizam a classe “MarkovModel” descrita nos ficheiros “markov\_model.hpp” e “markov\_model.cpp”. Primeiro é executado o programa “fcm.cpp” que necessita de 2 argumentos obrigatórios (alfa e k) e 2 opcionais (nome do ficheiro de input, e nome do ficheiro de output).

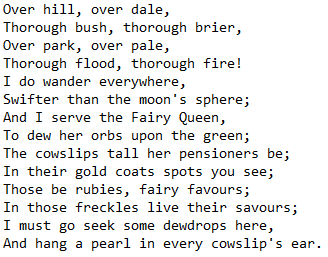
fcm.cpp

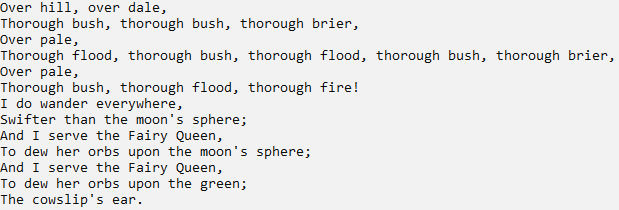
Este programa analisa o ficheiro de input e retira informações estatísticas sobre o texto. Conta o número de vezes em que cada letra aparece individualmente e com isso, calcula a probabilidade de aparecer qualquer letra isolada. Também calcula a probabilidade de aparecer uma letra seguida de um dado contexto de tamanho k. Ambos estes resultados são guardados em “unordered maps” na classe MarkovModel. Depois de calculadas estas estatísticas, o programa escreve no ficheiro de output. A escrita é feita em binário pois é mais simples e menos propício a erros.

Generator.cpp

Autista da piça, this is on you, i have no fucking clue of what u did here!!

Análise de Resultados





Conclusão

Referências