Universidade de Aveiro
Departamento de Electrónica, Telecomunicações e Informática
Mestrado em Engenharia Informática
Ano Letivo 2020/2021

Teoria Algorítmica da Informação

Trabalho Prático nº 1

Aveiro, 8 de novembro de 2020

António Ramos, 101193
Luís Laranjeira, 81526
Rafael Sá, 104552

Índice

1. Desenvolvimento do Programa

O objetivo do programa é criar um modelo com o objetivo de recolher informação estatística sobre textos, usando *finite-context models*.

Um *finite-context (Markov) model*, de ordem k, produz a distribuição de probabilidade do próximo símbolo numa sequência de símbolos, tendo em conta o histórico recente até à profundidade k.

1.1. Estrutura do Programa

O modelo contém uma *hash table* utilizada para guardar a contagem, que representa o número de vezes que cada símbolo aparece em cada contexto, o alfabeto usado no modelo, e a soma total do número de ocorrências.

A *hash table* utiliza uma estrutura chave-valor, na qual a chave é cada contexto e o valor é uma nova *hash table,* na qual é guardada a contagem de cada caracter no respectivo contexto e a soma do número de entradas. Com esta abordagem, o objetivo é aliar as vantagens de usar uma tabela às vantagens de usar uma *hash table*, sendo que a primeira *hash table* representa as linhas da tabela e a segunda *hash table* representa as colunas.

Cada linha representa um modelo de probabilidade que é usado para representar um determinado símbolo de acordo com os últimos símbolos processados. Os contadores são atualizados cada vez que um símbolo é processado.

As vantagens desta abordagem são o rápido acesso e o menor uso de memória.

Com esta estrutura podemos dar resposta às necessidades do problema: guardar a estimativa de probabilidades, fornecer a entropia do texto, e gerar novo texto.

Cada um dos programas recebe um conjunto de parâmetros no qual a sua execução se deve basear. Entre eles estão a ordem k, o parâmetro de *smoothing* alpha e o ficheiro de texto para treinar o modelo. Caso os parâmetros k e alpha tenham valores inválidos, os valores por defeito são k = 3 e alpha = 0.

1.2. Criação do Alfabeto

O alfabeto é produzido automaticamente com base nos caracteres existentes no texto fornecido. Isto é feito numa primeira leitura ao ficheiro que contém o texto, onde todos os caracteres não duplicados são adicionados à estrutura de dados responsável por armazenar o alfabeto.
Este alfabeto é utilizado tanto para a criação do modelo como para a geração de texto, ao invés de se utilizar todos os símbolos da tabela ASCII.

1.3. Criação do Modelo

Tal como referido na estrutura do programa, o modelo consiste numa *hash table* que guarda informação estatística sobre textos. Neste sentido é necessário ler um documento de texto para podermos recolher esta informação. Este processo tem como objetivo treinar o modelo. O processo para treinar o modelo consiste em recolher as ocorrências de cada símbolo num determinado contexto.

Para a determinação do contexto e do caractere atual utilizamos a ordem do modelo k, os dados do ficheiro e a posição de cada caractere. Esta operação vai ser feita percorrendo todos os caracteres do ficheiro, guardando para cada sequência de caracteres de tamanho k o seu sucessor, tal como o número de vezes em que o mesmo aparece no mesmo contexto.

No final deste processo podemos usar o modelo para o cálculo das probabilidades, obter a entropia do texto e, caso seja esse o objectivo, gerar um conjunto de texto.

1.4. Cálculo das Probabilidades e Entropia

Para o cálculo das probabilidades é utilizado o parâmetro de *smoothing* alpha, tanto no cálculo da probabilidade de um caracter sabendo o seu contexto, como no cálculo da probabilidade de um contexto. Optou-se por utilizar o mesmo valor para ambos os cálculos, em vez de se utilizar dois parâmetros alfa distintos.

Caso o parâmetro alpha seja zero, então o cálculo da entropia apenas considera os contextos, e respectivos caracteres, que surgiram durante a criação do modelo. Neste caso, existe o cuidado de impossibilitar o cálculo da entropia de um contexto que não exista no modelo, desta forma evita-se o cálculo de um logaritmo de zero.

Por outro lado, se o parâmetro alpha possuir um valor maior que zero, então todos os contextos e símbolos são considerados para o cálculo da entropia. Optou-se por não gerar todos os contextos que não surgiram no modelo e calcular a entropia para cada um deles, uma vez que esse valor seria igual em todos. Por isso, é calculada a entropia, e respetiva probabilidade, de um contexto genérico que não surge no modelo, e multiplicada pelo número de contextos que não surgiram no modelo.

1.5. Geração de Texto

Com o modelo carregado é possível aceder às estimativas de probabilidades. Com este modelo estatístico podemos gerar o texto a partir de um ponto de partida. A primeira abordagem foi gerar o texto tendo em conta apenas o caractere com maior probabilidade, algo que se mostrou errado dado que torna impossível que o texto gerado seja minimamente aleatório, tornando-se repetitivo ao fim de algum tempo.

Depois de discutido entre os elementos do grupo e com o professor chegou-se à conclusão que a melhor abordagem seria gerar um valor entre 0 e 1, e perceber em qual dos caracteres este valor atinge, o que permite que valores com grande probabilidade tenham maior probabilidade de ser escolhidos, mas também que outros caracteres com probabilidades menores possam ser escolhidos.

2. Análise de Resultados

Utilizando o programa desenvolvido para treinar o modelo e gerar texto, procedeu-se à análise de alguns resultados considerados relevantes.

Foi analisado o impacto da variação dos parâmetros tanto no cálculo da entropia e tempo de criação do modelo, como também no texto que é gerado. Para além disso, foi feita uma análise e comparação dos valores da entropia entre diferentes tipos de textos.

2.1. Variação dos Parâmetros *k* e *alpha*

Esta análise tem como objetivo analisar o impacto da variação dos parâmetros k (ordem do modelo) e alpha (parâmetro de *smoothing*) no cálculo da entropia, e o impacto da variação do parâmetro k no tempo que o modelo demora a ser criado.

Esta análise foi feita ao texto de exemplo fornecido “example.txt”.

A tabela 1.1. permite analisar como é que a entropia varia consoante a ordem do modelo e o valor do parâmetro de *smoothing*.

	Entropia
k = 2 ; alpha = 0	1.749233633789491
k = 4 ; alpha = 0	1.1352375651154194
k = 7 ; alpha = 0	0.7515610143141768
k = 2 ; alpha = 0.2	1.7636595567945261
k = 4 ; alpha = 0.2	2.1679496349274645
k = 7 ; alpha = 0.2	4.0073063449266675

Tabela 1.1. Análise da entropia com base na variação dos parâmetros k e alpha

Analisando o impacto do parâmetro k consoante a existência de um valor alpha no cálculo da entropia, é possível destacar dois comportamentos distintos.

Quando o parâmetro alpha é igual a zero (não é considerado o parâmetro), verifica-se que quanto maior a ordem do modelo, menor o valor da entropia. Já quando o valor do alpha é diferente de zero assiste-se ao comportamento contrário, ou seja, em vez do valor da entropia diminuir com o aumento do k, o valor aumenta.

A tabela 1.2. permite compreender como é que apenas o parâmetro alpha influencia o valor da entropia.

	Entropia	Taxa de Crescimento (%)
k = 4 ; alpha = 0	1.1352375651154194	-
k = 4 ; alpha = 0.3	2.4477908682491125	131,25533031336931
k = 4 ; alpha = 0.6	2.9509560438208275	20,55588907117811
k = 4 ; alpha = 0.9	3.2204722036885687	9,133181106919439

Tabela 1.2. Análise da entropia com base na variação do parâmetro alpha

É possível constatar que quanto maior for o valor do parâmetro de *smoothing*, maior será o valor da entropia. Apesar disso, verifica-se que a taxa de crescimento do valor da entropia vai diminuindo consoante o aumento do valor do alpha.

A tabela 1.3. permite analisar o impacto da ordem do modelo no tempo o programa demora a criar modelo.

	Tempo de Criação do Modelo (segundos)
k = 2	1.6153804
k = 4	2.5901657
k = 7	4.5040505
k = 10	5.1755904

Tabela 1.3. Análise do impacto da ordem no tempo de criação do modelo

É possível verificar que quanto maior o parâmetro k, maior será o tempo de criação do modelo.

2.2. Variação da Entropia em Diferentes Textos

Para testar a variação do valor da entropia, consoante o tipo de texto utilizado para criar o modelo, definiu-se um conjunto de textos em diferentes linguagens. Os textos utilizados são as seguintes obras literárias:

- Inglês: King James Bible (*example.txt*)
- Inglês: Mary Shelley, Frankenstein (*Frankenstein.txt*)
- Português: Eça de Queirós, A Cidade e as Serras (*A Cidade e as Serras.txt*)
- Português: Almeida Garrett, Viagens na Minha Terra (*Viagens na Minha Terra.txt*)
- Francês: Jean Aicard, Notre-Dame-d'Amour (*Notre-Dame.txt*)
- Chinês: Xuahua Biao, Hou Xiyouji (Book 2) (*Hou Xiyouji.txt*)

Estes testes foram efetuados para um modelo de ordem k = 3 e para um parâmetro de *smoothing* alpha = 0.01.

A tabela 1.4. permite estabelecer comparações entre a entropia de diferentes tipos de texto.

	Entropia
example	1.3662035911295678
Frankenstein	1.5913271788872076
A Cidade e as Serras	1.8231369148512446
Viagens na Minha Terra	1.7898624151753872
Notre-Dame	1.7091580960176544
Hou Xiyouji	8.004414771681583

Tabela 1.4. Variação da entropia nos diferentes tipos de texto

Neste contexto literário, podemos verificar que a entropia das obras europeias varia entre aproximadamente 1.36 bits e 1.82 bits. Constata-se que a língua inglesa é a que apresenta por norma uma entropia mais baixa e, por outro lado, a língua portuguesa apresenta uma entropia mais elevada. Analisando o contexto das obras, por exemplo na língua inglesa, é possível verificar que a obra de cariz religioso apresenta uma entropia menor do que a obra de ficção.

Porém, a conclusão mais interessante surge quando se estabelece a comparação entre a entropia das linguagens ocidentais com uma linguagem oriental, neste caso o chinês. Deve ser realçada a grande discrepância de valor, sendo que se passa de uma entropia abaixo dos 2 bits nas obras europeias, para uma entropia na ordem dos 8 bits na obra chinesa. Isto revela que cada símbolo chinês contém uma quantidade de informação muito maior do que um símbolo ocidental.

2.3. Variação dos Parâmetros na Geração de Texto

Para testar a geração de texto foram utilizados os mesmos parâmetros utilizados para testar o cálculo da entropia. Desta forma, foi possível compreender a relação entre a entropia e a qualidade do texto gerado.

Para este teste foi utilizado como ponto de partida o texto “They think”, um tamanho de 500 caracteres e o texto de exemplo fornecido para treinar o modelo. De notar que o texto gerado varia consoante a execução, o que poderá originar algumas diferenças nas conclusões.

A tabela 1.5. representa como é que o texto gerado é influenciado pela variação da ordem do modelo e pelo parâmetro de *smoothing*.

	Entropia	**Excerto do Texto Gerado**
k = 2 ; alpha = 0	1.749233633789491	“They think? amse and”
k = 4 ; alpha = 0	1.1352375651154194	“They think wall their”
k = 7 ; alpha = 0	0.7515610143141768	“They think that hand”
k = 2 ; alpha = 0.2	1.7636595567945261	“They think now this”
k = 4 ; alpha = 0.2	2.1679496349274645	“,d:erqfq?l(2hb;zq66”
k = 7 ; alpha = 0.2	4.0073063449266675	“@i99%e3).:r .-3nfe9(”

Tabela 1.5. Análise do texto com base na variação dos parâmetros k e alpha

Estas conclusões foram tiradas ao longo do desenvolvimento do trabalho e consolidadas com esta análise final.

É possível verificar que se não for considerada a existência do parâmetro de *smoothing*, quanto maior for a ordem do modelo, melhor será a qualidade do texto gerado, uma vez que o modelo é mais preciso. Como o modelo é mais preciso e não permite inovação a entropia será menor, como já foi analisado anteriormente.

Quando se considera o parâmetro de *smoothing*, pode-se verificar que deve existir um equilíbrio entre o valor do alpha e de k para que o texto gerado tenha qualidade. Para valores de k mais baixos, é possível colocar o valor do alpha a valores mais elevados e ainda conseguir obter texto com alguma qualidade. Isto deve-se ao facto de não existirem tantos contextos possíveis e o valor de alpha não influenciar o cálculo das probabilidades a uma larga escala.

Quanto maior for o valor do k, menor terá de ser o valor do alpha para que o texto gerado tenha qualidade. Isto acontece porque quando a ordem do modelo é muito elevada passa a existir um número muito maior de contextos possíveis, e se o valor do alpha for muito elevado, atribui-se uma probabilidade somada demasiado grande a contextos que não deveriam surgir num texto com sentido.

Estabelecendo uma relação com o valor da entropia, pode-se assumir que quanto maior for o valor da entropia, maior será a inovação do modelo e consequentemente menor será a qualidade do texto produzido.

3. Conclusões

Consideramos essencial uma programação cuidadosa com base em *hash tables*, para lidar com as necessidades de memória impostas pelos modelos de Markov com maior profundidade e dispersão inerente aos contextos associados.

Para calcular o valor da entropia do modelo é adotado o logaritmo negativo. Através desta informação é possível gerar o texto, sendo que este vai variar consoante as estimativas de probabilidade e profundidade do contexto.

Para concluir, podemos afirmar que os parâmetros com o qual o modelo é treinado tem um impacto muito grande, do qual é possível verificar que a qualidade e inovação do texto estão diretamente associados à ordem do modelo e ao alpha. Quanto maior a ordem do modelo e menor o alpha, maior é a qualidade mas menor a inovação.