Ficha 9

Scripting no Processamento de Linguagem Natural

23 de Maio de 2015

Distância de edição de duas strings

Vamos usar o seguinte exemplo de uma palavra em Português:

```
pelourinho

poelolurknho
```

Percebemos que provavelmente estamos na presença da mesma palavra mas que, da segunda vez, esta foi escrita com um bocado de pressa. Um alinhamento entre as duas palavras poderia ser:

```
p-elo-urinho
| ||| ||||||
poelolurknho
```

Mas se as duas palavras tivessem erros:

```
ployrinho

poelolukno
```

Então poderíamos ter por exemplo o seguinte alinhamento:

```
p--lo-yrinho
|  || | || |
poelolu-kn-o
```

Score de um alinhamento

O *score* de um alinhamento é um valor numérico que avalia a *qualidade* de um alinhamento. Quanto maior o valor melhor é a qualidade. Para calcular o score temos em conta:

Inserções Tratar como um espaçamento na outra string

Remoções Tratar como um espaçamento nesta string

Substituições Calcular o custo da substituição de um caractere numa das strings por outro caractere na outra string

Espaçamento Pode ser um custo fixo ou ter um custo para o primeiro espaçamento e outro custo menor para os restantes no caso de espaçamentos seguidos

Custo de substituições

Continuando com o nosso exemplo de Português o custo de substituição de um 'u' por um 'y' deverá ser menor do que de um 'u' por um 'l' porque o 'u' e o 'y' são letras vizinhas no teclado.

Assim, é frequente utilizar uma matriz de substituições que penalize menos o alinhamento de um 'u' e um 'y' do que de um 'u' e um 'l'.

Cálculo do score de um alinhamento

Imagine que queremos calcular o score do alinhamento entre `p-elo-urinho` e `poejolurknho` assumindo que:

- A mesma letra vale +4
- Letras vizinhas no teclado valem +2
- Letras não vizinhas valem -1
- Espaços valem -4

```
p - e l o - u r i n h o
|   | | |   | | | | | |
p o e j o l u r k n h o
+4-4+4-1-4+4+4+4+2+4+4+4 = 25
```

Encontrar o melhor alinhamento global entre duas strings

Consideremos as duas strings ABACO e BARCO. Vamos tentar descobrir o melhor alinhamento entre elas usando a técnica comum da decomposição de um problema noutro mais simples.

O score do melhor alinhamento entre estas strings pode ser obtido através do alinhamento entre as suas partes iniciais de 3 formas possiveis:

1. Melhor score de alinhar ABAC com BARCO e inserir um espaçamento na 2ª string
2. Melhor score de alinhar ABACO com BARC e inserir um espaçamento na 1ª string
3. Melhor score de alinhar ABAC com BARC e substituir o O na 1ª string pelo O na 2ª string

```
score('ABACO', 'BARCO') = max(
        score('ABAC', 'BARCO') + custo_espaço,
        score('ABACO', 'BARC') + custo_espaço,
        score('ABAC', 'BARC') + custo_subs('O', 'O')
)
```

Como esta função é recursiva e **muito ineficiente** (diga porquê) vamos usar programação dinâmica. O que se faz é guardar cada um dos scores numa matriz tal como se mostra na tabela a seguir.

	-	A	B	A	C	O
-	0	-4	-8	-12	-16	-20
B	-4	-1	0	-4	-8	-12
A	-8	0	-2	4	0	-4
R	-12	-4	-1	0	3	-1
C	-16	-8	-5	-2	4	2
O	-20	-12	-9	-6	0	8

O score do alinhamento entre ABACO e BARCO é dado pelo último valor da matriz visto que este corresponde ao score do alinhamento completo e por isso é 8.

Construção do melhor alinhamento

Para construir o melhor alinhamento, o que se faz é preencher duas matrizes ao mesmo tempo: a matriz com os scores e outra matriz com um código que identifica de qual das três células possíveis é aquela que tem o maior valor. Neste caso, 0 equivale à diagonal, 1 à vertical, 2 à horizontal e -1 a parar.

	-	a	b	a	c	o
-	0	-4	-8	-12	-16	-20
b	-4	-1	0	-4	-8	-12
a	-8	0	-2	4	0	-4
r	-12	-4	-1	0	3	-1
c	-16	-8	-5	-2	4	2
o	-20	-12	-9	-6	0	8

	-	a	b	a	c	o
-	-1	2	2	2	2	2
b	1	0	0	2	2	2
a	1	0	0	0	2	2
r	1	1	0	1	0	0
c	1	1	0	0	0	0
o	1	1	0	0	1	0

Assim, para reconstuir o alinhamento, começa-se no canto inferior direito e vai-se seguindo as direções.

```
s1: -barco
s2: aba-co      score:   8
```

Alinhamento local

Quando se pretende descobrir qual é a substring comum a duas strings mas com a possibilidade de erros utiliza-se um alinhamento local. Este é semelhante ao global só que não há custos negativos na matriz e escolhe-se o maior score na matriz (e não o valor do canto inferior direito). Para reconstruir o alinhamento, começa-se a partir da célula com o valor máximo e para-se logo que se encontre uma célula de score com o valor de 0.

	-	l	a	r	i	n	g	e
-	0	0	0	0	0	0	0	0
o	0	2	0	0	2	0	0	0
t	0	0	1	2	0	1	2	0
o	0	2	0	0	4	0	0	1
r	0	0	1	4	0	3	2	2
r	0	0	0	5	3	0	5	4
i	0	2	0	1	9	5	1	4
n	0	0	1	0	5	13	9	5
o	0	2	0	0	2	9	12	8
l	0	4	1	0	2	5	8	11
a	0	0	8	4	0	1	4	7
r	0	0	4	12	8	4	3	6
i	0	2	0	8	16	12	8	4
n	0	0	1	4	12	20	16	12
g	0	0	0	3	8	16	24	20
o	0	2	0	0	5	12	20	23
l	0	4	1	0	2	8	16	19
o	0	2	3	0	2	4	12	15
g	0	0	1	5	1	4	8	11
i	0	2	0	1	9	5	4	7
s	0	0	4	0	5	8	4	6
t	0	0	0	6	2	4	10	6
a	0	0	4	2	5	1	6	9

	-	l	a	r	i	n	g	e
-	-1	-1	-1	-1	-1	-1	-1	-1
o	-1	0	-1	-1	0	-1	-1	-1
t	-1	-1	0	0	-1	0	0	-1
o	-1	0	-1	0	0	2	0	0
r	-1	-1	0	0	1	0	0	0
r	-1	-1	-1	0	0	-1	0	0
i	-1	0	-1	1	0	2	1	0
n	-1	-1	0	-1	1	0	2	2
o	-1	0	-1	0	0	1	0	0
l	-1	0	0	-1	0	1	0	0
a	-1	1	0	2	2	0	0	0
r	-1	-1	1	0	2	2	0	0
i	-1	0	1	1	0	2	2	2
n	-1	-1	0	1	1	0	2	2
g	-1	-1	-1	0	1	1	0	2
o	-1	0	-1	-1	0	1	1	0
l	-1	0	0	-1	0	1	1	0
o	-1	0	0	0	0	1	1	0
g	-1	-1	0	0	2	0	0	0
i	-1	0	-1	1	0	2	1	0
s	-1	-1	0	2	1	0	0	0
t	-1	-1	1	0	2	0	0	2
a	-1	-1	0	1	0	0	1	0

O que corresponde a:

```
s1: laring
s2: laring      score:  24
```

Exercício

1. Crie os testes para o código que vai desenvolver a seguir
2. Crie uma função que dados dois caracteres devolva 4 se estes forem iguais, 2 se forem diferentes mas vizinhos no teclado QWERTY e -1 caso contrário
3. Crie um objeto chamado `Alinhamento` com os seguintes métodos:

   new O construtor que recebe como parâmetro um hash com os argumentos

   - **s1** a string s_1
   - **s2** a string s_2
   - **alin** o tipo de alinhamento, podendo ser `global` ou `local`
   - **scoring** uma função que recebe dois argumentos e devolve o score da substituição de um dos caracteres pelo outro (caso não receba este argumento, deverá contar um por cada caractere igual e -1 por caracteres diferentes)
   - **espaco** o custo do espaçamento (deverá ser 4 por omissão)

   get_score Este método deverá devolver o score do alinhamento

   get_alin1 Este método deverá devolver a string s_1 alinhada

   get_alin2 Este método deverá devolver a string s_2 alinhada
4. (Pontos extra) Use a função AUTOLOAD para não ter que escrever os métodos `get_*`
5. (Pontos extra) Implemente um score diferente (normalmente maior) para inserir um espaçamento do que para estender esse mesmo espaçamento

Segue-se um exemplo de utilização que demonstra algumas das potencialidades do sistema.

```
use Alinhamento;

$vizinho = ... # Algo que crie um hash em que $vizinho{$x}{$y} existe se a letra $x e' vizinha no teclado QWERTY da letra $y

sub teclado {
        my ($a, $b) = @_;
        return 4 if ($a eq $b);
        return 2 if $vizinho{$a}{$b};
        return -1;
}

my $alin = new Alinhamento(s1 => "sapato", s2 => "carpatos", alin => "global", espaco => 1);
printf "s1: %s\ns2: %s\tscore: %3d\n\n", $alin->get_alin1, $alin->get_alin2, $alin->get_score;

$alin = new Alinhamento(s1 => "sapato", s2 => "carpatos", alin => "global", scoring => sub {my ($a, $b) = @_; ($a eq $b) ? 2 : -1}, espaco = 1);
printf "s1: %s\ns2: %s\tscore: %3d\n\n", $alin->get_alin1, $alin->get_alin2, $alin->get_score;

Alinhamento->new(s1 => "reparo", s2 => "fvgelwro", alin => "global", scoring => \&teclado)->print;

Alinhamento->new(s1 => "poelolurknho", s2 => "pelourinho", alin => "global", scoring => \&teclado)->print;

Alinhamento->new(s1 => "ployrinho", s2 => "poelolukno", alin => "global", scoring => \&teclado)->print;

Alinhamento->new(s1 => "ployrinho", s2 => "poelolukno", alin => "global", scoring => \&teclado, espaco => 2)->print;

Alinhamento->new(s1 => "poelolurknho", s2 => "pelouro", alin => "global", scoring => \&teclado)->print;

Alinhamento->new(s1 => "poelolurknho", s2 => "pelouro", alin => "local", scoring => \&teclado)->print;

Alinhamento->new(s1 => "otorrinolaringologista", s2 => "laringe", alin => "local", scoring => \&teclado)->print;
```

Que deveria imprimir:

```
s1: sa-pato-
s2: carpatos    score:   2

s1: sa-pato-
s2: carpatos    score:   7

s1: r--eparo
s2: fvgelwro    score:  10

s1: poelolurknho
s2: p-elo-urinho        score:  30

s1: pl-oyrinho
s2: poelolukno  score:   9

s1: p--lo-yrinho
s2: poelolu-kn-o        score:  14

s1: poelolurknho
s2: p-elo-ur---o        score:   8

s1: oelolurk
s2: pelo-uro    score:  20
```

```
s1: laring
s2: laring      score:  24
```