Universidade do Minho

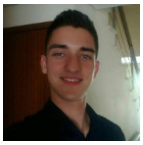
Departamento de Informática

Processamento de Linguagens

Relatório TP2 - Cartas setecentistas da Etiópia

Autores:

Alexandre Pinho (A82441)



Joel Gama (A82202)

27 de Abril de 2019

Conteúdo

1 Introdução

Este relatório surge no contexto da UC de Processamento de Linguagens do terceiro ano do Mestrado Integrado de Engenharia Informática. O trabalho proposto consiste no processamento de Cartas setecentistas da Etiópia, fazer a contagem de cartas por local, criar de um índice HTML com os anos e a cada ano associado o título e resumo das cartas esse ano, uma lista de cartas com a associação entre o numero da carta e os apelidos das pessoas relacionadas e por fim, utilizando *dot*, fazer um grafo que relaciona os autores com os destinatários.

Nas próximas páginas, é mostrado o modo como foi feito o processamento dos artigos através do *GAWK* e quais os resultados produzidos pelo programa. Por fim, é incluída uma descrição da interface por linha de comando do programa.

2 Resolução

2.1 Cartas por local

Na contagem das cartas pelo seu local, foram identificados dois casos a contemplar: uma carta pode estar associada a um ou mais locais, ou pode não estar associada a qualquer local. Assim, foi mantida uma tabela com duas dimensões, o local e o ano de escrita, com contadores para cada caso que ocorre no ficheiro.

No primeiro caso, depois de retirar os caracteres desnecessários do campo dos locais e da data, e de extrair o ano da data, incrementa-se o contador relativo ao local e à data para cada local no campo respetivo.

```
$3 ~ /\w+/ && $2 ~ /[0-9.]+/ {
    gsub(/^ +/, "", $3);
    gsub(/ +$/, "", $3);
    split($3, locais, ", ");
    gsub(/ /, "", $2);
    split($2, data, ".");
    for (l in locais) {
        gsub(/]/, "", locais[l]);
        conta[locais[l]][data[1]]++;
    }
}
```

No segundo caso, considera-se um valor especial de "NIL"para o local.

```
$3 ~ /^\s*$/ && $2 ~ /[0-9.]+/ {
    gsub(/ /, "", $2);
    split($2, data, ".");
    conta["NIL"][data[1]]++
}
```

No fim, imprime-se a lista dos locais, com o número total de cartas do local (calculado pela função `conta_datas()`), e o número de cartas enviadas em cada um dos anos.

```
END {
    for (k in conta) {
        print k ": " conta_datas(conta[k]);
        for (d in conta[k]) {
            print d " - " conta[k][d]
        }
    }
}

function conta_datas(lst) {
    total = 0
    for (d in lst) {
        total += lst[d]
    }
    return total
}
```

2.2 Índice de anos (HTML)

Para criar um ficheiro HTML com as cartas indexadas por ano, são adicionados os títulos e resumos das cartas para uma tabela com uma lista de strings (já em HTML) por cada ano, depois de retirados os espaços desnecessários do título e resumo.

```
$2 ~ /[0-9.]+/ {
    gsub(/ /, "", $2);
    split($2, data, ".");
    gsub(/^ */, "", $4);
    gsub(/^ */, "", $6);
    gsub(/ +/, " ", $6);
    anos[data[1]][size[data[1]]] = "<h1>"$4"</h1>\n" "<p>"$6"</p>\n"
    size[data[1]]++
}
```

No fim, são criados ficheiro para as cartas (um por cada), e um ficheiro `index.html`, que contém as hiperligações para os ficheiros relativos às cartas de cada ano.

```
END {
    print "<!DOCTYPE html>\n<html>\n<body>\n" > "index.html"
    print "<head>\n\t<meta charset=\"UTF-8\">\n" > "index.html"
    for (a in anos) {
        print "<p><a href=" a ".html" ">" a "</a></p>\n" > "index.html";
        print "<!DOCTYPE html>\n<html>\n<body>\n" > a ".html"
        for (i in anos[a]) {
            print anos[a][i] > a ".html"
        }
        print "</html>\n</body>\n" > a ".html"
    }
    print "</html>\n</body>\n" > "index.html"
}
```

2.3 Pessoas envolvidas

As pessoas envolvidas numa carta estão listadas no quinto campo pelo seu apelido. Assim, para saber quais as pessoas associadas com uma certa carta, identificada pelo seu número, é apenas necessário imprimir todos os pares número-apelido para cada carta, depois da limpeza dos respetivos campos.

```
    {
    gsub(/ /, "", $1);
    gsub(/ /, "", $5);
    split($5, envolv, ":");
    for (e in envolv) {
        if (envolv[e] !~ /^ *$/) {
            print $1 " : " envolv[e]
        }
    }
```

2.4 Grafo de autores e destinatários

Para criar do grafo temos de seguir uma estrutura pré definida.

```
digraph{
rankdir=LR
"a.b-c.html"-> c;
c-> d;
d-> e;
e-> e;
}
```

(Exemplo dado na aula teórica.)

Para completar o grafo são necessárias informações de dois campos, os apelidos das pessoas relacionadas com a carta ($5) e a data ($2).

Primeiro são retirados os caracteres a mais na data e nos apelidos e separar todos os valores em duas listas, respetivamente. Utilizando os dados dos apelidos criamos a ligação entre autor (infos[1]) e destinatário (infos[2]), caso estes não sejam vazios. Por fim, na ligação é imprimida a data da carta.

```
BEGIN {
    FS = ";"
    dot = "graph.dot";
    print "digraph{" > dot;
    print "rankdir = LR" > dot;
}

      { gsub(/ /, "", $2);
        split($2, data, ".");
        gsub(/ /, "", $5);
        split($5, infos, ":");
        if (infos[1] != "" && infos[2] != "") {
            print infos[1] "->" infos[2]
            "[label=\" "data[3]"-"data[2]"-"data[1]" \"]" > dot;
        }
    }

END   { print "}" > dot;}
```

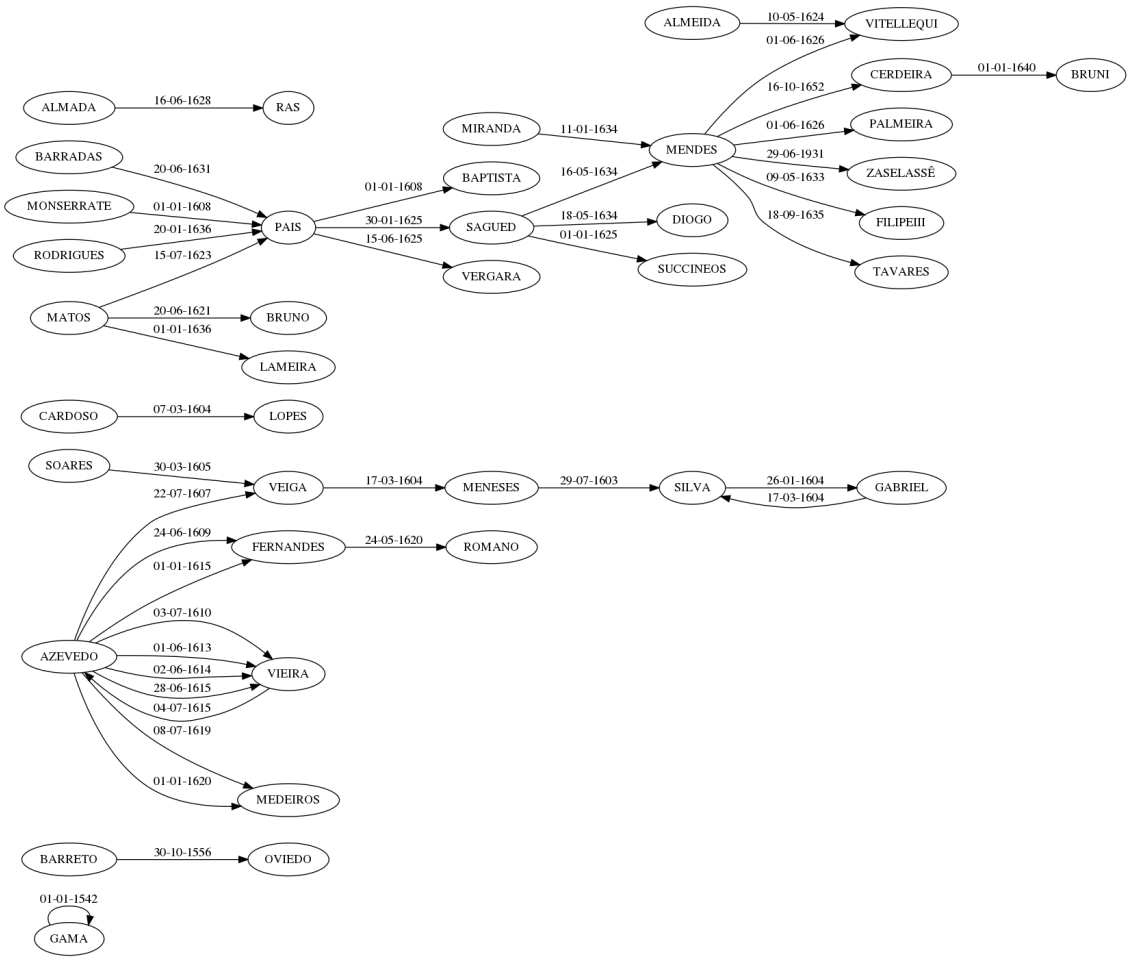


Figura 1: Grafo de autores e destinatários.

3 Utilização do programa

Para correr qualquer um dos programas descritos por este relatório, é necessária a utilização de uma distribuição do AWK. A execução é feita através do seguinte comando:

```
awk -f <programa>.awk cartasetiopia.csv
```

Onde <programa> pode ser uma das seguinte quatro opções, em ordem de apresentação neste relatório: `locais`, `anos_html`, `envolvidas`, e `dot`.

Para obter a representação visual do grafo de autores e destinatários (neste caso, uma imagem PNG), deverá ser utilizado, depois da execução do programa em AWK, o comando `dot`.

```
dot -Tpng graph.dot > graph.png
```

O grafo resultante é aquele representado na Figura 1.

4 Conclusões

Este trabalho revelou-se importante para os membros do grupo de trabalho, no sentido em que permitiu uma consolidação e interiorização de alguns conceitos e conhecimentos abordados nas aulas práticas da disciplina. Para além da aquisição de conhecimentos básicos de *GAWK*, foi também possível aprender a utilizar o *dot*, utilizado para gerar grafos.

Assim, em relação ao trabalho realizado, os objetivos definidos foram atingidos e, por isso, é feita uma apreciação positiva do trabalho.