Processamento de Linguagens
MiEI (3ºano)

Trabalho Prático nº 1 (ER + Filtros de Texto)

Ano lectivo 20/21

Objectivos e Organização

Este trabalho prático tem como principais **objectivos**:

- aumentar a capacidade de escrever *Expressões Regulares (ER)* para descrição de *padrões de frases* dentro de textos;
- desenvolver, a partir de ER, sistematicamente *Processadores de Linguagens Regulares*, ou *Filtros de Texto (FT)*, que filtrem ou transformem textos com base no conceito de regras de produção *Condição-Ação*;
- utilizar o módulo 're'—com suas funções de search(), split(), sub()—do Python para implementar os FT pedidos.

Para o efeito, esta folha contém 5 enunciados, dos quais deverá resolver um escolhido em função do número do grupo (NGr) usando a fórmula $exe = (NGr \ \% \ 5) \ + \ 1$.
Neste TP que se pretende que seja resolvido rapidamente (2 semanas), aprecia-se a imaginação/criatividade dos grupos ao incluir outros processamentos!

Deve entregar a sua solução **até segunda-feira dia 5 de Abril**. O ficheiro único com o relatório e a solução deve ter o nome '`pl20TP1grNGr`' e será submetido até ao fim da data indicada através do Bb.
O programa desenvolvido será apresentado aos membros da equipa docente, totalmente pronto e a funcionar (acompanhado do respectivo relatório de desenvolvimento) e será defendido por todos os elementos do grupo, em data a marcar.
O **relatório** a elaborar, deve ser claro e, além do respectivo enunciado, da descrição do problema, das decisões que lideraram o desenho da solução e sua implementação (incluir o código Python), deverá conter exemplos de utilização (textos fontes diversos e respectivo resultado produzido).
Como é de tradição, o relatório será escrito em LaTeX.

1 Processador de Inscritos numa atividade Desportiva

Neste exercício pretende-se trabalhar com um arquivo desportivo criado por um Organizador de *Provas de Orientação* realizadas em diferentes locais e para diferentes graus de dificuldade adaptadas a diferentes classes de participantes.

Construa, então, um ou vários programas Python para processar o ficheiro de texto '`inscritos.json`' anexo[1] conforme solicitado nas alíneas seguintes:

a) imprimir o nome (convertido para maiúsculas) de todos os concorrentes que se inscrevem como 'Individuais' e são de 'Valongo'.

b) imprimir o nome completo, o telemóvel e a prova em que está inscrito cada concorrente cujo nome seja 'Paulo' ou 'Ricardo', desde que usem o GMail.

c) imprimir toda a informação dos atletas da equipe "TURBULENTOS".

d) imprimir a lista dos escalões por ordem alfabética e para cada um indicar quantos atletas estão inscritos nesse escalão.

e) gerar uma página HTML com a lista das equipes inscritas em qualquer prova, indicando o seu nome e o número dos atletas que a constituem e que se inscreveram pelo menos uma vez numa prova; essa lista deve estar ordenada por ordem decrescente do número de atletas; além disso, cada equipe deve ter um link para outra página HTML com a informação que achar interessante sobre cada atleta indicando as provas em que cada um participou.

2 Processador de Pessoas listadas nos Róis de Confessados

Os *Róis de Confessados* são arquivos do arcebispado existentes no ADB que contém o registo de rapazes que pretendiam seguir a vida clerical e se candidatavam aos seminários, tendo-os sido possível obter uma versão digital desse arquivo.

Construa, então, um ou vários programas Python para processar o ficheiro de texto '`processos.xml`' anexo[2] conforme solicitado nas alíneas seguintes:

a) Calcular o número de processos por ano; apresente a listagem por ordem cronológica e indique o intervalo de datas em que há registos bem como o número de séculos analisados.

b) Calcular a frequência de nomes próprios (primeiro nome) e apelidos (último nome) global e mostre os 5 mais frequentes em cada século.

c) Calcular o número de candidatos (nome principal de cada processo) que têm parentes (irmão, tio, ou primo) eclesiásticos; diga qual o tipo de parentesco mais frequente[3].

d) Verificar se o mesmo pai ou a mesma mãe têm mais do que um filho candidato.

e) Utilizando a linguagem de desenho de grafos DOT[4] desenhe todas as árvores genealógicas (com base nos triplos < filho, pai, mãe >) dos candidatos referentes a um ano dado pelo utilizador.

[1] Comece por analisar com cuidado a estrutura desse dataset e perceber com detalhe a informação que ele contém.
[2] Comece por analisar com cuidado a estrutura desse dataset e perceber com detalhe a informação que ele contém.
[3] Observe que um candidato pode ter mais do que um parente conhecido, que apadrinha a sua candidatura.
[4] Disponível em `http://www.graphviz.org`

3 BibTeXPro, Um processador de BibTeX

BibTeX é simultaneamente um formato para criar uma BD textual de registos bibliográficos e uma ferramenta de formatação compatível com o processador de documentos LaTeX. BibTeX permite que se façam, no meio de um documento LaTeX, citações bibliográficas a outros documentos públicos que estejam caraterizados e classificados numa determinada BD escrita nesse formato, com o objectivo de facilitar a produção da referência e o respetivo bloco de texto que identifica o documento citado e que vai aparecer na secção final do documento LaTeX em edição.
O BibTeX foi criada por Oren Patashnik e Leslie Lamport[5] em 1985, tendo cada entrada nessa base de dados biblográfica textual o aspecto que se ilustra no exemplo a seguir:

```
@InProceedings{CPBFH07e,
  author =    {Daniela da Cruz and Maria João Varanda Pereira
               and Mário Béron and Rúben Fonseca and Pedro Rangel Henriques},
  title =     {{Comparing Generators for Language-based Tools}},
  booktitle = {Proceedings of the 1.st Conference on Compiler
               Related Technologies and Applications, CoRTA'07
               --- Universidade da Beira Interior, Portugal},
  year =      {2007},
  editor =    {},
  month =     {Jul},
  note  =     {}
}
```

De modo a familiarizar-se com o formato do BibTeX poderá consultar os ficheiros '`bibLayout.*`' que estão disponíveis no item '`Conteudo` > `Material-de-Apoio`' do Blackboard ou ainda a página oficial do formato referido (`http://www.bibtex.org/`).
Para já importa apenas que saiba que a primeira palavra (logo a seguir ao carater '`@`') designa a *categoria* da referência (havendo atualmente em BibTeX cerca de 20 categorias diferentes).
Construa, então, um ou vários programas Python para processar qualquer BD BibTex, como o ficheiro de texto '`exemplo-utf8.bib`' anexo[6] de modo a realizar as tarefas que se seguem:

a) Calcular o número de entradas por *categoria*; apresente a listagem em formato HTML por ordem alfabética.

b) Criar um índice de autores, que mapeie cada autor nos respectivos registos identificados pela respectiva chave de citação (a 1ª palavra a seguir à chaveta, que é única em toda a BD)

c) Transforme cada registo num documento JSON válido; a BD original deve ser então um documento JSON válido formado por uma lista de registos.

d) Construa um Grafo que mostre, para um dado autor (definido na altura pelo utilizador) todos os autores que publicam normalmente com o autor em causa.
Recorrendo à linguagem DOT do GraphViz[7], gere um ficheiro com esse grafo de modo a que

[5] O Doutorando de Donald Knuth que criou o LaTeX, a partir do original TeX concebido e implementado pelo seu Orientador.

[6] Comece por analisar com cuidado a sua estrutura para perceber com detalhe o formato e a informação que ele contém.

[7] Disponível em `http://www.graphviz.org`

possa, posteriormente, usar uma das ferramentas que processam DOT [8] para desenhar o dito grafo de associações de autores.

4 Conversor genérico de CSV para JSON

Neste enunciado pretende-se fazer um conversor de um qualquer ficheiro gravado em formato CSV (*Comma Separated Values*, original e tipicamente usado para descarregar uma Folha de Cálculo num ficheiro de texto) para o formato JSON[9](um formato textual neutro e muito simples, baseado no conceito de um conjunto de pares { "campo": "valor"}, concorrente do XML enquanto sistema de exportação/transferência de dados entre aplicações para assegurar a interoperabilidade).

Para poder realizar a conversão pretendida, é importante saber que a primeira linha do CSV dado funciona como cabeçalho que descodifica a que correspondem os valores que vêm nas linhas seguintes. Até aqui nada de novo ..., mas é claro que leva mais uns ingredientes.

O dataset dado poderá ter listas aninhadas nalgumas células. Mas nesse caso o cabeçalho terá um asterisco '*' a seguir ao nome do respetivo campo. Se nada mais for colocado o conversor deverá converter cada valor dessa coluna numa lista em JSON.

Mas a seguir ao asterisco pode haver uma função de agregação: `sum`, `avg`, `max`, `min`. Aí o conversor terá de aplicar a operação de fold respetiva sobre a lista e produzir o JSON de acordo.

Exemplos:

Este CSV

```
número;nome;curso;notas*
A71823;Ana Maria;MIEI;(12,14,15,18)
A89765;João Martins;LCC;(11,16,13)
A54321;Paulo Correia;MIEFIS;(17)
```

Daria origem a este JSON:

```
[
    {
        "número": "A71823",
        "nome": "Ana Maria",
        "curso": "MIEI",
        "notas": [12,14,15,18]
    },
    {
        "número": "A89765",
        "nome": "João Martins",
        "curso": "LCC",
        "notas": [11,16,13]
    },
    {
        "número": "A54321",
        "nome": "Paulo Correia",
```

[8]Disponíveis em `http://www.graphviz.org/Resources.php` ou a ferramenta Web `http://www.webgraphviz.com/`

[9]Ver mais em `https://www.json.org` ou `https://jsonformatter.curiousconcept.com/`

```
        "curso": "MIEFIS",
        "notas": [17]
    }
]
```

E este

```
número;nome;curso;notas*avg
A71823;Ana Maria;MIEI;(12,14,15,18)
A89765;João Martins;LCC;(11,16,13)
A54321;Paulo Correia;MIEFIS;(17)
```

Daria origem ao seguinte resultado

```
[
    {
        "número": "A71823",
        "nome": "Ana Maria",
        "curso": "MIEI",
        "notas_avg": 14.99
    },
    {
        "número": "A89765",
        "nome": "João Martins",
        "curso": "LCC",
        "notas_avg": 13.33
    },
    {
        "número": "A54321",
        "nome": "Paulo Correia",
        "curso": "MIEFIS",
        "notas_avg": 17.00
    }
]
```

5 Machine Learning: datasets de treino

Hoje em dia, a área de *Machine Learning* está na moda e as suas metodologias e tecnologias são usadas em muitas áreas.

A maior parte dos algoritmos de *Machine Learning* têm de ser treinados com um dataset especialmente anotado à mão e depois testados sobre outro dataset anotado para ver se o que a máquina descobre é o mesmo que um ser humano faria à mão.

Estes datasets anotadas são uma fonte preciosa de informação, para treino dos algoritmos, mas também podem ser usados para outros fins. Neste problema, terás de programar a extração de vários elementos informativos de um destes datasets. Nomeadamente do dataset de treino disponibilizado pela Google para treino docente TensorFlow: '`train.txt`'.

Apresenta-se a seguir um excerto deste dataset:

```
B-GENRE        science
I-GENRE        fiction
I-GENRE        films
O        directed
O        by
B-DIRECTOR        steven
I-DIRECTOR        spielberg
...
O        move
O        that
B-ACTOR        frank
I-ACTOR        sinatra
O        was
O        in
...
```

Cada linha pode começar por:

B-categoria Begin: marca o início de uma categoria;

I-categoria In: marca a continuação de uma categoria;

O Other: marca algo que não se quer anotar/recolher.

Do excerto acima teríamos as seguintes categorias com os respetivos elementos:

```
GENRE:        science fiction films
DIRECTOR:        steven spielberg
ACTOR:        frank sinatra
```

Ao longo do dataset cada categoria vai estando associada a mais elementos.
Neste projeto, pretende-se que cries um website muito simples com a seguinte estrutura:

- Uma página principal onde aparecem todas as categorias existentes no dataset, à frente de cada categoria deverá aparecer um contador indicando quantos elementos estão classificados como sendo dessa categoria;
- Clicando numa categoria, devemos saltar para a página dessa categoria onde devem aparecer listados os elementos dessa categoria (para cada elemento podes indicar em que linha do dataset ele foi capturado); Nestas páginas deverá haver um link para retornar à página principal.