Projeto MPEI

ENG. COMPUTADORES E TELEMÁTICA

Mário Liberato | 84917
Jorge Oliveira | 84983

DETI

DESCRIÇÃO

Este projeto tem como objetivo receber candidaturas a empregos numa hipotética empresa, eliminar os utilizadores cujas competências sejam menores que as necessárias e colocar os utilizadores relevantes por ordem de relevância num ficheiro CSV, o qual pode ser aberto com um programa como o LibreOffice Calc ou o Microsoft Office Excel (É de se notar que no caso deste último, há que se adicionar como primeira linha no ficheiro "sep=," para que este seja mostrado correctamente, ou mudar o delimitador manualmente. Como este comportamento não é standard do formato CSV, o programa não o faz automaticamente) e então analisado por uma pessoa, que então contactaria os utilizadores relevantes.

Uma descrição mais detalhada do programa pode ser encontrada no outro documento incluído com o projecto.

Para utilizar o programa o utilizador deverá executá-lo com o comando correcto ou utilizar os scripts fornecidos para o fazer.

Objetivo do Projeto

Como correr o programa

Para correr este programa o utilizador tem ao seu dispor *scripts* com o comando correcto para executar a classe certa.

- ***GenData***: gera um ficheiro "table.tb" com certo número de utilizadores, que é solicitado na abertura do *script*. O ficheiro contém informação aleatória relativamente a cada utilizador com: ID, Nome, Habilidades, Nível de Educação, Idade e Número de Telemóvel;

- ***Main Interface***: processa a informação contida sobre os utilizadores (possíveis candidatos) criando dois ficheiros: "X.csv" e "X_perfect.csv" (onde X é o nome do emprego ao qual os dados se referem). O ficheiro *perfect* tem os utilizadores com as habilidades pretendida pelo chefe, o outro ficheiro contém os candidatos com maior índice de semelhança, de forma ordenada. Este programa equivale ao que uma empresa usaria para selecionar potenciais trabalhadores;

- ***Test***: Testar possíveis erros. Como por exemplo, falso positivos no *Bloom Filter* (este último utiliza a classe BloomFilterTest em vez disto) ;

- ***Job Application Form***: Interface gráfica onde um candidato a emprego preencheria a sua informação para registar na base dados. Cada entrada neste programa adicionará uma entrada em "table.tb".

Módulos Implementados

O trabalho foi realizado utilizando linguagem Java. Os algoritmos de módulos como *Bloom Filter* e *MinHashing* foram baseados nos exercícios realizados ao longo das aulas em Matlab.

Os módulos principais são:

- *Main*
  - Bloom Filter (BloomFilter)
  - MinHash e distância de Jaccard (MinHash2 e Jaccard);
- Data Generator (DataGen)
- *Job Application* (JobApplicationUI)
- Testes (Test e BloomFilterTest)

MAIN

O Main, é o módulo que envolve maior parte do projeto. A informação, previamente dada ou gerada é processada e é devolvido dois ficheiros CSV: um que apresenta os utilizadores (e a sua informação) com um filtro preciso, ou seja, com as habilidades pretendidas; O outro ficheiro possuí utilizadores com um índice de semelhança elevado, organizados.

Está dividido em vários submódulos, os mais relevantes:

Bloom Filter

Existem pequenos *Bloom Filters* para cada utilizador, relativos às suas habilidades, desta forma é possível pesquisar candidatos com uma certa habilidade.

O número de *hash functions(k)* utilizado foi através da fórmula de *"Optimal number of hash functions" (*Wikipedia*)* :

$k = \frac{m}{n} \ln(2)$ | m = Tamanho *Bloom Filter* n = Nr. de membros

Como o tamanho do *Bloom Filter* é 10*Nr. de membros $k \approx 7$.

MinHash & Jaccard Similarity

Este modulo é crucial para calcular o índice de semelhança para uma dada empresa encontrar utilizadores com habilidades pretendidas.

O *MinHashing* consiste em determinar uma assinatura *hash* de cada documento relativo aos utilizadores usando *hash functions*, o número do mesmo corresponde ao tamanho da assinatura. A assinatura é um *array* com valores *hash,* o tamanho desse array é relativo à media de elementos do documento. Este processo pode reduzir bastante o tempo para o passo seguinte se o documento usar *strings.*

A partir do cálculo da assinatura, é utilizada a Semelhança de *Jaccard*. E é a partir destes processos que se podem encontrar utilizadores com uma dada semelhança ou requerimento

DATA GENERATOR

Este módulo gera um ficheiro com utilizadores e informação respetiva, aleatoriamente. A informação criada consiste em: ID, Nome, Habilidades, Nível de Educação, Idade e Número de Telemóvel.

JOB APPLICATION

Esta aplicação é o exemplo do que um centro de emprego, *website*, ou empresa poderia utilizar para adicionar utilizadores e a sua informação a uma base de dados.

TESTES

Fontes

Wikipedia(Bloom Filters, melhor número de hash functions)
https://en.wikipedia.org/wiki/Bloom_filter#Optimal_number_of_hash_functions