Projeto MPEI

Eng. Computadores e telemática

Mário Liberato | 84917

Jorge Oliveira | 84983

DETI

Resumo

Este projeto tem como objetivo ……………………………………….

Para utilizar o programa o utilizador deverá utilizar os scripts disponíveis na pasta */scripts*

# Objetivo do Projeto

# Como correr o programa

Para correr este programa o utilizador tem ao seu dispor *scripts* para uso mais rápido na pasta */scripts*.

- ***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”. 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 *script* equivale ao que uma empresa usaria para selecionar potenciais trabalhadores;

- ***Test***: Testar possíveis erros. Como por exemplo, falso positivos no *Bloom Filter*;

- ***Job Application Form***: Interface gráfica onde um candidato a emprego preencheria a sua informação para registar na base dados. Cada entrada neste *script* 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;
* Sort Skills;
* MinHash e distância de Jaccard;
* Data Generator
* *Job Application*
* Testes

## *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](https://en.wikipedia.org/wiki/Bloom_filter#Optimal_number_of_hash_functions)*)* :

| m = Tamanho *Bloom Filter* n = Nr. de membros

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

#### Sort Skills

Este módulo apenas pretende organizar as habilidades de um dado utilizador.

#### 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](https://en.wikipedia.org/wiki/Bloom_filter%23Optimal_number_of_hash_functions)