Trabalho Prático 1 - AED

Este trabalho prático tem como objetivo implementar algoritmos de *enriquecimento* de dados em dados tabulares, utilizando técnicas de *fuzzy matching*.

1-Descrição do problema

Os dados são um recurso valioso para as empresas e o enriquecimento dos dados é uma tarefa importante para aumentar o nível de informação e melhorar a tomada de decisão. O enriquecimento dos dados pode ser obtido simplesmente, juntando novas tabelas às tabelas existentes, no entanto, os campos das novas tabelas podem não ser coincidentes aos campos existentes, não sendo possível fazer o *match* para a ligação entre tabelas.

O *fuzzy matching* é uma técnica que pode ser utilizada para identificar as correspondências entre dados com variações ou erros de digitação. Neste contexto, a tarefa de enriquecimento de dados com *fuzzy matching* envolve a identificação de dados semelhantes em tabelas distintas, mas que representam a mesma entidade ou objeto.

Para realizar essa tarefa é necessário utilizar métricas de semelhança entre palavras, de modo a considerar a similaridade entre os dados, considerando variações como erros de digitação, abreviações, sinónimos, entre outros.

Observe a seguinte figura, que mostra um exemplo simples sobre o *enriquecimento* de dados.

country	population_in_millions
England	55.98
Scotland	5.45
Wales	3.14
United Kingdom	67.33
Northern Ireland	1.89

⋈

country	GDP_per_capita
Northern Iland	24900.00
Wles	23882.00
ScotInd	37460.00
EnglInd	45101.00
United K.	46510.28

→

country	population_in_millions	GDP_per_capita	Levenshtein
England	55.98	45101.00	92
Scotland	5.45	37460.00	93
Wales	3.14	23882.00	89
United Kingdom	67.33	46510.28	70
Northern Ireland	1.89	24900.00	93

Observe que apesar dos nomes não corresponderem, foram escolhidos os nomes mais próximos para a junção das duas tabelas. Assim, ao invés de uma simples junção, foi realizada uma *fuzzy join*.

Note também que neste caso é utilizada a distância de Levenshtein, como métrica para calcular o grau de semelhança entre duas palavras. Duas palavras apresentam maior grau de semelhança, quanto maior for a sua distância de Levenshtein.

O *score* obtido na junção difusa é dado pela formula seguinte, utilizando a distância de Levenshtein:

$$ration = (s1.length() + s2.length() - distance(s1, s2))/(s1.length() + s2.length())$$

Por exemplo, tendo `s1="England"` e `s2="Englnd"` é obtido a `ration` de 0.923076923076923, em percentagem $ration = 92$, tal como apresentado na tabela.

2-Dados fornecidos

Para o desenvolvimento deste trabalho prático as tabelas exemplo e um exemplo de utilização.

1. Exemplo da tabela `population`:

```
country,population_in_millions
England,55.98
Scotland,5.45
Wales,3.14
United Kingdom,67.33
Northern Ireland,1.89
```

2. Exemplo da tabela `gpt`:

```
country,GDP_per_capita
Northern Iland,24900.0
Wles,23882.0
Scotlnd,37460.0
Englnd,45101.0
United K.,46510.28
```

3. Exemplo de utilização da aplicação:

```
Caso de uso fuzzy-join:

fuzzy-join --filename1=population.csv --filename2=gpt.csv
    --name1=country --name2=country --distance=levenshtein
```

```
O resultado dever ser:

country,population_in_millions,GDP_per_capita,Levenshtein
England,55.98,45101.0,92
Scotland,5.45,37460.0,93
Wales,3.14,23882.0,89
United Kingdom,67.33,46510.28,70
Northern Ireland,1.89,24900.0,93
```

3-Tarefa

Dados dois *datasets*, com um ou mais campos de ligação, aplique uma junção difusa, considerando uma dada métrica.

Os *datasets* devem ser lidos desde o formato cvs (*comma separate values*) e para representar uma tabela, deve utilizar um `ArrayList` de `HashMap.`

A métrica é a medida de comparação entre duas palavras e deve generalizar o algoritmo para outras distâncias, tais como Damerau-Levenshtein, Hamming, Jaccard e Cosine.

Deve apresentar como output, o resultado da junção difusa, mais a coluna com o resultado do cálculo da métrica associada, no formato csv.

Realize um termo de comparação entre as várias distâncias e identifique possíveis casos de uso, para cada uma delas. Para obter resultados mais robustos, aplique o algoritmo a *datasets* com muitos registos, faça comparação entre verdadeiros e falsos *matches* e visualize os resultados em gráficos.

Faça uma análise ao algoritmo, sobre a robustez (aplica-se em todos os casos) e eficiência (o processamentos dos dados é rápido ou existem otimizações a realizar...).

Finalmente, analise os dados, discuta os resultados e conclua sobre o projeto.

3-Requisitos

(a) Utilize os princípios e técnicas estudadas em AED.

(b) Siga uma metodologia de desenvolvimento baseado em testes (*TDD - Test Driven Development*).

4-Relatório

O TP1 incluiu o desenvolvimento de um relatório, que deve conter:

(a) Uma descrição do problema e a abordagem do grupo para a sua resolução.

(b) O desenho de um diagrama de classes, com a análise realizada.

(c) O pseudo-código do algoritmo desenvolvido.

(d) Resultados.

(e) Discussão.

(f) Conclusão.

5-Entrega

O trabalho prático deve ser entregue em formato digital, incluindo o código do projeto e o relatório, com indicação dos nomes dos elementos do grupo (máximo 2 elementos), para o email *tiago.candeias@ismat.pt*, até ao dia 15-05-2023.

6-Avaliação

Esta parte do trabalho vale 50% da nota da componente prática da unidade curricular. Os critérios de avaliação são: clareza do texto, simplicidade do algoritmo, clareza das explicações, qualidade do código e comentários no código.