Relatório implementação clusteringRealities

Grupo 31

Diogo Alexandre Fernandes Valente, 53481

Diogo Miguel dos Santos Fernandes, 54967

Neste projeto, o Diogo Valente implementou parte do ficheiro clusteringRealities, no ficheiro kMeansClustering implementou a função do kmeans, a função do updateCentroid da classe Cluster, a função readCandidatesFile, enquanto o Diogo Fernandes implementou parte do ficheiro clusteringRealities, no ficheiro kMeansClustering implementou a função readTilesFile, translateTitleToScore, translateToFeatureVector, getRealCentroid, processOutputString, writeFile, __eq__ da classe Example e as funções __init__ da classe Cluster.

Implementação:

- **readCandidatesFile:** método que lê e processa o ficheiro de candidatos, retornando assim quatro listas: lista dos nomes dos candidatos, lista de listas com os títulos dos candidatos, lista dos nomes dos exemplars, lista de listas com os títulos dos exemplars.
- **readTitlesFile:** método que lê e processa o ficheiro onde se encontra os títulos possíveis para os candidatos. Retornado, uma lista com os títulos e outra com as pontuações equivalentes de cada título.
- **translateTitleToScore:** método na qual faz a conversão de um título para uma pontuação.
- **translateToFeatureVector:** método na qual recebe a lista de títulos de um candidato e retorna uma lista com as pontuações equivalentes.
- **getRealCentroid:** método que encontra o candidato mais próximo do centroid atual, insere este candidato como novo centroid, retirando assim também como candidato deste cluster.
- **processOutputString:** método que processa a informação dos clusters e cria o texto para escrever no ficheiro de saída.
- **writeFile:** método que produz o ficheiro de saída.
- **Class Example:**
    - **__eq__:** método na qual verifica se o próprio objeto Example é equivalente a outro objeto Example passado pelos argumentos da função.
- **Class Cluster:**
    - **__init__:** foi acrescentado um argumento a este método, centroid, isto para que possa ser criado um cluster com o centroid já predefinido, em vez de aleatório.
    - **updateCentroid:** método que atualiza o centroid do cluster.

Relatório implementação clusteringRealities

Grupo 31

Diogo Alexandre Fernandes Valente, 53481

Diogo Miguel dos Santos Fernandes, 54967

- **kmeans:** foi acrescentado um argumento a este método, centroids, isto para que possa aceitar centroids iniciais já predefinidos, em vez de aleatórios. Atualiza cada cluster e verifica se o centroid mudou.
- **clusteringRealities:** ficheiro na qual faz a instruções principais do projeto, utilizando o ficheiro kMeansClustering como auxiliar.
    - **Step 0:** lê os argumentos da linha de comandos.
    - **Step1:** 1º parte: lê os ficheiros de títulos e de candidatos, guardando os nomes dos candidatos e/ou de exemplars (caso aplicavel) e também os títulos destes. 2º parte: faz as conversões dos títulos dos candidatos e/ou exemplars (caso aplicável) para pontuações.
    - **Step2:** faz a criação de Examples com a informação obtida no passo anterior (Step1).
    - **Step3:** faz a construção dos clusters, utilizando a função kmeans se o ficheiro de candidatos conter exemplars, caso contrário utiliza a função trykmeans.
    - **Step4:** faz a procura do candidato mais próximo ao centroid e atualiza o centroid com o candidato mais próximo, isto em cada um dos clusters existentes.
    - **Step5:** 1º parte: faz o processamento da informação de cada cluster e gera o texto para escrever para o ficheiro de saída. 2º parte: produz o ficheiro de saída.

Funcionalidades que ficaram por implementar:

- Caso o cluster seja composto por um único elemento, este é considerado o próprio centroid deste cluster e sendo assim ficou por implementar uma solução que resolva este problema.

Erros conhecidos:

- Na execução do programa, caso os ficheiros de input não estejam na mesma diretoria do clusteringRealities.py, irá dar o erro: FileNotFoundError: [Errno 2] No such file or directory.