Guilherme Caeiro de Mattos

**Modelo Vetorial**

O modelo vetorial implementado nesta tarefa consiste em um modelo TF-IDF conforme descrito em Baeza-Yates e Ribeiro-Neto (1999). Nele, o cálculo do TF (*term frequency*) é realizado através da Equação 1, onde *freqi,j* é o número de ocorrências do termo *ki* no documento *dj*, e *maxl freql,j* é o número de ocorrências do termo mais frequente no documento.

*fi,j = freqi,j / (maxl freqi,j)* (1)

Já o IDF (*inverse document frequency*) é calculado como na Equação 2, onde *N* é o número de documentos disponíveis e *ni* é o número de documentos em que o termo *ki* ocorre. O TF-IDF, por sua vez, resulta da multiplicação das frequências *fi,j* por seus respectivos *idf*’s, dando origem a uma matriz do peso *wi,j* dos termos *ki* em cada documento *dj*. Esse cálculo é efetuado conforme a Equação 3.

*idfi = log (N/ni)* (2)

*wi,j = fi,j \* idfi* (3)

Quanto aos pesos *wi,q* dos termos nas queries, o cálculo é efetuado através da Equação 4, onde *freqi,q* é o número de ocorrências do termo *ki* na query *q*, e *maxl freql,q* é o número de ocorrências do termo mais frequente na query.

*wi,q = (0.5 + ((0.5 \* freqi,q) / (maxl freqi,q)) \* idfi* (4)

Para verificar quais documentos são pertinentes às queries, é calculada a similaridade entre o vetor = (w1, …, wn) dos pesos dos termos da query ao vetor j = (w1, …, wn) dos pesos dos termos documento dj (representado pela linha *j* da matriz do TF-IDF). Esse cálculo se dá através da Equação 5.

*sim(j, ) = (j \* ) / (||j||2 \* ||||2)* (5)

Quanto a representação do modelo no código da tarefa, ela ocorre através da classe “VectorModel” presente no arquivo “indexer.py”. Essa um objeto dessa classe recebe o TF-IDF já calculado (o que ocorre na classe “Indexer”) e armazena a matriz resultante para efetuar os cálculos de similaridade. Para efetuar esse cálculo, essa classe possui o método “evaluate\_query”, que recebe uma query como parâmetro e retorna a similaridade entre a query e todos os documentos conhecidos. Quando salvo em disco, esse modelo é serializado sob o formato do [*pickle*](https://docs.python.org/pt-br/3/library/pickle.html) e recebe o nome “vecmodel.pkl”.

**Referências**

R. Baeza-Yates e B. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press, New York, 1999.