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 $freq_{i,j}$ é o número de ocorrências do termo k_i no documento d_j, e $max_l\ freq_{l,j}$ é o número de ocorrências do termo mais frequente no documento.

$$f_{i,j} = freq_{i,j} / (max_l\ freq_{i,j}) \quad (1)$$

Já o IDF (*inverse document frequency*) é calculado como na Equação 2, onde N é o número de documentos disponíveis e n_i é o número de documentos em que o termo k_i ocorre. O TF-IDF, por sua vez, resulta da multiplicação das frequências $f_{i,j}$ por seus respectivos *idf*'s, dando origem a uma matriz do peso $w_{i,j}$ dos termos k_i em cada documento d_j. Esse cálculo é efetuado conforme a Equação 3.

$$idf_i = log\ (N/n_i) \quad (2)$$

$$w_{i,j} = f_{i,j} * idf_i \quad (3)$$

Quanto aos pesos $w_{i,q}$ dos termos nas queries, o cálculo é efetuado através da Equação 4, onde $freq_{i,q}$ é o número de ocorrências do termo k_i na query q, e $max_l\ freq_{l,q}$ é o número de ocorrências do termo mais frequente na query.

$$w_{i,q} = (0.5 + ((0.5 * freq_{i,q}) / (max_l\ freq_{i,q})) * idf_i \quad (4)$$

Para verificar quais documentos são pertinentes às queries, é calculada a similaridade entre o vetor $\overline{q} = (w_1, \ldots, w_n)$ dos pesos dos termos da query ao vetor $\overline{d_j} = (w_1, \ldots, w_n)$ dos pesos dos termos documento d_j (representado pela linha *j* da matriz do TF-IDF). Esse cálculo se dá através da Equação 5.

$$sim(\overline{d_j}, \overline{q}) = (\overline{d_j} * \overline{q}) / (||\overline{d_j}||_2 * ||\overline{q}||_2) \quad (5)$$

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* e recebe o nome “vecmodel.pkl”.

Referências

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