Luan Coelho Vieira da Silva

IMPLEMENTAÇÃO DE UM SISTEMA DE RECUPERAÇÃO EM MEMÓRIA SEGUNDO O MODELO VETORIAL

Para este trabalho foi utilizado o modelo vetorial TF-IDF. O termo TF (term frequency) é calculado da seguinte forma: $\mathrm{TF}(t,d) = \frac{f(t,d)}{\sum_k f(k,d)}$.

Na equação acima, $f(t,d)$ representa a frequência do termo t no documento d, e k varia entre os termos presentes no documento d.

Já o termo IDF (inverse document frequency) $\mathrm{IDF}(t) = \log\left(\frac{N}{df(t)}\right)$, com N representando o total de documentos na coleção e $df(t)$ o número de documentos que contém o termo t.

O peso do $w(t,d)$do termo t no documento d é dado por $w(t,d) = \mathrm{TF}(t,d)\,.\,\mathrm{IDF}(t)$.

Esta mesma lógica poderia ser aplicada para o cálculo do TF-IDF na consulta. Porém, optou-se por utilizar peso 1 para os pesos na consulta. Isto é, seja t um termo presente na consulta q. Então $w(t,q) = 1$.

Se o termo t_i não pertence ao conjunto de termos da consulta q_j então $w(t_i, q_j) = 0$.

Para definir quais documentos serão recuperados pela busca é calculada uma medida de similaridade entre o vetor de pesos dos termos da consulta com cada um dos vetores dos documentos, de forma a recuperar os mais similares (classificação em ordem decrescente).

A similaridade de um documento e uma consulta é calculada por $sim(d_j, q) = \frac{\vec{d_j} \cdot \vec{q}}{|\vec{d_j}| \times |\vec{q}|}$, com $\vec{d_j}$ sendo o vetor representando os pesos do modelo TF-IDF para os termos considerados (Baeza-Yates, Ribeiro-Neto, et al., 1999).

No código deste trabalho, o modelo é inicializado no módulo 3 – indexador e é calculado a partir do método *TfidfVectorizer* importado de *sklearn.feature_extraction.text*. A matriz TF-IDF(t,d) resultante possui uma linha para cada documento e uma coluna para cada termo considerado, e é esparsa (a maioria dos elementos possui valor 0).

O modelo, já ajustada aos dados, é salvo para replicabilidade utilizando o pacote *pickle*, ficando disponível em raiz/RESULTS/ ModeloTFIDF.pkl.

Em seguida, no módulo 4 – busca, é criado o vetor de pesos dos termos para as consultas, no qual o termo possui peso 1 se está na consulta. O resultado para cada combinação de consulta e documento é salvo em raiz/RESULTS/ RESULTADOS.csv. O cálculo de similaridade é feito conforme explicado anteriormente, utilizando *cosine_similarity* de *sklearn.metrics.pairwise*.