Web Information Retrieval (67782)
Ex3: Review Search

Submitted by: Daniel Kerbel (CSE: danielkerbel, ID: DELETED)

1 Product Search

The search will occur in two steps:

1. First, finding a list of up to k products which are relevant based on the query. To do this, I'll define the relevancy of a product p with respect to a query q as follows:

$$\text{relevancy}(q,p) = \frac{\sum_{r \in \text{reviewsForProd}(p)} \text{documentQueryRank}(r,q)}{|\text{reviewsForProd}(p)|}$$

   Where `documentQueryRank` can be any query-document ranking function(My implementation will use `lnn.ltc` as I already implemented it previously) and `reviewsForProd(p)` is the set of reviews that belongs to product p.

   Intuitively, if more reviews for a certain product match the given query, then it's more likely the query refers to said product. Taking a mean ensures that less popular products(which have less reviews) but higher match scores, will be deemed more relevant than popular products which by sheer volume might have a lot of weak matches to said query.

2. Secondly, once we have retrieved the relevant products, they will be ranked as follows:

$$\text{rank}(p) = \sum_{r \in \text{reviewsForProd}(p)} \frac{\text{score}_r \cdot \text{helpfulness}_r}{\sum_{r \in \text{reviewsForProd}(p)} \text{helpfulness}_r}$$

   That is, the rank of a product p is the weighted arithmetic mean of its reviews' scores, using their helpfulness values as weights. The idea here is that, at least from my experience, reviews deemed less helpful tend to imply either delivery issues(which might not have anything to do with the product itself) or improper usage/expectations from the product.