

Problem 6

Custom Relevance Scorer Implementation

I developed a custom relevance scoring function within the YourRanker class, which extends the RelevanceScorer base class. This scorer integrates several key components to evaluate document relevance effectively:

1. Term Frequency (TF): Measures how often a query term appears in a document, increasing the relevance score as TF increases.
2. Inverse Document Frequency (IDF): Assigns higher importance to rare terms across the corpus, ensuring that common terms contribute less to the relevance score.
3. Document Length Normalization: Adjusts scores based on the document's length relative to the average, preventing longer documents from being unfairly favored.
4. Position Weighting: Gives additional weight to terms appearing earlier in the document, based on the assumption that early mentions are more indicative of relevance.

Design Considerations with Respect to Relevance Axioms

My relevance scorer adheres to several fundamental axioms:

1. TF Axiom: Higher term frequency within a document boosts its relevance.
2. IDF Axiom: Rare terms across the corpus are weighted more heavily, enhancing discriminative power.
3. Length Normalization: Balances the influence of document length, ensuring fair comparison between documents of varying sizes.
4. Position Axiom: Prioritizes terms found earlier in the text, aligning with the notion that introductory content is often more relevant.

Thought Process

I based the scorer on the BM25 framework, known for its effectiveness in information retrieval. By adding a positional component, I aimed to incorporate the intuitive idea that the placement of terms within a document can impact its relevance. The parameters $b=0.75$ and $k1=1.2$ were chosen to balance term frequency saturation and length normalization, following common BM25 configurations. Although the positional weighting is an experimental addition, it provides valuable insights into how term placement affects ranking performance.

In []: