Explanation for Implementation of Custom Ranker

Cameron Milne

SI 650 HW #2

For the CustomRanker() formula, I decided to implement a variation of the BM25Ranker() with a consideration of the importance of highly relevant key words.

I used Spacy’s entity recognition labels to assign key terms or phrases (before analyzing a term with Pyserini) for both the android document json file and the android query csv file. The range of entity labels are expansive though, so over time I discovered that many weren’t as important. Labels like CARDINAL, ORDINAL, MONEY, QUANTITY, and DATES weren’t particularly useful due to their ubiquity and ambiguity. More important labels were PERSON, NORP, ORG, GPE, and PRODUCT. For example, one frequent product was “Android” which was useful for matching relevant documents with queries that consisted of the same label. These key phrases, I suspected, would be useful in determining which documents were more relevant than an untuned regular BM25.

Within the scoring subclass of CustomRanker(), for every match of a key and value in the relevant entity labels, I increased a custom weight by 0.25, which I multiplied by the term frequency and added to the document length normalization. This resulted in a small boost in the score.