Lab 1 - Boolean and TF-IDF Matching

Instructor
Parth Mehta (parth_mehta@daiict.ac.in)

Teaching Assistants
Adarsh Gupta (202411083@daiict.ac.in),
Bhavesh Baraiya (202101241@daiict.ac.in)

August 2024

Lab Manual

Topics Covered from the Introduction to IR book (Manning et. al): Pre-processing (Section 2.2), Boolean Matching (Section 1.3), Vector Space scoring (Section 6.3).

You are given a dataset consisting of approx 32,000 news articles. The data is structured in JSON format; each article has four fields: id, title, summary and text. In Lab 1 you created a boolean and tf-idf index from the articles. For this lab session, we will use title fields as a query to search the boolean and tf-idf index created previously.

Task 1: Boolean Search: For boolean search use the disjunction (OR) operator for querying. This means documents which have at least one of the terms from the query are relevant. Extend this matching model to a scoring function by counting the number of terms in the query appearing in the document. Rank documents based on that score.

Task 2: Vector space matching: Compute the tf-idf scores for each document for a given query and rank based on the scores.

You are expected to complete the list of tasks mentioned below *during the lab hours*. You can use existing python libraries for preprocessing (NLTK or Spacy) and basic matrix manipulation (e.g. numpy). For all other problems, you are expected to write a solution from scratch. Specifically you **can not** use tools like scipy or scikit learn to vectorize the data.

Note: The use of GPT for such trivial tasks is generally frowned upon and highlights your lack of interest or/and ability. Also since the instructor uses it almost daily in his other life (to solve real problems, not tf-idf) he can easily detect it with a few simple questions. Save yourself some embarrassment.

Advanced Exploratory Topics

This lab is designed as a warm-up exercise and some of you might find it too easy. In that case you can look ahead and experiment with the following problems, which we will cover in a future lab session.

1. Explore Pyterrier
2. Implement preprocessing and tf-idf indexing pipeline in PyTerrier
3. Compare the vocabulary size, index size and tf-idf values from the index created by PyTerrier with the one created in the previous exercise.
4. [**new**] Perform query matching and ranking using pyterrier.