NYU TANDON SCHOOL OF ENGINEERING

Sensing the world; designing the future.

Understanding and Improving Document Reordering Techniques

Sandy Zhang[1], Joshua Lee[2], Torsten Suel[3]

1. Undergraduate student, NYU Tandon 2. Undergraduate student, NYU Tandon 3. Professor, NYU Tandon Computer Science and Engineering Department

Abstract

Our project is to understand and analyze different document reordering techniques for the execution of search engine queries. The algorithms that we have studied are some of the current state of the art document reordering techniques that perform well in the aspect of speed and data size, such as the BP method, the Slashburn method, and the TSP method. Our goal is to improve these techniques for faster and better retrieval.

Background

The primary aim of a search engine is to provide relevant documents in response to a user query rapidly and efficiently. This is especially challenging to do considering the number of documents that need to be considered in queries. In most search engines, the dominant data structure that stores document/term information is the inverted index or some variation. This index structure contains a list of terms, and for each term, a corresponding list of document IDs (referred to as postings). A part of the index structure that is often overlooked but critical towards compression is document reordering. Rather than storing each document ID as is, most compression schemes store IDs relative to each other. Hence it is ideal to have some form of ‘clustering’ effect where documents with similar inverted indexes can be placed next to each other.

Methodology

The scope of our research primarily focused on building improvements to the BP algorithm. BP is a recursive algorithm using graph bisection. Hence it splits the set of documents in half and then uses a cost function to determine whether a document should go to the other side or not. Our improvements to BP were explored in two directions, either using some other document reordering algorithm in the base recursive call of BP or to implement a different cost function.

We chose the latter approach, switching from BP’s classical log model based on gaps between documents to a model that uses measurements of the actual compressed size of the inverted lists.
For each inverted list, we would record how long the list is, the number of documents, and the actual resulting size in bytes. These three parameters would be recorded for each recursive call and
stored in a table ordered by the length of the inverted list. A majority of the entries in the table would be left blank, but this would be resolved by using a form of linear interpolation between the nearest valid table entries

Future Work

Our next step is to write and build our method of document reordering based on the understanding and analysis of existing techniques and by using the algorithm outlined in the methodology. One of the possible ways we could try this is to extract the advantages of multiple current techniques and integrate them into our algorithm for better performance. There is still much to uncover in document reordering, either by improving existing algorithms or by innovating a new one.

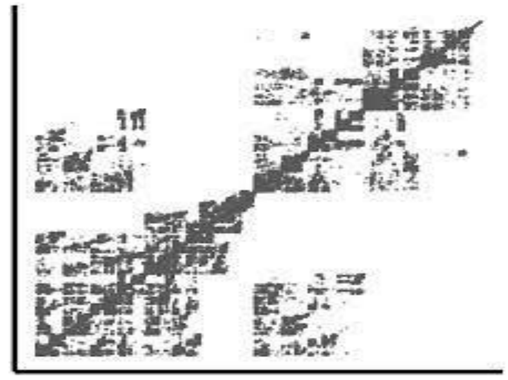

Figure 1: A scatterplot of document reorderings using BP from J. Mackenzie, M. Petri, A. Moffat. Faster Index Reordering with Bipartite Graph Partitioning. SIGIR ’21 pages 1910-1914.

Acknowledgement

The authors thank NYU Tandon School of Engineering’s Office of Undergraduate Academics for generous funding of this project and Torsten Suel for his guidance and mentorship.