Google is built as a large-scale search engine to attempt solving problems of existing systems, such as human maintained indices and automated search engines. With the growth of web, the improvement in hardware capabilities and cost has balanced the rising technical challenges required in web searching. However, the slow disk seek time is still a bottleneck.

Google's aims to get a result list of any search, which is not only complete, but also of high quality. It dedicates to push more development into the academic field, to involve more usage, and to support innovative research activities on large-scale web data.

Google prioritizes its search results by making use of PageRank, which is an objective measure and subjective evaluation of a webpage's citation. A webpage's PageRank takes account of the PageRank of other webpages pointing to it, number of links going out and a damping factor. PageRank is justifiable. Firstly, it could be seen as the probability of a webpage visited by a random surfer. Damping factor can be varied on certain pages to personalize searching of a user. Secondly, pages which are cited by many places are good; and if these places include famous web pages, they are even more valuable.

Besides PageRank, anchor text is also utilized by being associated with the page that the link is on and the page the link points to. This feature helps Google to obtain more accurate details about web pages and search non-text information. One consequence is that some pages, including non-existing ones, which have not been crawled are also returned.

Google also considers proximity between web pages and users, records some visual representation details for better page ranking, and saves raw HTML of pages in a repository.

Historically, search research on the web starts from works on information retrieval, which are done on small well-controlled homogeneous collections. The model developed here does not yield good results on the much bigger web. Also, Documents are varied both internally and in external meta information, forming a heterogeneous web environment. Also on the web there is no control on people's freedom to publish. Some companies use this to manipulate the search engines for profit.

Inside the Google architecture, the distributed crawlers perform web crawling; URLserver sends lists of URLs to crawlers; storeserver compresses and stores the web pages according to docID into a repository, which are then uncompressed, parsed and converted to hits which record word, position in document, font size and capitalization, by the indexer; the hits are allocated into a set of barrels, creating a partially sorted forward index; the indexer parses links in web pages to create an anchors file, which is then read by URLresolver to convert to docIDs; the sorter re-sort the barrels by wordID to generate the inverted index; at last, the searcher uses the lexicon, inverted index and PageRanks to answer queries.

Comment [S1]: Content: A
Organization: A-
Writing: A-

Grading Criteria
Content: Does your summary contain the most important information presented in the paper? Is the content correct, or does it contain technical errors?
Organization: Do you have good ordering of the sections/subsections? Are the sections linked together well or do they appear abruptly? Does the paper flow? Did you motivated/introduce/define items appropriately before using them?
Writing: Did you use words and grammar correctly. Does the writing have a clear and easy to understand style?

Comment [S2]: Not clear what this means

Comment [S3]: Rather long. Probably better to break up the sentence into shorter ones.

There are some data structures used by Google for faster processing of a large document collection. BigFiles are virtual files ranging a few file systems; they deal with allocation and deallocation of file descriptors. Repository stores web pages that are compressed using zlib, a tool with fast compression speed rather than high compression ratio; to ensure data consistency, there is no other data structure besides document packets. Document index keeps all relevant statistics about each document; such compactness is to ensure one disk seek is sufficient for fetching one record; batch mode of update is also utilized for faster speed. Lexicon is implemented as a list of words and a hash table of pointers. Hit lists are better implemented efficiently as they occupy most space in both forward and inverted indices, so a hand optimized compact encoding is chosen for less storage space and less bit manipulation than other methods; hits occurring in a URL, title, anchor text, or meta tag are fancy hits whilst others are plain hits and their formats are carefully designed. Forward index is stored in a number of barrels, each holding a range of wordIDs; wordIDs are stored as a relative difference from the minimum worded in the same barrel to save space. Inverted index maps wordIDs to a doclist of docID's and hit lists which correspond to the word; two different copies of doclists are kept so that quick merging of different doclists and finding answers near the list head for multiple word queries are both possible.

Google has a fast distributed crawling system which major performance stress is DNS lookup. Some problems about crawling include people's ignorance about crawling and its testing difficulty. In the respect of web indexing, Google generates a lexical analyzer for parsing to achieve maximum speed; it also writes a log of all extra words not in base lexicon to ensure parallelization of indexing; sorting occurs one barrel at a time, thus requiring little temporary storage.

For single-word queries, Google examines a document's hit list and compute the count-weights and type-weights to obtain an IR score, which combines with PageRank to rank a document. For multiple-word queries, matched sets of hits are checked and a proximity, which calculates how far apart the hits are in the document, is computed to get the type-prox-weights, which is later combined with count-weights to compute the IR score for final ranking. In addition, a feedback system is implemented to continuously improve the system.

The quality of search result is good. Un-crawled but relevant results are shown due to utilization of anchor text; all links are non-broken due to high PageRank; results are highly relevant because Google places heavy importance on the proximity of word occurrences. Its storage is used efficiently because documents in the repository are well compressed and most queries could be answered by short inverted index. Its system performs efficiently mostly because the indexer is optimized to fasten major operations. Searching performance is improved by caching IO, which spends most time.

Google will further improve itself in the future, by putting more emphasis on search efficiency, research on updates, relevance feedback, user context and result summation.