Parsing MetaMap Files in Hadoop

Amy L. Olex, M.S., Alberto Cano, PhD, Bridget McInnes, PhD
Virginia Commonwealth University, Richmond, Virginia, USA

1 Problem Description

Processing and interpreting text is natural to humans but extraordinarily difficult for machines. Much of how a person interprets texts is based on prior knowledge of concepts and associations between concepts. The UMLS::Association Perl package aids in interpreting biomedical texts by quantifying levels of association between UMLS (Unified Medical Language System) concepts. The UMLS Metathesaurus contains over 2.5 million names representing over 900,000 concepts identified by Concept Unique Identifiers (CUIs). MetaMap parses biomedical and clinical texts and links them to UMLS CUIs, and this output is processed by UMLS::Association to collect CUI bigram frequencies and quantify CUI associations. The collection of CUI bigram frequencies provides a background of knowledge from which to extract associations between biomedical concepts for further processing. However, parsing this information is time consuming, as there are hundreds of large (several gigabytes) MetaMap files for each processed text corpus.

UMLS::Association relies on the collection of CUI bigram frequencies identified by the CUICollector module, which parses the nested MetaMap structure one utterance at a time. As the CUI bigrams are parsed they are stored in a nested hash of hashes that is periodically dumped to a MySQL database. This serial implementation has several limitations: *1)* Perl code is serial and not parallelizable due to Perl's limitations in sharing nested hashes across threads for synchronization. *2)* The bigram hash can become very large, very fast and fills up program memory, which forces the program to periodically write results to a MySQL database, introducing additional latency.

2 Purpose of Project

CUICollectorMapReduce takes advantage of the Hadoop MapReduce framework to overcome serial implementation limits. MapReduce algorithms are well suited to parsing text and counting occurrences. In addition, the Hadoop environment reads and writes all results directly to disk thereby resolving the memory issue. MapReduce was chosen over other distributed processing technologies (e.g. SPARK, Flink) due to system limitations in memory, a large data set size, and the batch nature of the problem that requires accessing the data only once. CUICollectorMapReduce has two modes: *cui* and *article*. In *cui* mode MetaMap output is parsed directly, one utterance at a time, and duplicates the results of the serial implementation. In *article* mode CUICollector concatenates utterances to allow the processing of an entire PubMed citation, which includes collection of CUI bigrams that cross utterances.

3 Results

Evaluation was conducted on the MetaMap 2015 MEDLINE Baseline, a 132GB compressed dataset of biomedical citations. The efficient implementation in Hadoop resulted in a 28x speedup in *cui* mode, reducing run time from 229 to 8 hours on a 4-core, single-node Hadoop system. Analysis of Hadoop output in comparison to that of the serial implementation revealed equivalent results. Required database operations in the serial implementation account for the majority of the serial running time. To more directly compare just the parsing speedup of Hadoop, all database operations were removed from the serial program, resulting in no program output. Hadoop still processed the data faster with a speedup of 2.8x when run on the full MetaMap 2015 MEDLINE Baseline compared to the modified serial implementation.

4 Contributions

The contributions of CUICollectorMapReduce are: *1)* Parallel processing of MetaMap files with seamless scalability. *2)* Eliminating the need to periodically write to a MySQL database by utilizing Hadoop's framework where all intermediate and final results are written and read directly from disk. *3)* Algorithm improvements allow for a full PubMed citation to be processed at once instead of one utterance at a time. In conclusion, implementing CUICollector in the Hadoop MapReduce environment provided significant speedup as well as additional flexibility in the type of data being processed. The Hadoop implementation also allows parsing of larger datasets that were not feasible using the serial module, thus, opening up additional avenues of research.