On string representation for efficient mining and searching over string databse

Metwally Rashad Metwally Omar

One of the most important primitive data types in modern data processing is strings.String data is ubiquitous, and its management has taken on particular importancein the past few years. An important research topic in string processing is stringsimilarity join. A string similarity join finds all similar pairs between two collectionsof strings. It is an essential operation in many applications, such as data integra-tion and cleaning, web page detection, and pattern recognition. Many similarityfunctions have been proposed to quantify the similarity between two strings. In thisthesis, we study string similarity joins with edit distance constraints. There existmany algorithms to support string similarity join with edit distance constraint, suchas All-Pairs-Ed, PPJoin, and ED-Join. Most of these algorithms follows the filterand verification paradigm, where indexes are used to filter many of the unpromisingstring pairs and generate candidate pairs, then these candidates are verified to out-put the final result. Recently, a verification-free, trie-based similarity join frameworkis proposed. It uses a trie structure to index all strings in a given -dataset. Though,this framework has shown many advantages over the filter-and refine framework, themain problem with current trie-based join algorithms is that they generate and main-tain lots of candidate prefixes called active-nodes which need to be further removed.With large edit distance, active-node sets becomes large as well. To overcome thisproblem, they use many pruning techniques to remove false positive prefixes ina subsequent phase. This makes the existing algorithms inefficient for processingvvilarge data sets with long strings, and higher edit distance threshold. In this thesis,we propose a new trie-based join algorithm called PreJoin, which improves uponcurrent trie-based join methods. It efficiently finds all similar string pairs using anew active-node set generation method, and a dynamic preorder traversal of thetrie index. We improve PreJoin in order to be suitable to deal with large edit dis-tance. This improvement called PreJoin-Plus algorithm. Experiments show thatour approaches is highly efficient for processing short as well as long strings, andoutperforms the state-of-the-art trie-based join approaches by a factor of five.