Computer Science 4750
Natural Language Processing
Submitted by: Ryan Martin
Student Number: 201039054
Due Date: Nov. 7/14

Dictionary Data Structures And Computational Time Complexity

The topic I have chosen to present is the sub-discipline Lexicography within the Natural Language Processing area of Computer Science. More specifically, the data structure(s) used for holding dictionaries of lexicons from a natural language. When dealing with such dictionaries like pronounciation or semantic dictionaries within Natural Language Processing, various types of data structures are not only required to implement these dictionaries, but also to efficiently carry out their operations. Both time and space complexity must be taken into consideration when utilizing various data structures for use in Natural Language Processing because each data structure has its own efficient and inefficient operations. Therefore, time complexity and overhead space are a concern given that we regularly have dictionaries that hold massive amounts of data on which we want to perform these operations.

When designing a dictionary to hold a lexicon for a natural language, one must carefully examine all possible data structures and their respective operations for time and space complexity comparison. The article referenced, "Dictionary Data Structures for Smartphone Devices" on the next page portrays the constraints of both time and space complexities with respect to handheld mobile devices. In general, one should choose a data structure with operations that have the smallest time and space complexities; however, the structure type can vary on both the dictionary required and any emphasis on operations the dictionary should perform more efficiently. For instance, one may want retrieval to have the smallest time complexity possible to quickly find a word and output it to the user. To name a few of the data structures that are used in the framework of a dictionary, one is given the options of using a hash table, graph, trie, or a combination of two or more data structures. Alternatively, one may use a combination of multiple data structures to implement a dictionary and hence they will write their own algorithm that will either reduce time or space complexity of an exiting method, while still tailoring the data structures or algorithms to their specific needs.

When contrasting the complexity of each operation of one or more data structure, one should be vigilant in choosing a single data structure as a skeleton for their dictionary. In other words, one must outweigh the positive features with the negative, as decreased time complexity often follows a higher space complexity, and vice versa. This paradigm is well-known to those working in the field of Natural Language Processing and must be taken into careful consideration by those developing software in the area. An instance showing this pattern occurs with a (singly or doubly) linked list and an array list, where the two structures accomplish the same task but are designed differently leading to varying efficiencies. The wasted space of a linked list is lower because there are no empty locations in the list, while the space necessary for an array list is higher due to containing empty indexes with its resize function to handle additional items. A linked list can retrieve an element in O(n) time and an array list can take O(1), portraying the time-space complexity pradigm.

References

1. “A BST-based approach to dictionary structure for Chinese word segmentation.“ Ge, Chang; Ma, Ningjing; Chen, Xudong. **http://ieeexplore.ieee.org/xpls/icp.jsp?arnumber=5953238#citedby-section**.

2. “Compressed data structures: dictionaries and data-aware measures.” Gupta, A.; Wing-Kai Hon; Shah, R.; Vitter, J.S. **http://ieeexplore.ieee.org/xpls/icp.jsp?arnumber=1607256**.

3. “AN IMPLICIT DATA STRUCTURE FOR THE DICTIONARY PROBLEM THAT RUNS IN POLYLOG TIME.” J. Ian Munro. **http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=715937**.

4. “Dictionary Data Structures for Smartphone Devices.” Bentevis, Alexandros; Kerkinos, Ioannis; Kalogeraki, Vana. **http://delivery.acm.org/10.1145/2420000/2413155/a46-bentevis.pdf?ip=134.153.23.138&id=2413155&acc=ACTIVE%20SERVICE&key=FD0067F557510FFB%2EE6F865C24D7DB1C8%2E4D4702B0C3E38B35%2E4D4702B0C3E38B35&CFID=452658846&CFTOKEN=98697682&__acm__=1415337974_f72dff2b7c70117bb2e7c91d3a8a19c0**.