Rohit Mahankali

rm1162

Professor Krzyzanowski

Computer Security

8 April 2019

Ciphers Project Part 3 Write-up

1. My approach to this third part was to use n-gram analysis to crack more complicated cipher inputs. Given that frequency mappings from the known text file to the cipher text file were imperfect, an increased amount of data analysis would be necessary to accurately ascertain character mappings via the frequency analysis paradigm.
2. I chose to use n-grams with a length of 2, or bigrams. I began with this length because it was the logical next step from analyzing frequencies of individual characters. I thought that incorporating these frequencies into the single character frequencies would rectify discrepancies. I played with using trigrams and larger n-grams, but the frequency values started to become so small, that it seemed that given the size of input text files, larger n-grams would not really contribute useful information to our mapping analysis. So the happy medium between simplicity and an n-gram that contributed additional information turned out to be the bigram.
3. N/A
4. My program performed rather poorly on the sample cipher text, and was considerably less accurate in recovering plaintext compared to part two's straight frequency matching. I believe the reason for this is my erroneous logic. In my "SmartCrackCipher" program, I enumerate all possible bigrams from 'aa' to 'zz' for both the cipher text and known text and determine each bigram's frequency. I then sort both the ciphered bigrams and known bigrams and map them by index in their respective list. I then increment the letters that make up each bigram in the known letters list and ciphered letters list by the respective bigram frequency. Finally, I sort the lists of known letters and ciphered letters and map the letters by index to each other. Using these mappings, I attempt to decrypt the input cipher text.

   I believe I do can make this code better and possibly effective. A logical idea that I would have tried next, time-permitting, is to try to make a trie data structure for the ciphered text and known text. This trie data structure would be organized to quickly search for common prefixes, or in other words, organized on a frequency basis. After iterating through the cipher text and known text, for both single-character, bigram analysis and trigram analysis, two tries would be created. I would then map each ciphered character to its twin known character based upon matching node positions in the respective tries. This approach should be sound and crack the cipher text correctly.