**Possible Relations to look into:**

1. Types of sentiments categorized by sender email domain
   1. Does email domain affect what type of emotion the attacker exploits?
      1. i.e. Do .edu emails have a different major sentiment than .gov or other domains?
2. Does sender title have anything to do with the type of emotion the email tries to exploit?
   1. If the sender title is someone of power, do they try to exploit fear more or something like that?

**polarityScoresbyDomain.py**

1. Take in dataset and put information into data structure called rows
2. Go through each entry row and check to see if there is a sender email
3. If yes, then separate the results into different lists in the results dictionary based on the email domain of the sender
4. Tokenize the email content using english.pickle tokenizer from nltk
5. Once results calculated for each row, print out the average compound sentiment value for each domain
6. Output results to csv file

**polarityScoresAll.py**

1. Take in dataset and put information into data structure called rows
2. Go through each email entry and tokenize the email content using english.pickle tokenizer from nltk
3. Once results calculated for each row, print out the average sentiment values for each domain
4. Output results to csv file

**sentimentExtraction.py**

1. Take in dataset and put information into a data structure called rows
2. Clean message text and simplify it down to a list of words that provide some sort of info on the sentiment of the sentence
3. Compare each word to the emotions.txt and see if any of them are in there
4. If they are, add the corresponding emotion to the emotions list array
5. Once each row has been processed, count up all the emotions that have been extracted to see which one is most prevalent

**To Do:**

1. Figure out what tokenizer is best to use.
2. Extract sentiments from email content instead of just polarity scores
3. Use other fields to find relations
   1. Email subject line
   2. Sender title