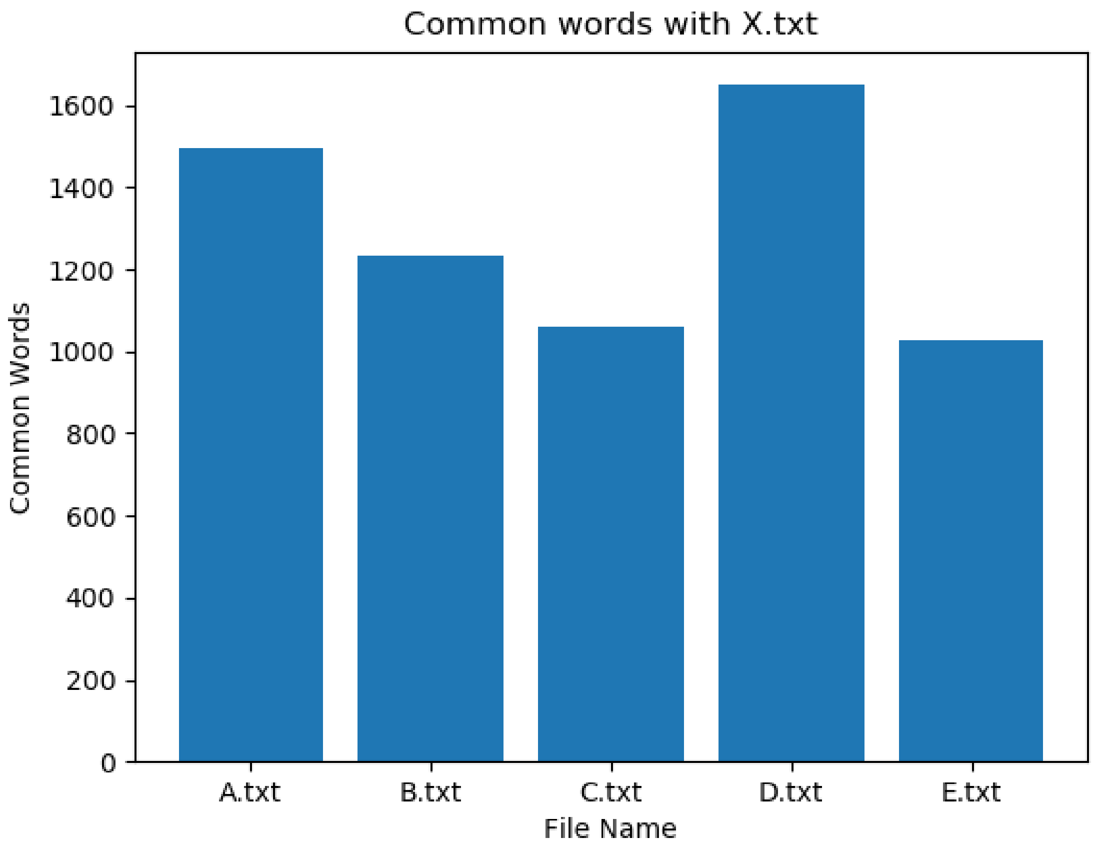
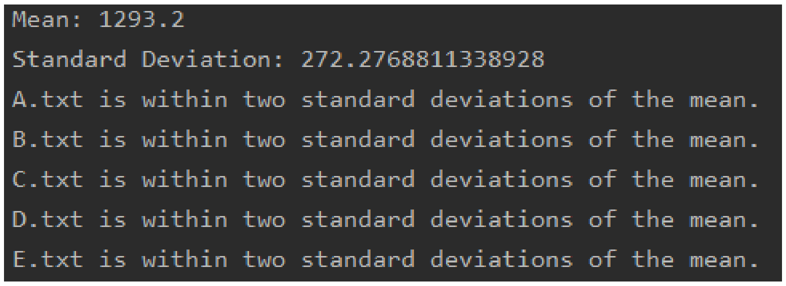
Andrew Breslauer

46671343

# CS 5320 Final

**Q1.C**

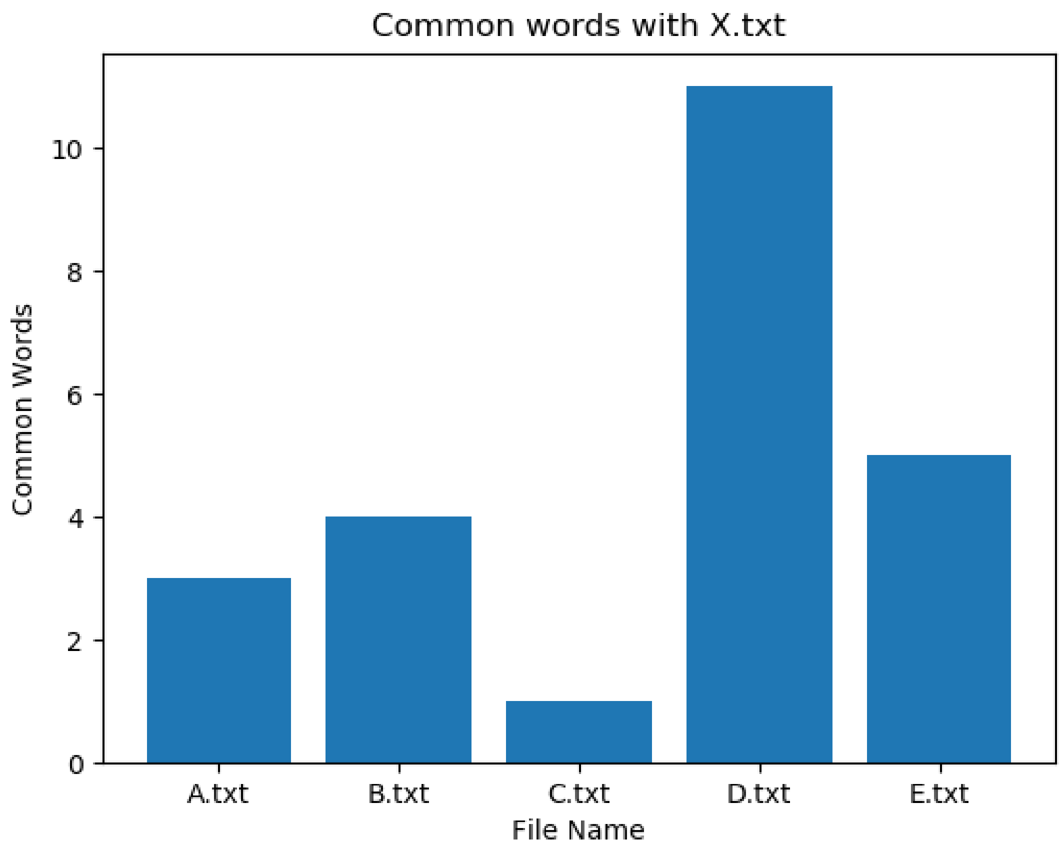


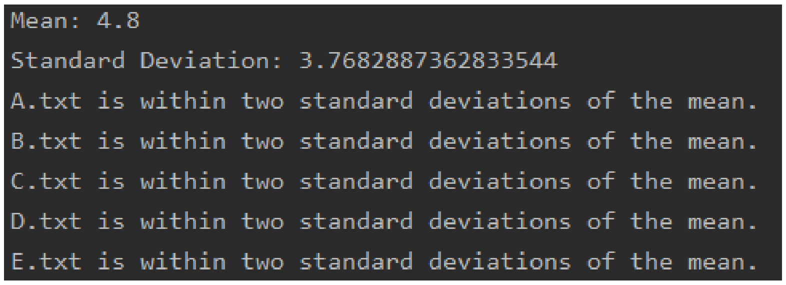


```
Mean: 1293.2
Standard Deviation: 272.2768811338928
A.txt is within two standard deviations of the mean.
B.txt is within two standard deviations of the mean.
C.txt is within two standard deviations of the mean.
D.txt is within two standard deviations of the mean.
E.txt is within two standard deviations of the mean.
```

Based on this first graph, the two most likely contenders are either the author of A.txt or D.txt. This comparison was done with only comparing unique appearances of words, so if an author uses a certain word a lot, it won't matter as much.

**Q1.D**

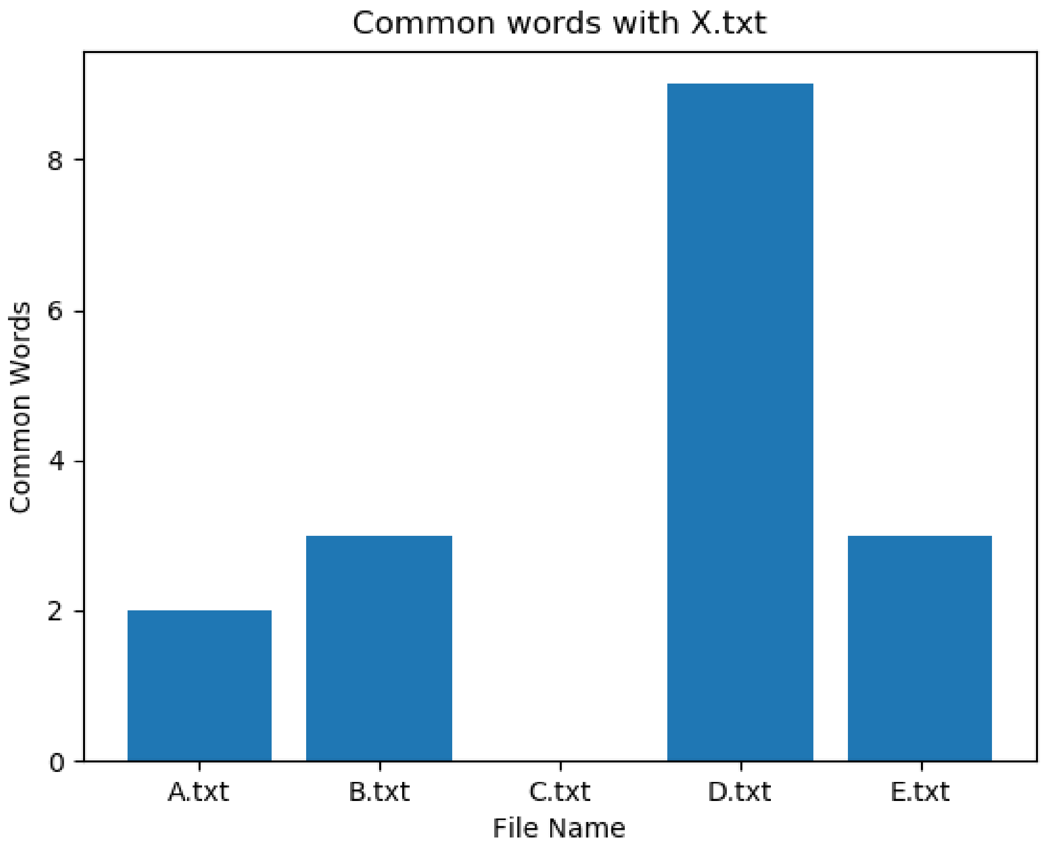


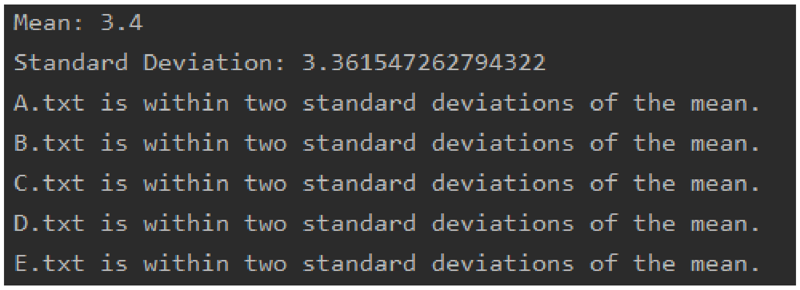


```
Mean: 4.8
Standard Deviation: 3.7682887362833544
A.txt is within two standard deviations of the mean.
B.txt is within two standard deviations of the mean.
C.txt is within two standard deviations of the mean.
D.txt is within two standard deviations of the mean.
E.txt is within two standard deviations of the mean.
```

This graph compares words longer than 12 letters. As we can see, the author of D.txt is much closer to X.txt than the author of A.txt, who were our two candidates from Q1.C. After this graph, the author of D.txt is our leading candidate.

**Q1.E**





```
Mean: 3.4
Standard Deviation: 3.361547262794322
A.txt is within two standard deviations of the mean.
B.txt is within two standard deviations of the mean.
C.txt is within two standard deviations of the mean.
D.txt is within two standard deviations of the mean.
E.txt is within two standard deviations of the mean.
```

This graph compares words that only appear once in each document (one in X.txt and also once in the listed document). As seen above, the author of D.txt continues to distinguish themselves from the other authors.

**Q1 Conclusion**

In all 3 graphs, but especially in graphs D and E, the author of D.txt is much closer to the unknown author of X.txt than any of the others.

**Q2:**

See attached final_results.txt, and then come back for explanation.

I don't know why the sparql query returned no results, when there is clearly at least Jax who matches all of the descriptions. Jax knows Cat, and Rollo knows Cat. Jax lives in California. Jax dislikes the Lakers, but likes Titanic. Those are all of the qualifications the query asked for and Jax matches all of them, but the query isn't recognizing him. I was also having some problems getting semweb to recognize my ontology file, as it kept converting SymmetricProperty into SubProperty (i.e., schema:knows is a subproperty of schema:knows rather than being Symmetric) among other issues, so I eventually gave up and built the ontology rules into the code. I still have the ontology file, so you can look through that and see if anything went wrong that wasn't supposed to. Apart from that, I made some minor edits to the slang file (i.e., the comma separated file had "pits, the" as a singular word but it was being recognized as "pits" and " the" separately, so I removed that completely because it isn't used). If you have any other questions about decisions I made, feel free to e-mail me, or just take the points off.

Specifics:

**Q2A:**

I found all possible phrases that could mean "to know", and regex replaced them within the sentences.

**Q2B:**

Followed the same pattern as assignments 11 and 12 from class. Join together the original sentence and the NLTK.pos_tag version of the sentence, then look for the correct patterns. Once those are recognized, pull them out and mark them as the appropriate category (people, work, location), and then add them to the n3 file.

**Q2C:**

See ontology file.

**Q2D:**

Included as part of the final_results.txt file.

**Q2E:**

Created lists of positive and negative adjectives, then regex replaced all instances of both lists with "good" or "bad".

**Q2F:**

Found NNP V..? J..? patterns with regex (noun verb adjective), then located those in the sentences and isolated the speaker, target, and sentiment. I didn't use the e3code.py because by replacing them with "good" or "bad" in Q2E, I already had my sentiment.

**Q2G:**

Added to original n3 file.

**Q2H:**

I wasn't successful in either of the suggested options, because all non-coding-language based ontology examples didn't even get as complex as this and I couldn't figure out how to operate the commands. In the end, I just incorporated the intention of these commands into the original n3 file creation part (Q2G).

**Q2I**:

Even after all the effort I went through to operate around the owl files, the sparql query still didn't work. However, I can't see where it's going wrong, and it isn't giving any errors, so in the end I'm just going to turn in recognizing that it isn't returning the right things still.