Progress to date:

------------------------------------------

Most of my work so far has been in two areas: reading related literature and building a baseline implementation of Word2Vec. My research into the literature at the beginning mostly focused on high-level questions about task feasibility, including about the success of word embeddings for authorship attribution tasks. Recently, it has been more methodological, focused on papers that describe methods for the design of loss functions for class-specific training (like the bag-of-embeddings model) and methods for analyzing and visualizing sets of word vectors. The more tangible product of my work is that the baseline implementation of the skip-gram network that I'll use to train my embeddings is working. I have been trying it on example documents, which has been successful but has driven home the need for useful, simple tools for visualization and analysis of the resulting word vectors.

Current difficulties:

------------------------------------------

The biggest concern that I have right now is that the training time for the baseline implementation is still very high. The model takes a while to create meaningful word embeddings from even a single document, which bodes poorly for its ability to quickly categorize novel documents. I'm sure that all computational times involved are, strictly speaking, feasible, but whether the proposed use of word embeddings for predictive tasks is useful at all depends partially on whether its computational tradeoff is offset by increased performance, which is still very much an open question. Consequently, I'm investigating methods to speed up training without losing accuracy, and hoping that some of these optimization strategies will be effective at increasing the speed.

Next steps:

------------------------------------------

One of several next steps will be obtaining the text data and using it to train the baseline model to create the class-specific word embeddings. If I haven't heard back about funding soon, I can begin with online books, which can be sorted by author to create author-specific classes for an attribution task (which is well-supported by the literature). Training multiple vectors efficiently may require me to implement some of my planned optimizations now, rather than later, and once I create a full training set I will experiment with parameter tweaking, like minimizing the vocabulary size. This requires me to have a way to benchmark model performance, which is the next and largest step. I want to begin building the testing and visualization tools that I'll need, which require more reading about the best methods for evaluating task performance and for visualizing high-dimensional data. I would like to create a tool that would allow me to visualize the difference between two different embeddings for the same word between classes, as well as implementing some strategies for discovering high-level patterns in the data.