**Tim Zhang (110746199)**

**CSE519 Progress Report: Dating Documents**

**Overview**

Since my initial proposal I have implemented an evaluation environment and a variety of models for the document dating task. Using the Corpus of Historical American English (COHA) dataset, kindly provided by Yingtao Tian, I found my implementations to compare favorably with the previous work done by your students. It should be noted that I did not replicate the previous work explicitly. From what information was provided in the draft paper, the proposed models were trained using Google Book Ngram data (presumably the 5-gram data). However, I trained my algorithms with the COHA corpus itself. I also experimented with learning algorithms that were not considered in the previous work.

I found that an SVM classifier trained on documents represented using unigram bag of words had a mean absolute error of 6.13 years. This MAE score is quite an improvement over the results provided in the paper compared against all models investigated on the COHA dataset. Furthermore, I did not tune any hyperparameters for SVM, which implies that my results may be improved substantially. I also experimented with Random Forest and Naïve Bayes classifiers using unigram BoW and tested the models using higher order n-gram features.

**Next Steps**

While my initial results are very promising, there is quite a bit of experimentation and extensions which I intend to implement before my final submission.

**Datasets:** The top objective is to parse and incorporate more datasets such as Gutenberg and New York Times. I am in the process of scraping the NYT using a provided API. Unfortunately, this API is throttled to 1000 requests a day, so I have been forced to religiously scrape 5 years a day (at 200 requests per year) for the past month. I have calculated that I will have the required data before the deadline is due, and I am currently at year 1900 (moving backwards).

I have already obtained a subset of the Gutenberg dataset. The problem is that there is no uniform way to parse the publication year using regular expressions or by any other means. As such, I implemented a script which I can interface with to manually enter the publication date for the 2000+ documents. This is very tedious but I am about 50% complete with this task.

In addition to processing more corpora, I will need to carefully analyze the relative distributions of the total dataset to keep the data balanced with respect to decade distributions and word counts. The intent is to balance the frequency of decade labels which are imbalanced in COHA as well as increase the volume of my data.

**Models:** Although SVM has proved to be very effective at learning in this space, I believe I can do better. The first thing which I will implement is hyperparameter tuning via grid search. After I have a working automatic optimizer I will then incorporate learning curves which will provide intuition on how well my models are fitting to the data (bias/variance decomposition, under/overfitting). Using this information, I will have a feedback mechanism for if I need to incorporate more data to fit my models. I also will implement cross validation for more accurate performance results.

As a stretch goal, I would like to experiment with neural language models such as Word2vec and training a Convolutional Neural Network over the data. When applied to NLP tasks a CNN implements an n-gram processor. Unfortunately, I am not entirely confident that I have enough processing power to train such a network. This is of high interest for me as I have never deployed a neural network from scratch before.

**Implementations**

**Project Repository:** <https://github.com/moduio/CSE519-2017-110746199/tree/master/Project>

Currently I have implemented a Jupyter notebook (**project.ipynb**) for my preliminary experiments on the COHA dataset. Additionally, I have implemented a web scraper for the New York Times (**NYT\_scraper.py**) and an interface for providing the publication dates for Gutenberg files (**Gutenberg\_parser.py**). Both data harvesting interfaces are straightforward, with the resulting files saved as text files prepended with the year label for ease of future integration.

I will briefly explain what I have implemented in the main project notebook, with selected results presented in the following section. My first task was to parse the COHA files into a Pandas DataFrame. After having stored and cleaned the data, I performed exploratory analysis over the word distributions and frequency of the year labels for each topic source (fiction, nonfiction, news, magazines). Next I merged the sources and did an aggregate statistical analysis over the entire dataset at the decade label level.

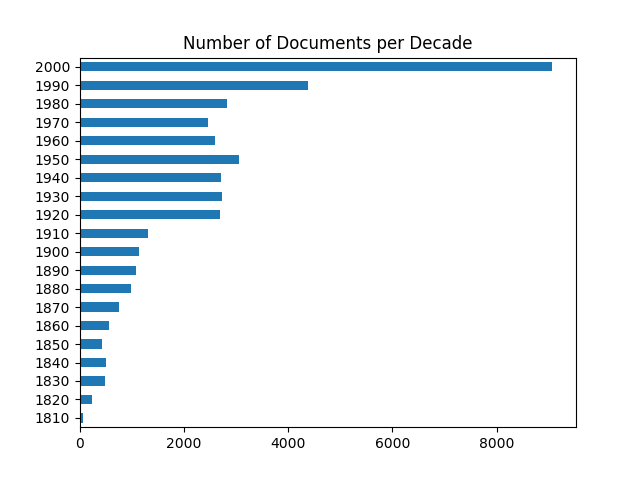
I removed topic level explorations from the current notebook to save on visual clutter, however this analysis was useful for my understanding of the dataset. From this data exploration phase I realized that the corpus was heavily imbalanced with respect to label distribution. However, this problem seems to be partially mitigated as the total token count over each decade partition is relatively stable. I also realized that the year resolution had relatively poor coverage guarantees, with some years having no text examples at all.

Next I implemented an evaluation environment function. This function takes as input the ground truth labels and the predicted labels from an arbitrary learning algorithm. Using these arguments, a confusion matrix is generated along with various statistics such as F1, recall, accuracy, MAE, etc. I used my evaluation environment to test all models and visualize my experiments.

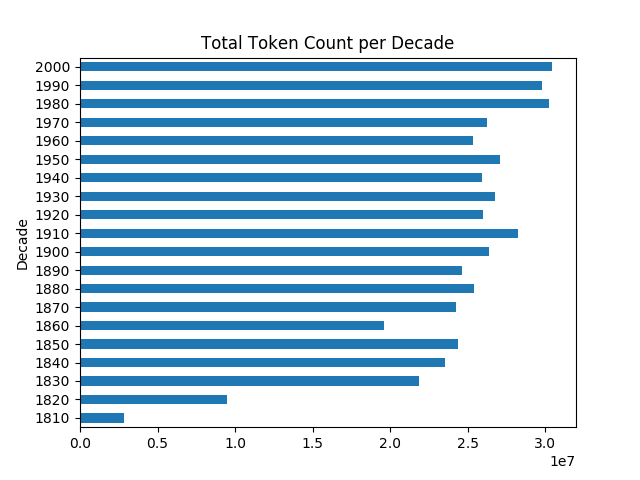
Finally, I implemented my learning algorithms by processing the corpus into BoW vectors normalized using TF-IDF. As mentioned before, I trained Naïve Bayes, Random Forest, and SVM models over the unigram BoW features. I also experimented with higher order Markov processes (up to trigrams), which showed slight improvements. The increased processing time for the higher order processes makes me hesitant to incorporate them into my final model due to the relatively minimal gains. It may turn out that these higher order features work better when hyperparameters are sufficiently tuned however.

**Results**

All model results are given on an 80/20 train/test split using unigram BoW features normalized with TF-IDF. Please see the project notebook for more detailed and complete results including higher order n-gram features and more detailed exploratory data analysis.



Here we see that the COHA dataset is imbalanced with the majority of the corpus constructed with recent texts.



However, the token frequency per decade is relatively balanced. This indicates that the text sources from the various decades are different in nature. Works from the 1800s are likely books or longer texts, while recent data samples are likely to be shorter documents such as news articles.

**The Monkey:**

Accuracy: 0.0515 / MAE: 71.4225

Precision Recall F1-score Samples

0.10 0.05 0.06 8000

The “monkey” does quite poorly with uniform random predictions as expected. We also see that this model has a noisy MAE score.

**Naïve Bayes:**

Accuracy: 0.549375 / MAE: 11.5375

Precision Recall F1-score Samples

0.55 0.55 0.54 8000

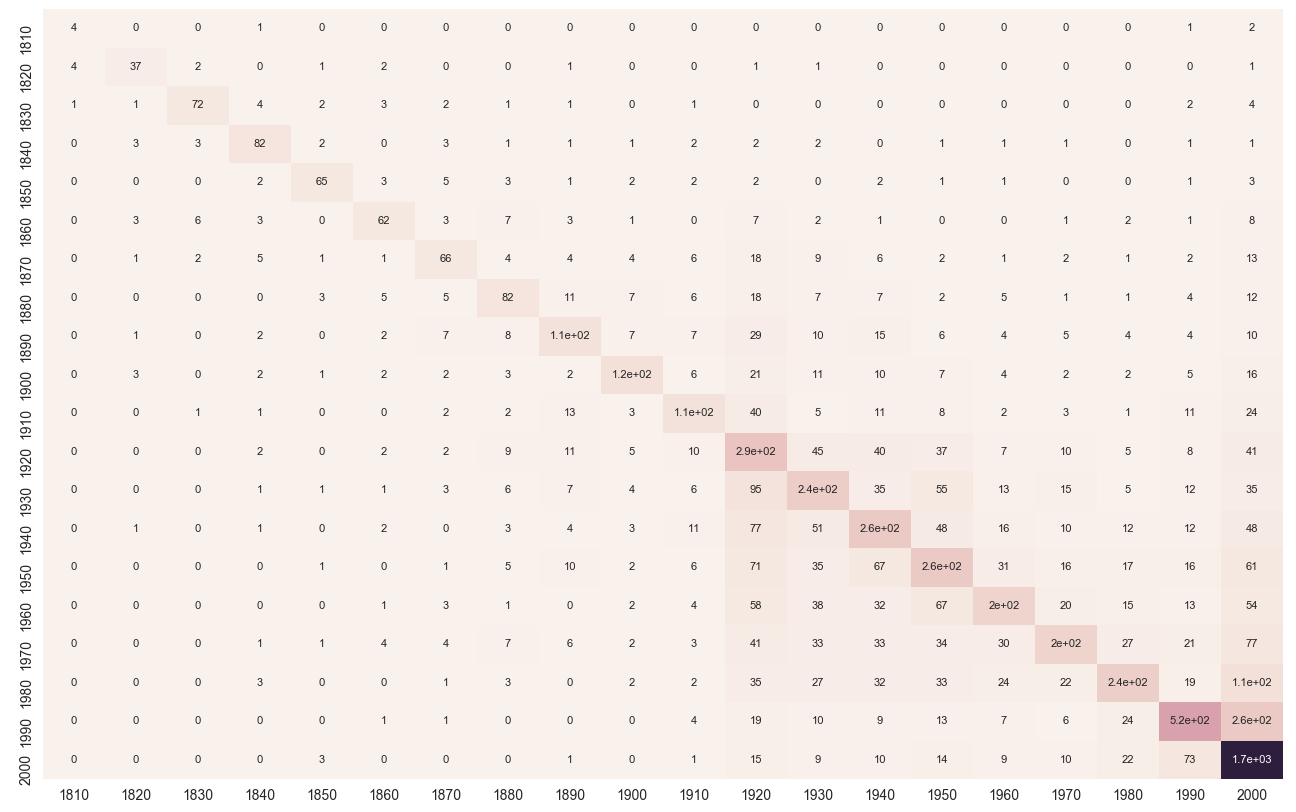
Naïve Bayes does not perform very impressively but has true predictive power and serves as a good baseline model. We see that the predictions are approximately within a single decade on average, which is a much tighter range than uniform random guessing provides.

**Random Forest:**

Accuracy: 0.588125 / MAE: 13.81

Precision Recall F1-score Samples

0.60 0.59 0.58 8000



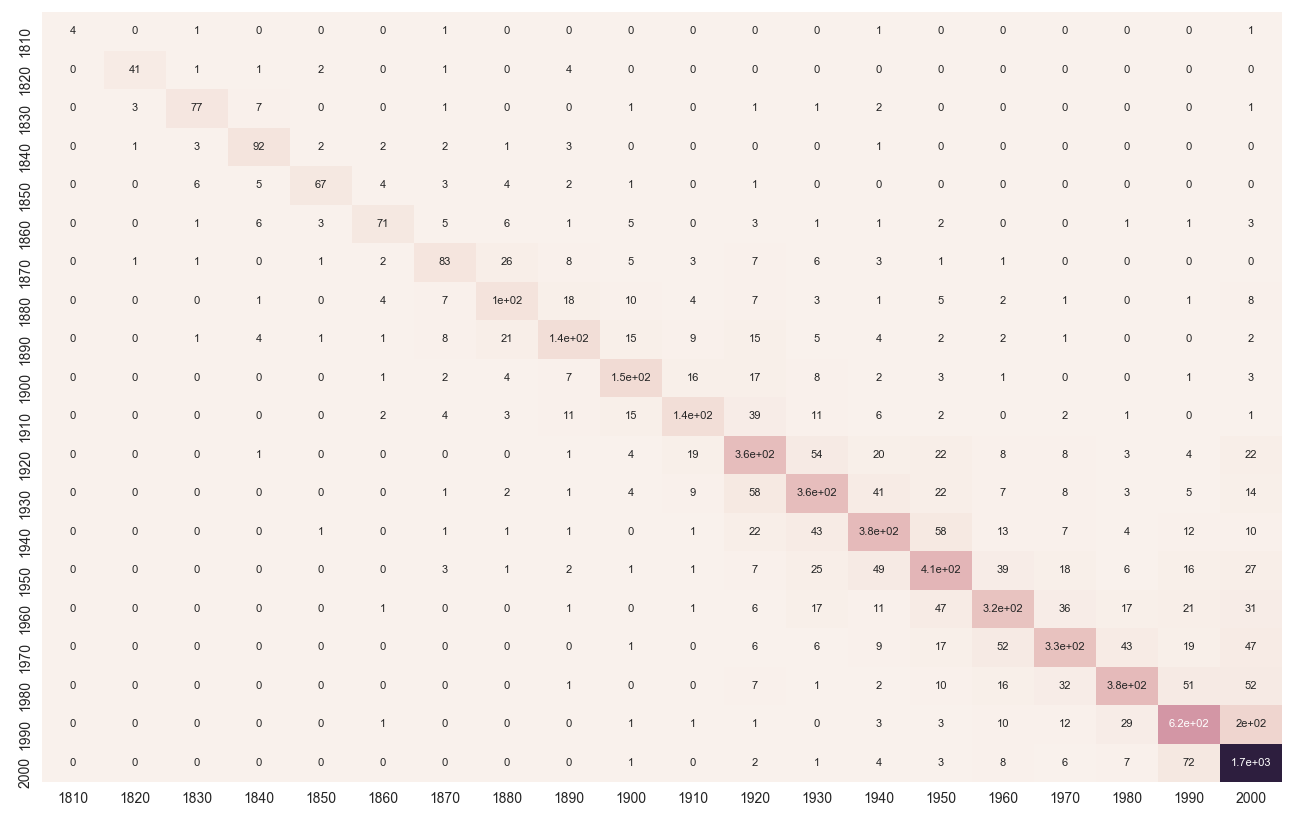
Random Forest does slightly better than Naïve Bayes with respect to the precision metrics but has a worse MAE score on this test split. We see that the classifier is somewhat noisy, with 2000 being a popular classification. There is also noise along the 1920-1950 range.

**SVM:**

Accuracy: 0.73325 / MAE: 6.13

Precision Recall F1-score Samples

0.73 0.73 0.73 8000



SVM performs significantly better than any of the other models with a MAE of approximately 6 years on this test split. Additionally, we notice that the predictions in the confusion matrix are less noisy than for Random Forest and Naïve Bayes. Most prediction mass falls within a tight region near the main diagonal. These results are promising and model tuning may yield even stronger performance.

**Conclusion**

Now that most of the heavy lifting has been implemented I will focus on optimizing my learning algorithms in an intelligent manner using learning curves, grid search, and cross validation. Integrating additional corpora should be relatively trivial since I have already implemented the logic to handle parsing to the correct DataFrame representation.

I believe that SVM can be improved and I also think that Random Forests have a lot of untapped potential. Additionally, I am interested in Softmax Regression and Gradient Boosted Trees as candidate learning algorithms. As mentioned earlier I will attempt a deep learning solution as well. I am very interested to see if a deep neural network performs better than these classic learning algorithms.