“marGINAlly optimized”
Author Attribution — CIS 530 Final Project

Vidur S. Bhatnagar
MSE Robotics,
University of Pennsyvania,
vidurb@seas.upenn.edu

Prerna Srivastava
MSE Computer and Information Science,
University of Pennsyvania,
prernasr@seas.upenn.edu

***Abstract*—The goal of this project is to attribute the authorship of excerpts from a set of New York Times articles to author Gina Kolata. The training data included labeled excerpts from New York Times as Gina Kolata vs other authors. We extracted and tried a multitude of features, setting the baseline as an SVM on 1000 most frequent words and then moving onto more complex features like Parts-of-Speech, Synsets, etc.**

I. Introduction

Authorship attribution has had a long history and has traditionally been used for applications like attributing anonymous or disputed literary works to known authors and plagiarism detection. Since the advent of Internet, a plethora of electronic text in the form of user generated content like blogs, articles, tweets, states has been produced. This has opened up many more avenues for the applications of authorship attribution like attribution of messages or proclamations to known terrorists on social networks and identifying the authors of source code of malicious software. One of the most recent cases where authorship attribution came to the fore was identification of J.K Rowling as the author for Cuckoo’s Calling which was written by her under the pseudonym Robert Galbraith.

The main motivation behind our approach was to evaluate different syntactic and semantic features that would help most in discriminating the writing style of Gina Kolata from other authors. To begin with, we used the bag of words hypotheses from [1] to represent the excerpts in terms of the words used. We employed various techniques to select the words that would best represent the excerpts so as to exploit the differences between the writing styles of Gina from other authors. Then we used this representation to train SVM and logitBoost classifiers which would be able to learn the differences encoded in the features and correctly predict the excerpts written by Gina.

We tried a lot of concepts introduced in the course of study, such as counting the basic frequency of tokens, type-token ratio, tf-idf and chi-squared statics to rank the words. One of our main focus areas was to reduce sparsity in the vector representations so as to generate better statistics. We achieved improved counts by collapsing words into clusters using stemming, principal component analysis, part-of-speech tagging. We also tried extracting further features from the excerpts based on word and sentence statistics and using them in tandem with bag of words to train the models. An obvious approach was to train language models, specifically the bigram language model in the hope to tap the syntactic structural differences between the writings of Gina Kolata and the other authors.

A lot of our implementations followed from existing work or concepts prevalent in Machine Learning and/or Natural Language Processing domain. However, most surprisingly, some of the methods gave very poor performance on cross-validation. For e.g. a standard approach of feature reduction is to select top-k scores from Princinple Component Analysis (PCA) on the raw data. Yet, an SVM trained on PCA’d data drastically dropped in performance for our feature representation. Likewise, another startling discovery was the fact that standardization of differently scaled features led to poorer performance.

The report is organized as follows - Section II mentions the methods and the and the various features that we used to train the model. It also describes feature selection process, the improvements that we made and the tools that we used. Section III talks about the system that we used for our final leader-board submission. In Section IV we talk about the other experiments that we did and results that we achieved. Section V is a discussion about what we learned and how we can extend the system to achieve better performance.

II. Method

We experimented with four sets of features for the classification task. We employed cross validation to test the accuracies of our models on the training set. We started with holding out 20% of the training data (which was used as a dummy true test) and then ran a 10-fold cross validation for building the models on the 80% held-in data. Our training accuracies very closely followed the true test accuracies achieved on the leaderboard.

A. Features

1) Feature Selection/Ranking for Words: For our bag-of-words approach, our aim was to bring down the number of features from 40,000 odd unique words to a more computationally acceptable number. To this effect, we used several word ranking techniques for selecting the top words in our excerpt representation, like:

- Raw Token Frequencies
  Sum of word counts per observation
- Raw Token Frequencies with Stemming
  Stemming was achieved with the Porter Stemmer, and counts of words with same morphological form were added/collapsed together
- Relative Token Frequencies
  Sum of word counts per observation divided by total words in that observation
- Tf-Idf
  Term Frequency was calculated per observation and multiplied by Inverse Document Frequency calculated for the entire corpus
- Chi-Squared Statistics
  Topical words were ranked for 2 corpora - one for Gina Kolata and other for the rest of the authors. A hard threshold was set on Chi-square statistic ¿ 10 for an entry to be considered. Here we selected words based on 3 rules. First, words present in both corpora, second words present only in corpus representing Gina Kolata's articles and third, words present only in the other corpus. Chi-squared statistic was used to rank the features and we got a total of 4640 features using this method.

2) Post-processing the Counts: The different weights in our document representation - raw frequencies, relative frequencies, tf-idf, etc. were (L2) normalized and/or standardized, when we combined these with other derived features. To reduce sparsity in our feature vectors, we clustered the words into various classes and collapsed the member counts into a single sum, which helped in reducing data sparsity (as most words occur only once in the corpus as per Zipf's law). The following methods were used for clustering the words

- Stemming
  Porter Stemmer was used to reduce words to their stems and count of words with same roots were collapsed together.
- WordNet Synsets
  We used WordNet synsets to improve our counts by collapsing synonyms in our representation

3) Derived Features: We also extracted some word level [3] and sentence level features and used them in our models:

- Parts-of-Speech (POS)
  Created 12 new features by using the universal tag set of 12 POS (ADJ, ADP, ADV, CONJ, DET, NOUN, NUM, PRT, PRON, VERB, "." , X)
- Total number of characters (C)
- Total number of words (N)
- Average length per word (in characters)
- Vocabulary richness (total different words/N)
- Average Sentence Length
- Sentence Length Deviation

4) Bigram Features: We derived the following bigram features from the excerpts

- Bigrams of words
- Bigrams of Universal POS Tags
- Bigrams of function words and Universal POS Tags for all other word categories (ADJ,NOUN, NUM,VERB)

B. Other Improvements

SVM with a Linear Kernal was our primary classifier as it has shown promise in a plethore of natural language processing tasks. Several variations on SVM including Gaussian and Intersection kernels were tried. Along with these, AdaBoost, LogitBoost and Linear Discriminant Analysis were evaluated as additional classification models. We tried using PCA on all the unique words in the training corpus for dimensionality reduction as well. Another technique used for feature reduction and selection was to rank the features on different metrics and then selecting a small subset of the top-ranked features. For getting the word ranking we generally used the whole training corpus. For getting topical words for Gina Kolata, we used Chi-squared statistics on excerpts written by Gina and others separately and then extracted 3 sets of features from Chi-squared statistics as mentioned above.

C. Resources and Tools

The text pre-processing like collecting top words and bigrams for each excerpt was done in python. The word features and the sentence features were derived in Matlab and the classifiers were run and models created in matlab as well. Porter Stemmer was made use of in Matlab and python to get stemmed feature for words for the purpose of reducing sparsity in the document representation. The POS tagger interface provided by NLTK was incorporated to collect pos tags for words in the corpus. For getting the different synsets/synonyms for tokens to boost the token counts, WordNet interface provided by NLTK was used. Srilm was made use of extensively for getting bigrams language models.

III. Final System

Our final system is an ensemble of multiple classifiers. Ensemble methods work best when each model in the set is trained on disparate features, so as to learn substantial variance from the dataset. We used a weighted vote in the ensemble so that the mistakes of an individual classifier could be corrected by other classifiers in the ensemble, depending on the individual confidence of the classifier. An ensemble approach boosted our accuracy by roughly 2%. We used the following classifiers in the system

- SVM with top 2000 words, ranked by raw frequencies
- SVM with top 5000 words stemmed and ranked by raw frequencies, along with derived word and sentence features
- SVM with top 10,000 bigram features, ranked by raw frequencies
- SVM with top 4640 Chi-square ranked features
- Logitboost with top 4640 Chi-square ranked features

TABLE I
LEADERBOARD SUBMISSION DETAILS

	Train Accuracy	**Test Accuracy**
Final Submission	92.71%	92.94%
Attempt #1	88.93%	89.61%
Attempt #2	89.22%	89.10%
Attempt #3	91.19%	91.15%
Attempt #4	82.49%	81.12%
Attempt #5	92.01%	92.71%

IV. EXPERIMENTS

The following table I shows the accuracies of our final system and the models that we had used for previous submissions.

We carried out a host of experiments with the word features that did not get included in the final system. The entire arduous journey is documented here (https://goo.gl/VwGxTT).

- The foremost experiments we did were with the number of words that we wanted to include in the excerpt representation. 3000-5000 word features proved to be an ideal number for feature vector representations as it gave us the best accuracies.
- Stemming for words was used to collapse words (and their counts) and reduce sparsity in the representation. This resulted in a decrease in the overall accuracy by a small percentage. A possible explanation of the same is that we were losing covariance due to collapsed counts.
- Using word bigram models did not help in getting a higher accuracy either. The resulting feature vector was extremely sparse and the results we got were approximately the same as compared to using unigrams.
- A bigram model of POS tags was implemented in the hope that it would be would be able to capture some syntactic representation in the sentences. However, this resulted in a very low accuracy because the span of bigrams for POS is too small to capture any syntactic differences and the construction is approximately the same for different authors. We also tried using a mix of function words and POS tags for (NOUN,NUM,ADJ,VERB) as specified in [2] but in vain.
- As suggested in [2], normalization using relative frequencies and L2 norm of feature vectors per excerpts was tried. This resulted in an extremely low accuracy of 79% to our surprise.
- WordNet was used to derive synsets in order to generate synonyms for words, again with an aim to reduce sparsity. This representation provided a modest gain in the accuracy.
- While using combination of raw word counts and derived features, standardization was performed to account for different scales. Contrary to our beliefs, this again led to inferior performance.

V. DISCUSSION

Figures 1 and 2 are wordcloud representations of the topical words used in Gina Kolata's corpus and other authors' corpus respectively. Figure 3 is a treemap visualization to show the final ranking derived from the Chi-square statistics that were used as a feature-set for one of our models. Please note, this is an interesting way to show a Zipfian Distribution of words, rather than the usual line-chart.

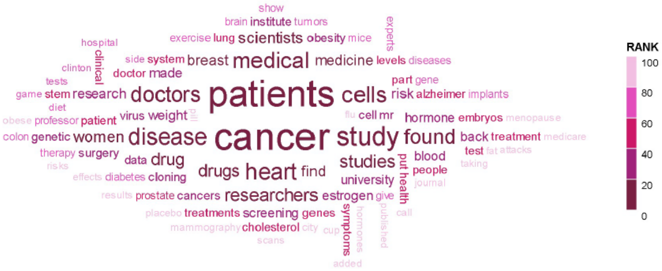

Fig. 1. Chi-squared Ranked Words for Gina Kolata

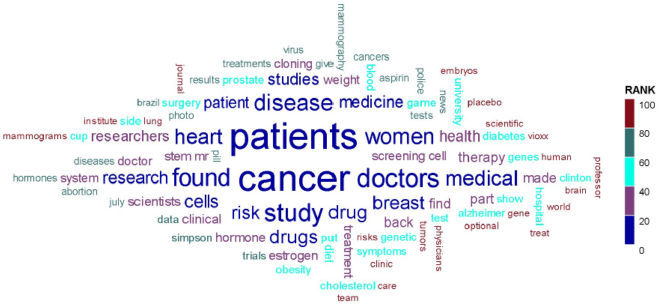

Fig. 2. Chi-squared Ranked Words for Other Authors

This project proved to be good avenue for us to apply the assortment of natural language approaches we were taught in class to solve a real world problem. This not only helped in cementing our understanding of the material but also apprised us of the shortcomings of some of these in practical applications. One of the major takeaways for us was that it was easy to achieve a reasonable accuracy by applying a fairly simple solution. To register any further substantial growth in the accuracy requires the use of clever ways to incorporate features and combine models.

The bag of words hypothesis model proved to be a reliable starting point and it remained a mainstay to the extent that all our models were built around it. The feature selection employed worked fairly well and helped reduce the training time by a substantial amount. The SVM and AdaBoost classifiers used traditionally in natural language processing performed the best out of the many models tried as expected.

Having trained several models, it was quite a non-trivial task to efficiently combine the models into a single model. Ensemble techniques have proven to work very well in such conditions as they work as error-correction systems. None of our individual models was able to reach an accuracy of 92%, however, the ensemble of different models pumped our overall leaderboard accuracy to 92.94%. This is due to the fact that ensembling reduces the generalization error of the models and

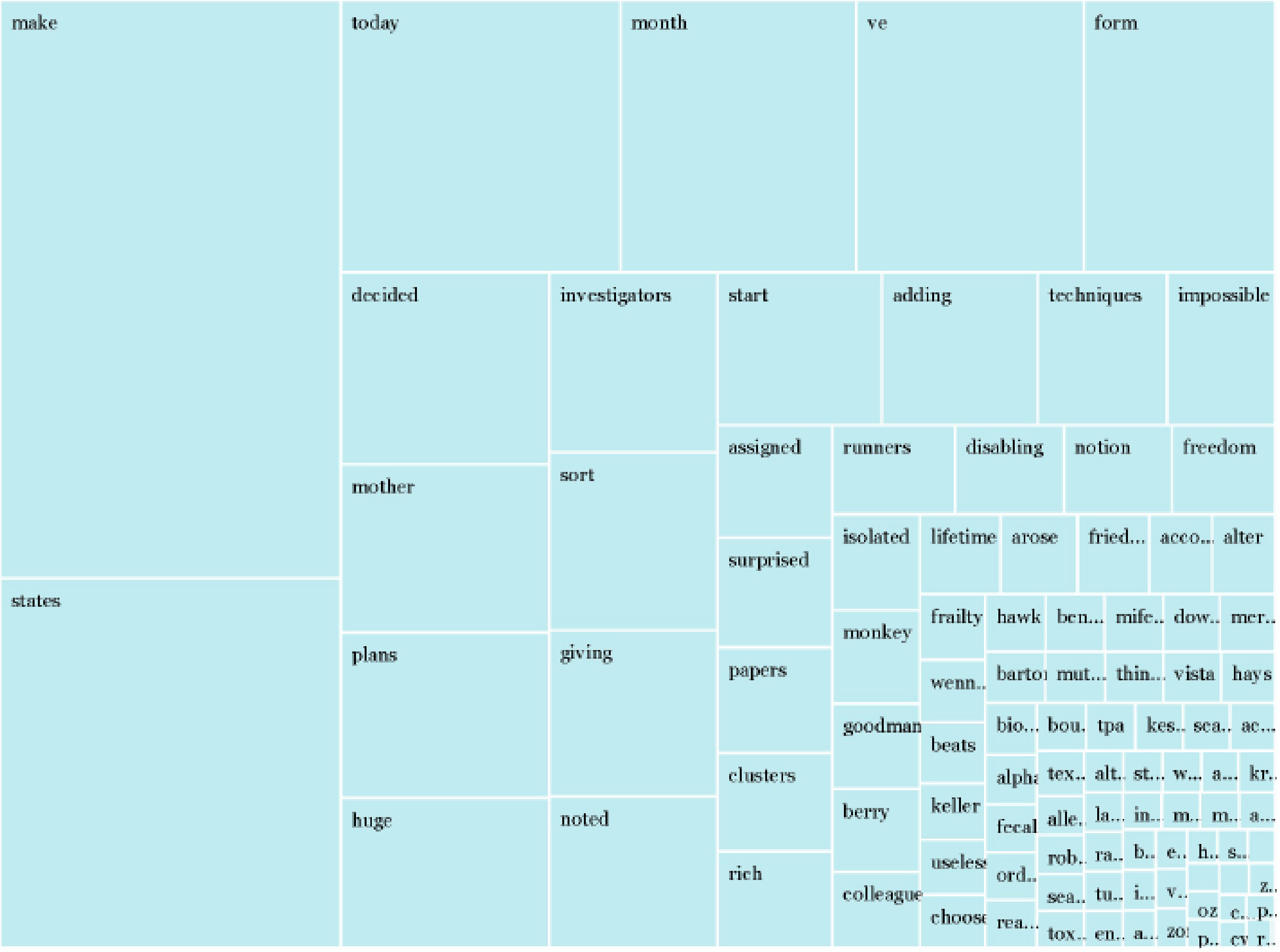

Fig. 3. Top Ranked Words for the Entire Corpus

helps achieve a better true test accuracy.

A startling observation that came to the fore during our experimentation was that standard procedures for dimensionality reduction such as Principal Component Analysis and sparsity reduction such as stemming and other forms of clustering did not perform well in practice. They did not help in achieving superior accuracy as compared to classifiers tried.

Future Scope

A method we really wished to, but couldn't explore due to time constraints was that of K-ee subtrees [4]. The implementation of K-ee syntactic subtrees, although highly complex, shows promising results.

Acknowledgment

The authors would like to thank Prof. Ani Nenkova and TA, Anne Cocos, for their constant guidance through the course.

References

[1] Turney and P. Pantel, *Frequency to Meaning: Vector Space Models of Semantics*, 2010

[2] diederich et al., *Authorship Attribution with Support Vector Machines*

[3] Athanasios Kokkos, and Theodoros Tzouramanis, textitA robust gender inference model for online social networks and its application to LinkedIn and Twitter, Vol. 19, No. 9, 2014.

[4] Sangkyum Kim, *Authorship Classification: A Discriminative Syntactic Tree Mining Approach*,2011

[5] Statamatos,*A survey of modern authorship attribution methods*,2009

[6] Segarra et al.,*Authorship attribution through function word adjacency networks*,2014