"Honey, We Shrunk the Weights" Gender Prediction using Twitter Feeds and Profile Images

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

Nitin J. Sanket
MSE Robotics,
University of Pennsyvania,
nitinsan@seas.upenn.edu

Sarath Kumar Barathi
MSE Robotics,
University of Pennsyvania,
sarathba@seas.upenn.edu

***Abstract*—The goal of this project is to predict the gender of a person using the user's twitter data which was extracted using Twitter API. The data included tweets of a particular person, profile picture and other derived features namely, age, smile, orientation of face, type of glasses worn, and how much percentage of the picture is occupied by the person's face. Using this data, our team built several models which were finally ensembled to create a model with fairly high accuracy to predict the gender, which helped us in topping the test-set leaderboard. However, we could not replicate our performance on the validation set due to size and time constraints.**

I. Introduction

The rapid growth of social networks has produced an unprecedented amount of user-generated data, which provides an excellent opportunity for text mining. Authorship analysis, an important part of text mining, attempts to learn about the author of the text through subtle variations in the writing styles that occur between gender, age and social groups. Such information has a variety of applications including advertising and law enforcement. This project aims to identify the gender of a person using the person's twitter data.

The report is organized as follows - Section II mentions the methods used for feature selection and the various derived features that we used to train the model. Section III talks about the process of model selection and the methodology used to combine the various trained models. Section IV discusses about some of the important results we learnt from this project and future scope. Section V talks about the results and we conclude the project in Section VI.

II. Features Selection and Feature Extraction

The most important features were words and their counts, which gave us the highest accuracy. This is what is commonly known as the bag-of-words model.

A. Feature Selection on Words

As suggested in [1], we employed multiple feature selection techniques, namely, Chi-Square, Information Gain, Information Gain Ratio, Symmetrical Uncertainty, and Filtered Attribute Evaluation, to rank our features [2]. After deriving individual ranks from all of these methods, we created an average rank for every word. This helped in reducing our feature space and improved training time massively. Mostly, all of our models ran on Top 1000-2000 ranked words. A tree map of the ranked words by counts can be seen in Fig. 1.

B. Feature extraction from Words

We derived the following features from the raw word frequencies in the dataset [3]:

1) Parts-of-Speech (POS) Counts Created 12 new features by using the universal tag set of 12 POS (ADJ, ADP, ADV, CONJ, DET, NOUN, NUM, PRT, PRON, VERB, "." , X)
2) Total number of characters (C)
3) Total number of words (N)
4) Average length per word (in characters)
5) Vocabulary richness (total different words/N)
6) Word Stemming to improve data density - For this purpose, we employed the Porter Stemmer technique to stem words to their root words. Once all the roots were found, we collapsed the sum of all the common roots, which helped in reducing data sparsity as most words occur only once in the corpus (as per Zipf's law) .

All these derived features boosted our accuracy by 0.5 - 1%. A simple word cloud visualization between the words used by males and females, Fig. 2 and 3, respectively shows a pattern of words usage by both genders. This shows how easy it is to classify genders based on textual information.

C. Image Features

The initial idea we started was to try detecting the faces in the image using the Viola Jones Haar cascades [4] and then using features from these faces to learn a classification which separates the males from females. However, due to the lower resolution of images this was not feasible. Many of the images (53%) had faces with size less than 24×24 pixels. A sample of an image with very small faces where the viola jones face detector fails is shown in Fig. 4 and a sample of an image where the face size is big enough for the detector to work well is shown in Fig. 5.

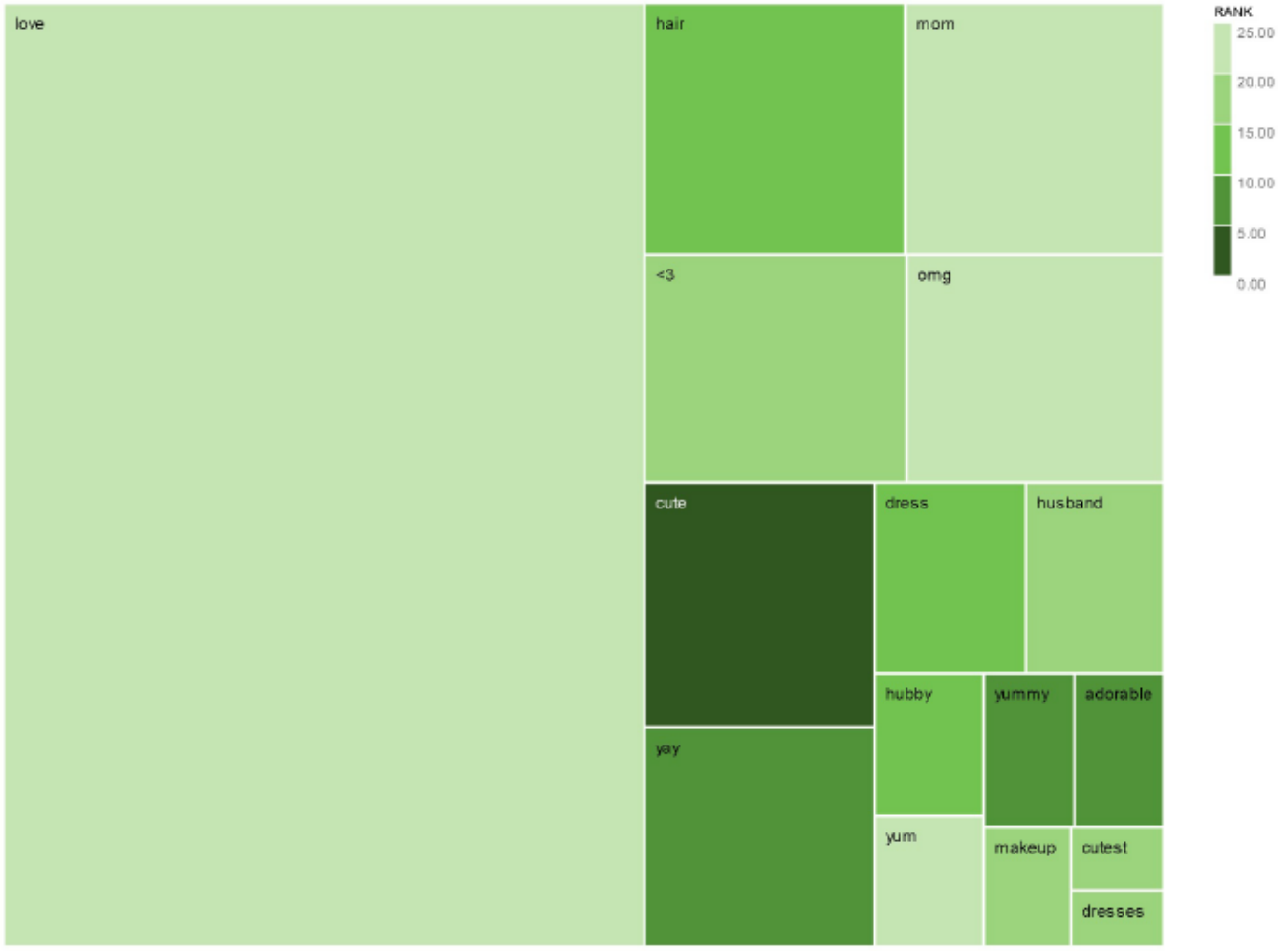

Fig. 1. Top Ranked Words by Counts.

Fig. 2. Ranked Words used Majorly by Males.

Fig. 3. Ranked Words used Majorly by Females.

The next thing we tried was using an off-the-shelf gender classifier based on face images. We used LibGRSM [5] which learns a structural classifier based on a generative approach and multiple-image features. The image features used by this package are LBP-Pyramid, Histogram of Gradients and Histogram of LBP. This package gave us an accuracy of 71% and took about 0.5secs to compute per image, hence we did not end up using this package. A simple output from LibGRSM

is shown in Fig. 11.

The third feature which we tried was GIST proposed by A. Torralba [6] for scene recognition. The idea here was GIST would capture the scene-like content in the image, i.e., males can generally have photos which which have more adventurous scene than females. To depict this, we plotted the GIST features of the mean male and female faces, these are shown in Figs. 6 and 7 respectively. Just by looking at these two images it is hard to see why this might work, however, plotting the difference between the 2 GIST Features (Refer to Fig. 8) we can clearly see the differences between male and female

Fig. 4. Case when Viola Jones Face Detector Fails.

Fig. 5. Sample Face size for Viola Jones Face Detector to work.

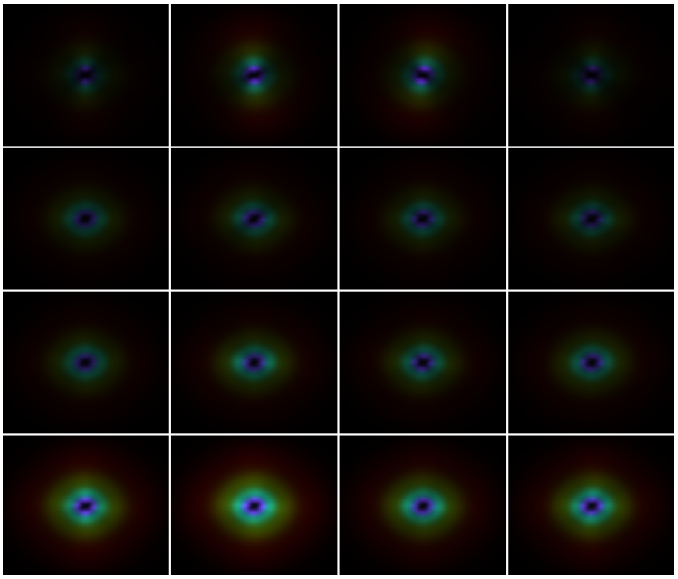

Fig. 6. GIST Descriptor for Average Male Face.

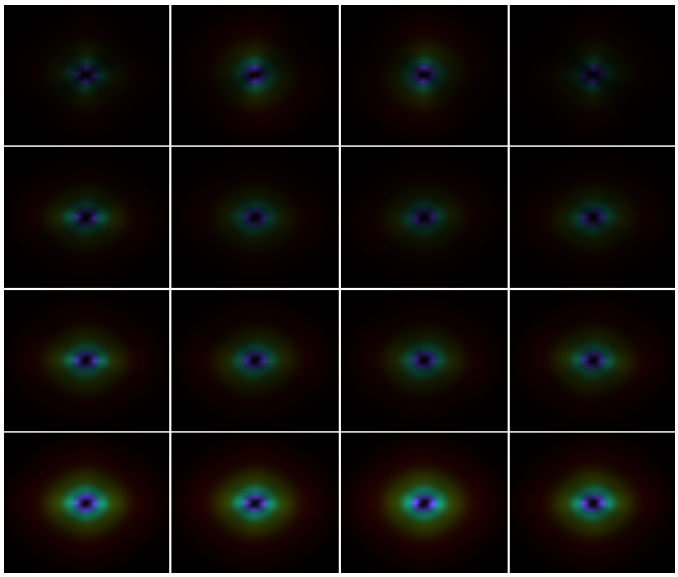

Fig. 7. GIST Descriptor for Average Female Face.

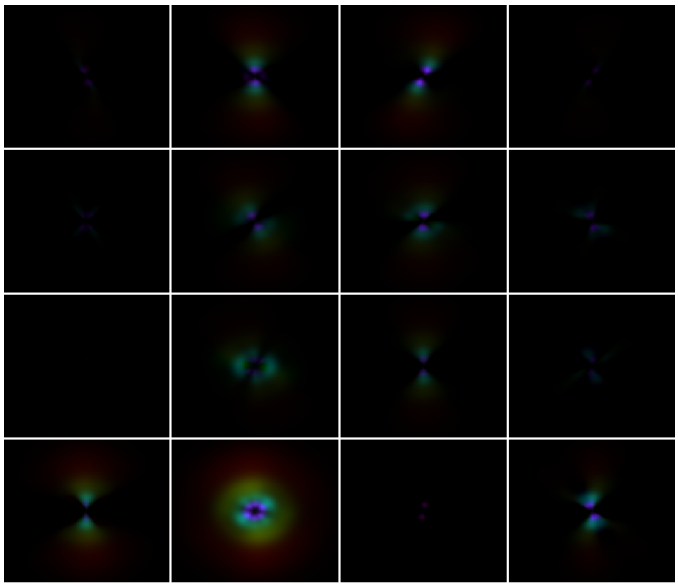

Fig. 8. GIST Descriptor for difference of Average Male and Female Faces.

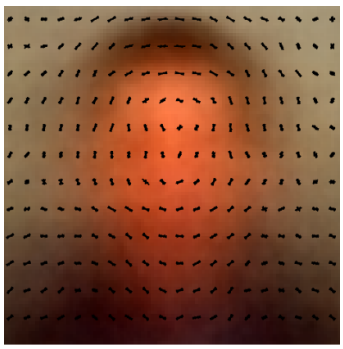

Fig. 9. HOG Descriptor for Average Male Face.

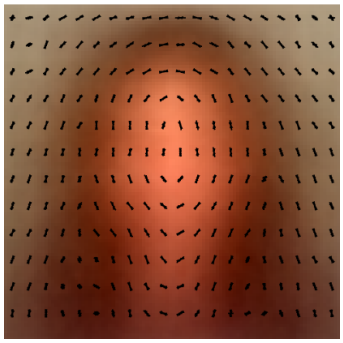

Fig. 10. HOG Descriptor for Average Female Face.

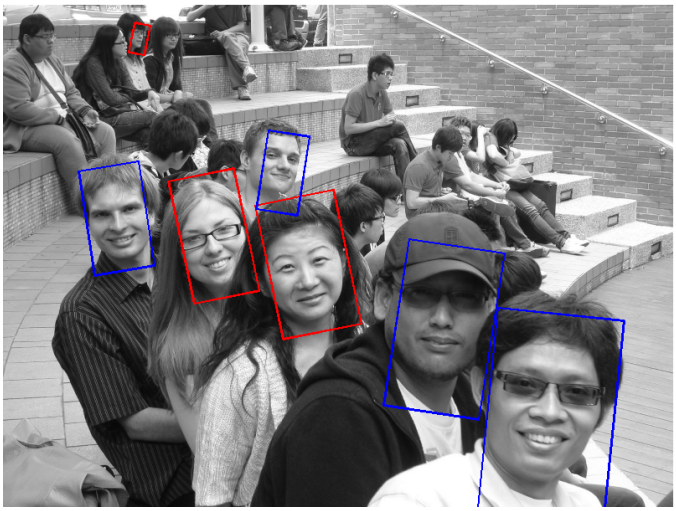

Fig. 11. Sample output from LibGRSM, red boxes show detected females and blue boxes show detected males.

GIST feature components. The GIST descriptor is a vector of features g, where each individual feature g_k is computed as

$$g_k = \sum_{x,y} w_k(x,y) \times |I(x,y) \otimes h_k(x,y)|^2$$

where $\otimes$ denotes image convolution and $\times$ is a pixel-wise multiplication. $I(x,y)$ is the luminance channel of the input image, $h_k(x,y)$ is a filter from a bank of multiscale oriented Gabor filters (6 orientations and 4 scales), and $w_k(x,y)$ is a spatial window that will compute the average output energy of each filter at different image locations. GIST by itself gives about 70-72% accuracy but when concatenated with words gave a 0.5 to 1.2% boost on the accuracy depending on the method.

The final feature which we tried was HOG and this is the one we ended up using due to its speed and accuracy boost. A very brief overview of steps used to compute HOG is given below:
To see why this might work, we plotted the HOG features of the mean male and female faces, these are shown in Figs. 9 and 10 respectively. Clearly, male pofile images have dominant horizontal edges and female profile images have dominant vertical edges partially because females generally have long hair. Just using HOG gives us an accuracy of about 68-70%. However, when concatenated with words gave a 0.1 to 0.3% boost on the accuracy depending on the method.

Algorithm 1: Algorithm to compute HOG Features

1. Divide the image into small connected regions called cells.
2. Compute edge orientations for the pixels within the cell.
3. Discretize each cell into angular bins according to the gradient orientation.
4. Normalize the histograms.

TABLE I
COMPARISON OF ACCURACIES OF DIFFERENT CLASSIFICATION METHODS ON RAW 5K WORD COUNTS.

Method	Train Accuracy (%)	Test Accuracy (%)
NaiveBayes	68	62
Decision Trees	78	72
K-Means	82	77
SVM	100	86
Adaboost	98	88
Logistic Regression	98	87
Logitboost	94	86
Robustboost	97	86

III. Model Selection and Model Integration

A. Model Selection

We started out by trying different classification methods like Naive Bayes, K-Nearest Neighbours, Decision trees, K-means, SVM with linear, intersection kernels, Logistic Regression and different boosting methods like Adaboost, RobustBoost, Logitboost on the raw word features. Then, we kept those methods that gave us at least 80% accuracy on the held out data set. The methods that gave us atleast 80% were SVM with intersection kernel, Logistic Regression, RobustBoost, LogitBoost and Adaboost.(Refer Table I)

Then we applied these four models on the different kinds of features which were extracted and selected in Section 2. Also, apart from feature extraction and feature selection we applied these models on those features with and without standardization and normalization (divide by L_2 norm of an observation). In total, we modelled 48 scenarios on different combinations of selected and extracted features. This led to a generation of roughly 220 models by our team. The entire summary of our experiments is documented here (https://goo.gl/XLqgrR). A table containing some of the best models from SVM, Logistic Regression and RobustBoost is given in Table II. Plots of these different types of features for SVM, LR and RobustBoost models are given in 12, 13, 14 respectively.

Analyzing these plots, we kept those models which had training accuracy of less than 97%(to avoid over fitting) given by the blue line in the plot and testing accuracy (cross-validation accuracy) of more than 84% (to avoid underfitting) given by red line. We then picked the top 10 models from the

TABLE II
COMPARISON OF ACCURACIES OF SVM, LOGISTIC REGRESSION AND ROBUSTBOOST ON RAW SELECTED FEATURES.

Method	Features Used	Train Accuracy (%)	Test Accuracy (%)
SVM-Intersection Kernel	5k Rank	92.11	89.29
SVM-Intersection Kernel	5k Rank + ImF	94.23	87.49
SVM-Intersection Kernel	3k Stem + Rank + ImF	92.14	87.99
SVM-Intersection Kernel	1k Std + Stem + Rank	95.92	85.19
LR	5k Rank	95.52	86.9
LR	5k Rank + Stem	95.6	86.8
LR	3k Rank + Stem	94.87	86.4
LR	1k Rank + ImF	92.3	87.3
RobustBoost 2000 Trees	3k Rank + ImF	97.7	88.19
RobustBoost 2000 Trees	3k Rank+ ImF + Std + L2 Norm	97.9	88.29
RobustBoost 2000 Trees	2k Rank + Stem + ImF	97.67	86.39
RobustBoost 2000 Trees	1k Rank + Stemm + ImF	95.79	83.88

overall list of models obtained from five different methods (SVM, LR, Adaboost, LogitBoost, RobustBoost) that had best test accuracies on the held out data set. The final models which we selected were

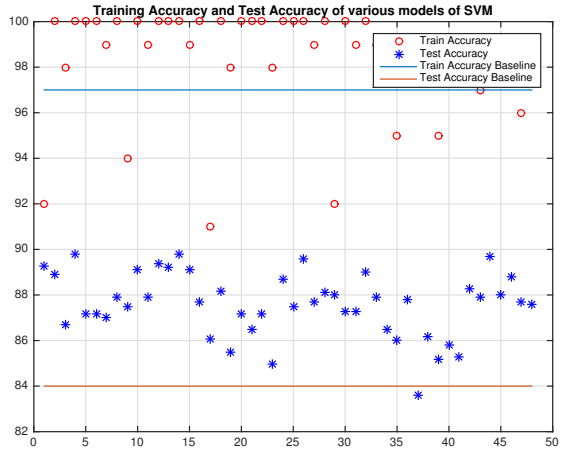

Fig. 12. Training accuracy and Test accuracy of various models using SVM with Intersection Kernel

- Logisitic Regression on 5000 word features, 7 image features and extracted HOG features from image. Then each observation was divided by L_2 norm.
- AdaBoost on 1000 ranked word features, 7 image features, 17 extracted word features and extracted HOG features from image. Then each observation was divided by L_2 norm.
- AdaBoost on 1000 stemmed and ranked word features, 7 image features, 17 extracted word features and extracted HOG features from image. Then each observation was divided by L_2 norm.
- AdaBoost on raw 5000 word features and 7 image features. Then each observation was divided by L_2 norm.
- RobustBoost on 1000 stemmed and ranked word features, 7 image features, 17 extracted word features and extracted HOG features from image. Then each observation was divided by L_2 norm.
- RobustBoost on 3000 stemmed and ranked word features,

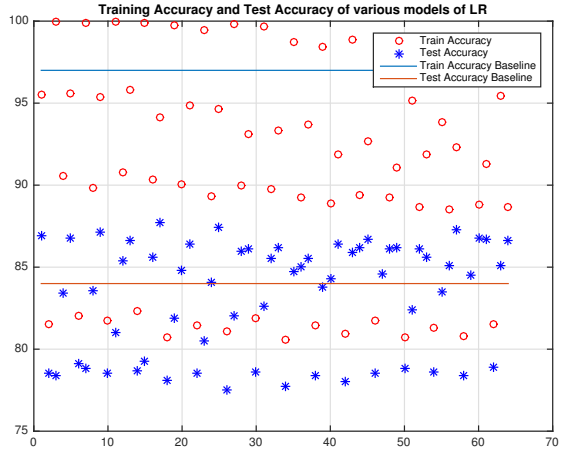

Fig. 13. Training accuracy and Test accuracy of various models using LR with Ridge penalty

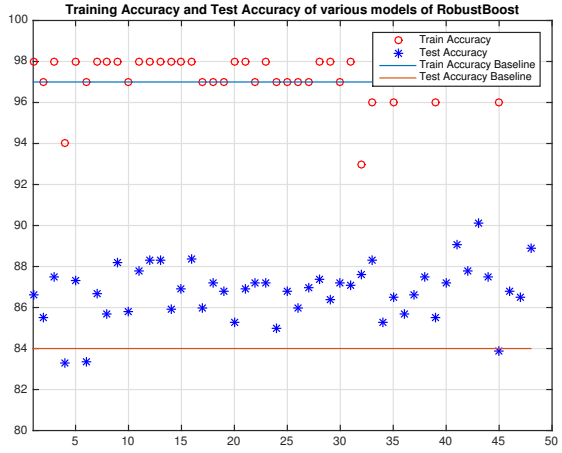

Fig. 14. Training accuracy and Test accuracy of various models using RobustBoost

7 image features, 17 extracted word features and extracted HOG features from image. Then each observation was divided by L_2 norm.
- LogitBoost 1000 stemmed and ranked word features, 7 image features, 17 extracted word features and extracted HOG features from image. Then each observation was divided by L_2 norm.
- SVM-Intersection Kernel on 1000 ranked word features, 7 image features, 17 extracted word features and extracted HOG features from image. Then each observation was divided by L_2 norm.
- SVM-Linear Kernel on 1000 ranked word features, 7 image features, extracted HOG features from image. Then each observation was divided by L_2 norm.

B. Model Integration

For combining different models, we tried a bunch of different methods.

- Voting : Voting did not help us much.
- Confidence scores : When making final prediction for a particular observation, we used only those models for which the confidence score was more than 90%. But since different models had different scales and ways of computing confidence scores, this did not help us much. We tried standardizing, normalizing, divide by max, but nothing helped us.
- Models on Confidence score : Then we tried fitting various models on these confidence scores to find the underlying pattern, but again it did not help us much because of the same reasons stated above.
- Models on Labels of different models: Finally, we tried fitting different models to the outputs of each model. Out of all the models we tried, Lasso gave us promising results. So we trained a lasso model on the output labels of each model and then obtained weights for different models. These weights and a manual threshold were used for the final predictions.

IV. Surprising Observations

During the course of this project, we dealt with several observations which were different from the intuitions we learnt in theory. Few of those are,

- PCA did not work as expected for feature reduction and dropped the accuracy drastically
- Standardization is an expected step when combining features of different scales. However, post-standardization, our models were always over-fitting. Standardization led to a reduction of accuracy by about 2%.
- Normalizing (l_2 norm) in Logistic Regression did not help as it was consistently under-fitting.
- Selecting top ranked 1000 or 2000 or 3000 words did not make a massive difference, albeit the rankings of the words made perfect sense.
- Image features by themselves gave very poor results with a maximum acccuracy of 75-77%.

V. Results

To beat the leaderboard, we ran a complex ensemble of several different models, one of which was the model built on GIST features. This helped us in reaching an accuracy of 96% on the leaderboard test-set. However, our final models had to fit under less than 50 MBs and needed to execute within 10 minutes. Due to these constraints, we had to immensely trim down our selection of models. The removal of model that hurt our accuracy the most was the one build on GIST features. The extraction of GIST features on the final validation set would have taken 13 minutes and hence it was impossible to include it in our final model set.

VI. Conclusion

This project was a great learning opportunity to understand the various Machine Learning techniques and their applications on real-world dataset. From feature selection to deriving more features, from running cross-validation to avoiding over-fititng, we understood the various aspects of applied Machine Learning.

Acknowledgment

The authors would like to thank Prof. Lyle Ungar and TAs, Barry Slaff and Levi Cai, for their constant guidance through the course and the project.

References

[1] Miller, Zachary, Brian Dickinson, and Wei Hu, *Gender prediction on Twitter using stream algorithms with N-gram character features*, 2012.

[2] WEKA, http://www3.stat.sinica.edu.tw/stat2005w/download/weka_050930.pdf

[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] Paul Viola, and Michael J. Jones, *Robust real-time face detection*, International journal of computer vision, Vol. 57, No. 2, pp. 137–154, 2004.

[5] Ondrej Fisar Bc., *Structural classifier for gender recognition*, Master's Thesis for Czech Technical University in Prague, 2011.

[6] Aude Oliva, and Antonio Torralba, *Building the gist of a scene: The role of global image features in recognition*, Progress in brain research, Vol. 155, pp. 23–36, 2006.