CIS 520 Project Final Report

GULTI (Venkata Bharath Reddy Karnati, Uzval Dontiboyina, Bhargav Kalluri)

Fall 2016

Contents

Introduction

For CIS 520 machine learning course final project, we developed a system for sentiment (Joy/Sad) analysis of twitter tweets. Out of the training data provided, we worked with a training set of 4500 labeled training samples, each having 10000 word features, 32 pre-extracted image features, 30000 (100x100x3) raw RGB image pixel features and 4096 CNN features. The run-time constraint for the auto-grader submission was 15 minutes for 4,500 test samples. Also, The submission size was limited to 50 Mb. Our best model on test set achieved an accuracy of 80.64%. Our submitted model for the final competition, involved feature selection based on bi-normal separation and information gain and application of multinomial Naive Bayes.

In the following sections, we present the cross-validation and auto-grader accuracies of each method we tried and discuss in detail about our models.

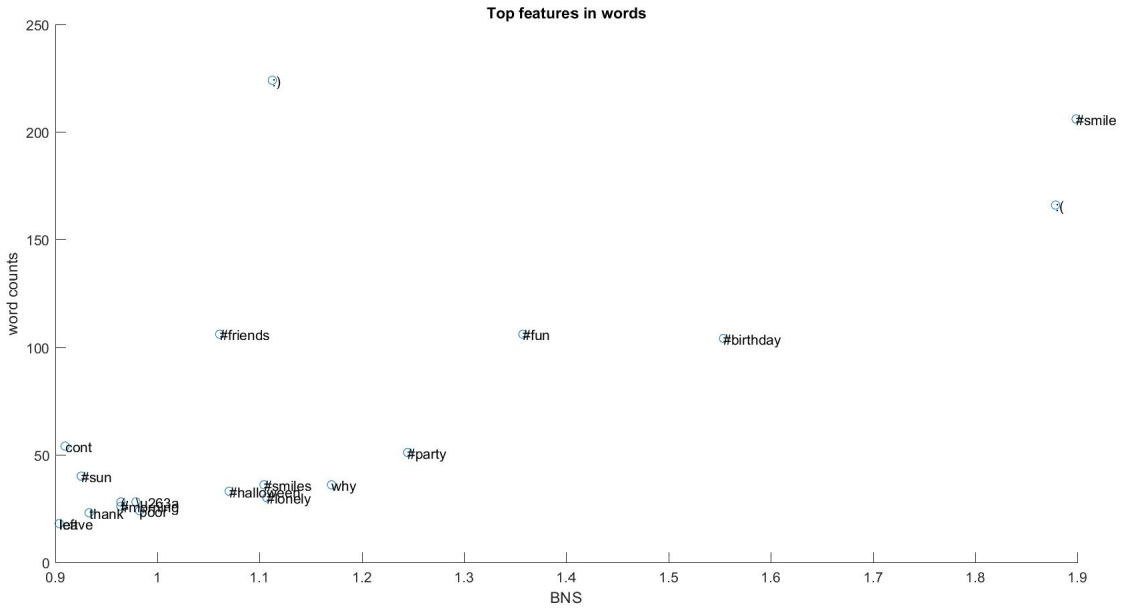
1 Methods Implemented

In this section, we discuss the results of several methods we implemented for dimensionality reduction, extracting features and classification of data.

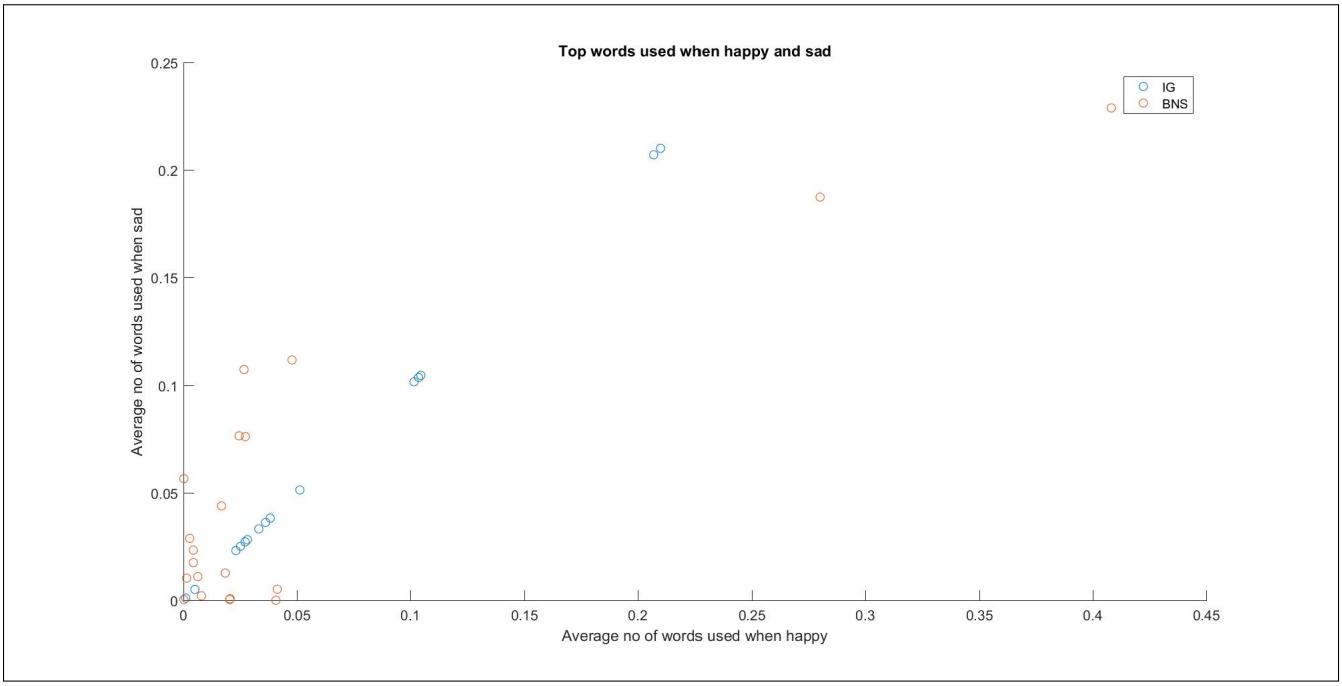
1.1 Dimensionality Reduction

1.1.1 Feature Selection Using Bi-normal Separation and Information Gain (BNS + IG)

To extract informative features from the raw words and image features, we first ranked word features based on their Information Gain (IG) and Bi-normal Separation (BNS), below is the comparison of the top 20 words selected by IG and BNS.



We can observe an improvement of around 0.7% - 0.9% by taking features selected based on IGs, over taking all the features into consideration.
And with features selected based on bi-normal separation, we observe an improvement of around 1.5% in accuracy.
Our final model included features selected from both the IG and BNS with an overall improvement of around 2%.



Finally, we use the effectiveness of this feature selection over several models like Naive Bayes, logistic regression and Logitboost with trees etc. We observed that when features selected by IG and BNS are combined, the results are better.

1.1.2 PCA

We used PCA on both words and images features. On testing this over several model, we found that it works relatively well on image features compared to words features. The details will be discussed in the following sections. Also because of the timing and size restrictions, we could not use desired number of principal components and get desired accuracies, and hence did not use this in the final model submission.

1.2 Classification on Tweets (Word Data)

1.2.1 Logistic Regression on Raw Word Data

As this is a binary classification task, logistic regression can be a good option to implement initially. We trained and classified PCA-ed word features with logistic regression using L2 regularization and obtained 70.61% accuracy. We later tried just using only raw word data which increased accuracy to 73.88%.

1.2.2 Adaboost

Adaboost is considered as an ideal algorithm for binary classification when there are sufficient number of features. Therefore, we implemented the algorithm for decision trees (weak classifiers) on BNS+IG feature selected word data. This resulted in a cross validation accuracy of 75.20%.

1.2.3 Logitboost (with Trees)

Logitboost is an optimization method which combines the logistic regression's cost function with adaboost. This can be used for binary classification of data. When implemented without feature selection, This model achieved a testing accuracy of 78.59%. This model was submitted for BASELINE 1. Later, when implemented with feature selection using "BNS+IG" , this gave an accuracy of 80.02%

1.2.4 PCA with Logitboost

This is a similar implementation of the Logitboost method used earlier.The only difference being that the feature selection is done with the help of PCA instead of BNS+IG. This method gives a cross validation accuracy of 75.20%.

1.2.5 KNN (with 3 Nearest Neighbors)

KNN is an instance-based approach to classification. This is non-parametric. For binary classification, it is helpful to choose K as an odd number as this helps in avoiding tied votes. So, we took 3 nearest neighbors for our classification on BNS+IG feature selected data. This performed rather poorly with an accuracy of 65.76%. This was probably because KNN is not good with word data and tries to over-fit.

1.2.6 SVM on Words

We trained a SVM with Linear Kernel on the top 3056 features selected by a combination of features obtained from information gain(851) and features obtained from BNS(3013). This gave a cross validation accuracy of 74.33%.

1.2.7 Multinomial Naive Bayes

Naive Bayes classifier is one of the most effective text classifiers.As the binomial classification for this project is heavily dependent on classifying text properly, we decided to go with this method.

Multinomial Naive Bayes is a variation of Naive Bayes that estimates the conditional probability of a particular word/term/token given a class as the relative frequency of term t in documents belonging to class c.

We obtained an accuracy of 80.64% using this method on BNS+IG feature selected word data. This accuracy was probably due to the low inter-correlation between the selected features.This was the best accuracy that we were able to achieve for this project.

1.3 Classification on Image Data

1.3.1 PCA with SVM on Images

SVM's generally tend to classify images better than most classifiers. So, we trained SVM on PCA'ed image train data with RBF kernel to generate our model. This resulted in an accuracy of 64.63%. The low accuracy might be due to the irrelevance of the image data to our particular classification.

1.4 Classification on Word + Image data

1.4.1 Cascading

1) We tried using cascading to ensemble our models for achieving better performance over single models. We performed "Naive Bayes" on BNS+IG feature selected word data and "SVM" on PCA'ed image data. Then finally performed "Logistic Regression" on normalized scores of above models. This yielded an accuracy of 80.24%.

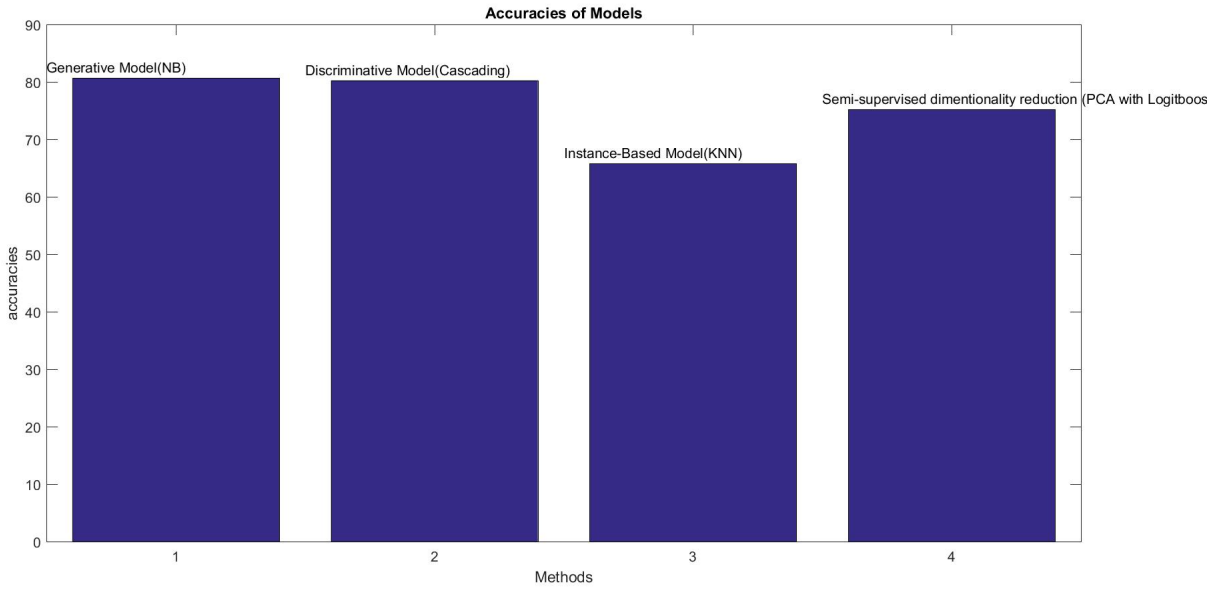
2) We also tried cascading by using "Logitboost" instead of "Naive Bayes" on the BNS+IG feature selected word data and SVM on the PCA'ed CNN features data. But, there was no significant improvement in the accuracy and the model was hardly fitting the memory constraints of the project. So, we tried stripping the model which resulted in a even lower performance.

1.5 Final Submission

Considering the project constraints and duration, we submitted all the models discussed above. As, per the submission requirements we implemented the following 4 methods:

Generative method: Naive Bayes
Discriminative methods:Logistic Regression, LogitBoost, SVM, Cascading, Adaboost
Instance-based method: K-nearest Neighbors
Semi-supervised dimensionality reduction of the data : PCA with SVM on images, PCA with Logitboost

The following figure compares the performance of the best models in each method.



2 Conclusion

Working on this project was a great learning process for our team and a chance to implement the things we learned in class on real world data albeit in a small scale . Here is a list of things that have surprised us (or have taught us a lesson):

- The main takeaway for us was to not consider all the data available while trying to solve a classification problem.
- We learnt to filter data according to our classification requirements by employing various feature selection methods like PCA, BNS and IG etc. Upon doing this, we realized that words data was more relevant than image data for sentiment classifier.
- As Images were less relevant to us, there was no need of normalizing the data, as only one dataset was needed at a time.
- The model which delivered the highest accuracy for us only trained word data. Though we observed an improvement in cross validation when we cascaded both image and word datasets, we were not able to replicate this with the test data due to time constraint.
- We also observed how powerful cross validation is in mimicking the test data results and successfully learnt to tune different parameters governing each model.

- We were surprised that simple algorithms like Naive Bayes gave far better results over much complex algorithms.
- We observed that Ensemble methods really boosted the performance of several classifiers with fair accuracies.

3 Appendix

The table shows the 5-fold cross-validation classification accuracies of different models on data whose feature selection(if any) has been specified in the brackets 3.

Table 1: Accuracies of different models

Methods	Models	Accuracy (%)
Generative Method	NB	$\approx 80.64\%$
Discriminative Method	Logistic Regression(IG+BNS)	$\approx 70.61\%$
	Logistic Regression	$\approx 73.88\%$
	Logit Boost(IG+BNS)	$\approx 80.02\%$
	Logit Boost	$\approx 78.59\%$
	SVM on words (IG+BNS)	$\approx 74.33\%$
	Adaboost (IG+BNS)	$\approx 75.02\%$
Cascading	NB on words + SVM on words +Logistic Regression	$\approx 80.24\%$
Instance based Method	KNN with K=3(IG+BNS)	$\approx 80.02\%$
Semi-supervised dimensionality reduction of data	PCA with SVM on images	$\approx 64.63\%$
	PCA with logit boost on words	$\approx 75.20\%$