CIS 520 Project Final: Twitter Gender Classification

Woodpecker (Yiren Lu, Xiang Deng, Dongni Wang)

Fall 2015

0 Introduction

In CIS 520 machine learing course final project, we developed a system for twitter users' gender prediction (male/female) based on their tweets and profile images. We were provided with a training set of 4998 labeled training samples, each has 5000 word features, 7 pre-extracted image features and 30000 raw RGB image pixel features. The time constraint for the final model(s) initialization and prediction is 3 minutes and 10 minutes, respectively, for 5,000 test samples. Also, The submission size is limited to 50 Mb in the final checkpoint. Our full system, including 7 models on different feature sets achieved 92.42% on test set. Our submitted model for the final competition, including 5 models, achieved an accuracy of 91.04% on the validation set, ranked 6^{th} in a total of 60 teams.

In the following sections, we present the cross-validation accuracies of each method we tried and discuss the rationale of our models. We also provide some interesting visualization such as the most predictive words and eigenfaces.

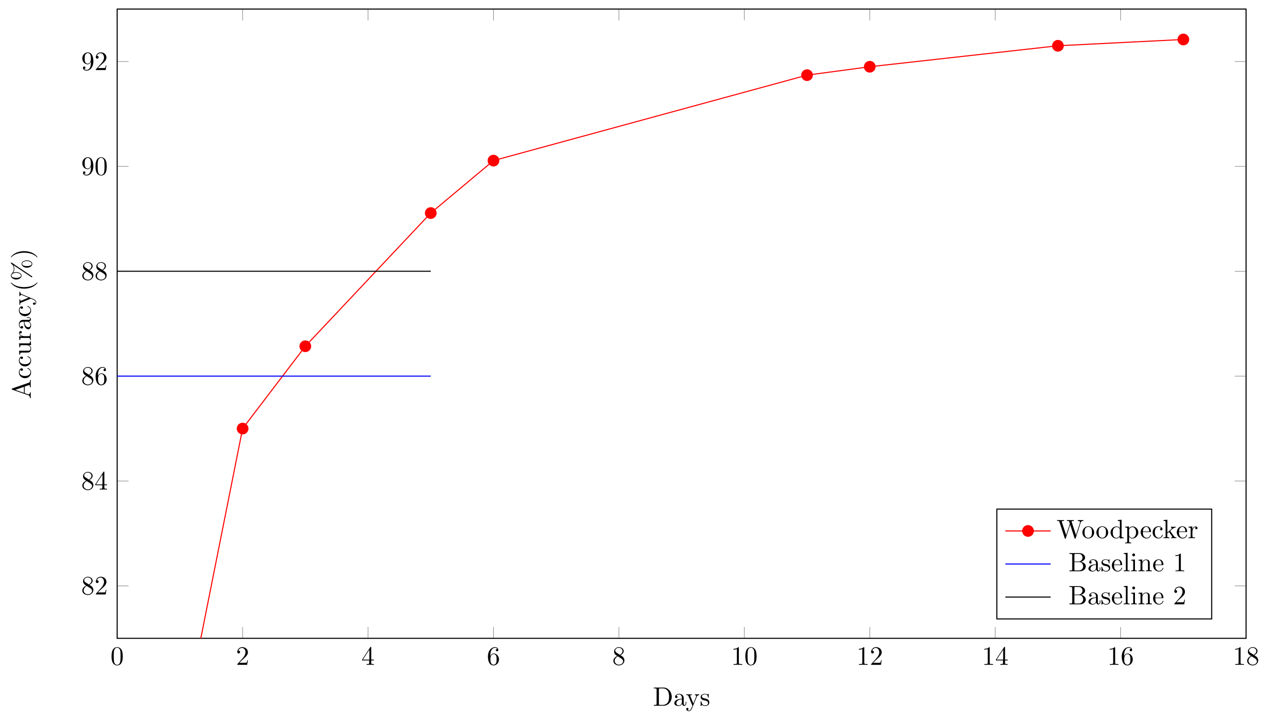


Figure 1: A plotting for Woodpecker's progress

Contents

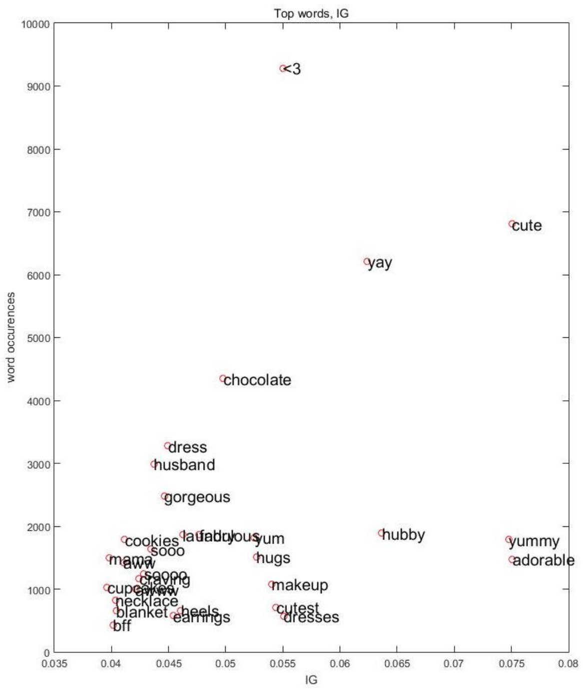
1 Methods

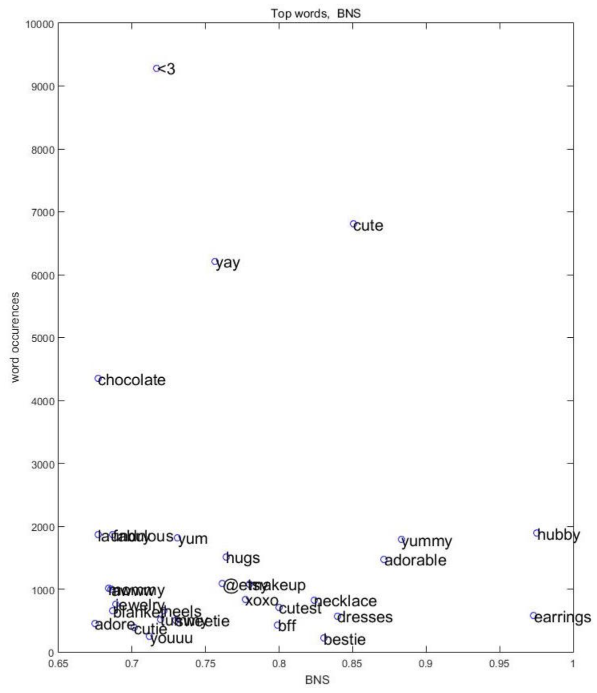
In this section, we report the results of multiple methods we tried for feature extraction, dimension reduction, and classification.

1.1 Dimensionality Reduction

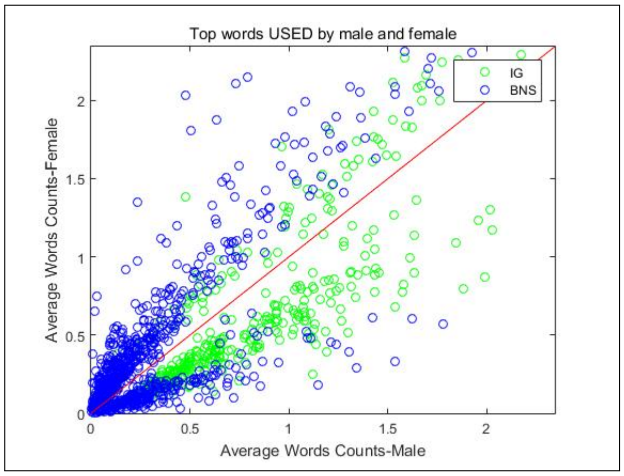
1.1.1 Feature Selection Using Information Gain and Bi-normal Separation

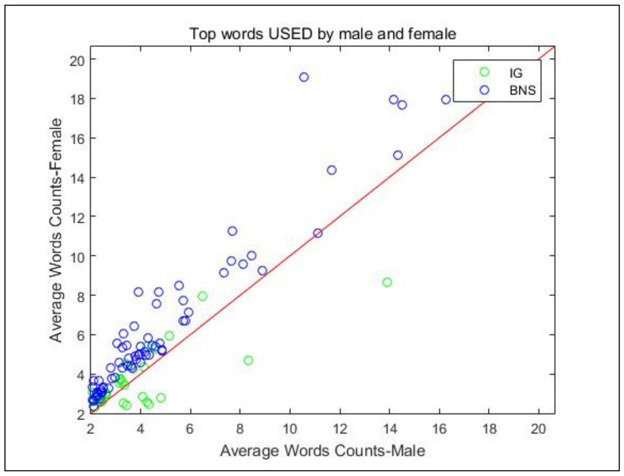
To extract informative features from the raw word and image features, we first ranked words features by based on their Information Gain (IG) and Bi-normal Separation (BNS), below is the comparison of the top 30 words selected by IG and BNS.





Next we examine the frequency of the top 1000 words selected by IG and BNS. Each circle in the graph below represents one word.





When the average words counts is less than 2.5, BNS selects the words that are more frequently used by either male or female.

Furthermore BNS tend to select more high frequency words used by female.
Finally, we test the effectiveness of the two feature selection methods by examining the improvement on cross validation accuracy over the following models: Logitboost with stump trees, logistic regression and naive bayes. BNS works well on Naive Bayes (though we didn't include this model in our final submission); IG improves the accuracy of Logitboosting with decision stump trees, no feature selection works better for logistic regression with L2 regularization on words.

1.1.2 PCA

We used PCA on both words and images feature. In this project, we found that PCA works relatively well on image features but not as good on words features. The details will be discussed with the following sections.

1.2 Classification on Tweets

1.2.1 Logistic Regression on Raw Words and Extracted Image Features

Since the project is a binary classification task. logistic regression might be a good choice for the first try. We first put PCA-ed words features and extracted images features into logistic regression with l2 regularization for training and classification, which beat baseline 1 with 86.01% accuracy. We later tried just use raw words features without extracted image features, which surprisingly, obtained 86.83% accuracy.

1.2.2 Neural Network

We employed 4 layers neural network (5000-100-50-2). We trained neural network 200 epochs with decreasing learning rate. For every 50 epochs, we decreased the learning rate to one-tenth of the last. The neural network produced around 86% accuracy on words features.

1.2.3 Logitboost with Stump Trees

Logitboost can be seen as a convex optimization which combines Adaboost with the cost function of logistic regression, which is suitable for binary classification. Using this model, our team acheives a tesing accuracy of 89.11%. More details will be discussed in 1.5.1.

1.2.4 SVM with Intersection Kernel

We trained two SVMs with Intersection Kernel on the top 1000 words+image features ranked by information gain. One model is trained on the original data and the other is trained on L2 normalized data. The former gives a cross validation accuracy of 87.10% and the latter gives 88.68%. More details please refer to 1.5.2.

1.3 Images Features Extraction and Classification

1.3.1 Grey-scale Images

We first converted the profile images data into R, G, B and Grey-scale images.

1.3.2 Face Detection with Viola-Jones Algorithm

We extracted all the faces. We used matlab built-in Viola-Jones algorithm to do face detection on RGB images, which detected faces on 72% of the profile pictures. All the following classifications over images were based on the faces detected. Later when ensemble, we only used samples with detected faces. For those samples without faces in them, we just set the output of the classifier as 0 (the output of the classifier is either positive or negative for male or female).

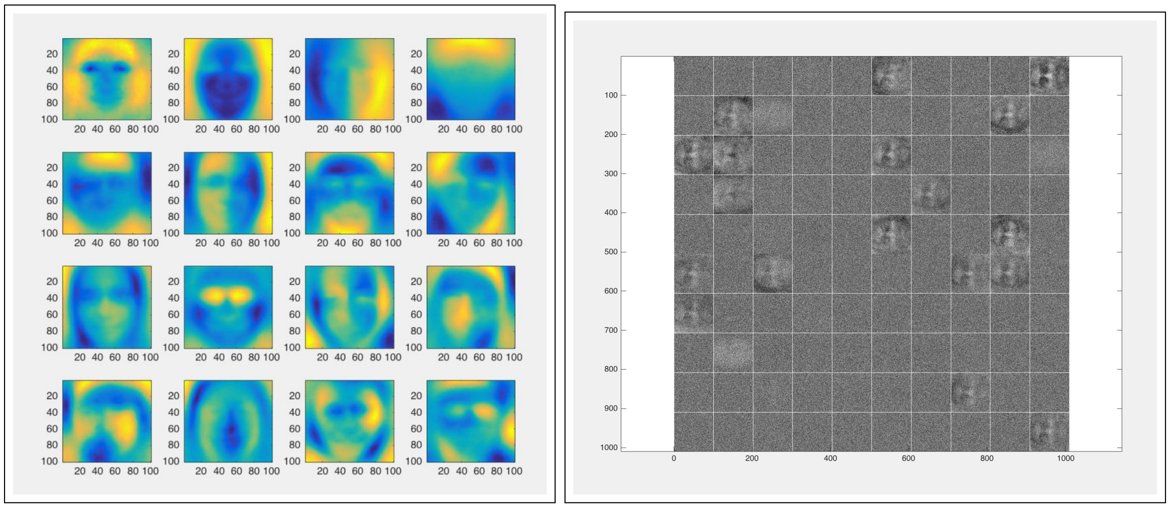


Figure 2: Visualization of Eigen Faces (left) and Auto-encoder (right)

1.3.3 Logistic Regression with Eigen Faces

After extracted faces, we did PCA over grey-scale images, the visualization of the top principal components are shown in figure below. We put top PCs into logistic regression classifier and it yielded around 70% accuracy over detected faces. The visulization is shown in figure 2

1.3.4 HOG Features, Gaussian Pyramid, Eyes and Nose

We used Viola-Jones algorithm to detect and crop eyes and nose. After that, we extracted Gaussian Pyramid HOG features on faces, eyes and nose into 7020 feature vector. Classifying the features with logistic regression yielded 82% accuracy by itself on 72% detected faces. The ensemble with this classifier produced 91.9% over all accuracy on the test set.

1.3.5 SVM on PCA-ed HOG Features

We then replaced the logistic regression classifier with a more powerful RBF kernel SVM. The accuracy on detected faces was improved up to 83%. The ensemble with this classifier got 92.3% accuracy on the test set. Then we did PCA (1500 PCs) on HOG features to reduce the dimensionality and the size of SVM model. The accuracy of the single model with improved to 84%. However, the ensemble with this classifier dropped to 92.14% accuracy on the test set.

1.3.6 SVM on PCA-ed Dense LBP Features

LBP works complementary with HOG features, we extracted spatial pyramid LBP features (15871 features) on detected faces and trained a SVM classifier over 2000 principal components of the LBP features. This model achieved 85% accuracy on detected faces. We integrated this model to the ensemble yielded our highest accuracy on the leaderboard of 92.42% on test set.

1.3.7 Auto-encoder

The Auto-encoder was employed for image dimensionality reduction on Grey-scale face images we detected and extracted. We experimented with various neural network settings and found the optimal number of hidden nodes is 100. We also set the learning rate to be 0.5, input zero-masked fraction to be 0.8, scaling-learning rate to be 0.95, non-sparsity penalty to be 0.1, and drop-out fraction to be 0.25. Our cross-validation using logistic regression shows over 75% accuracy on the face-detected images. The visualization of the auto-encoder is shown in figure 2.

1.4 Ensemble Methods

1.4.1 Stacking

We used stacking to ensemble our models to achieve better over all performance than any of the single models. First, we used 80% of the training data to train all the single models including logistic regression, neural network, logitboost with feature selection, kernel SVM with feature selection, kernel SVM with normalization, SVM over PCA-ed HOG, SVM over PCA-ed LBP. And then, use those models to generate scores (raw outputs) for the rest 20% data. Training a logistic regression model using the 20% scores with labels yields a ensemble classifier. Finally, we use all the training data to re-train all the single models. Along with the ensemble classifier, we got our final model.

1.4.2 Normalization

When ensemble all the models, we noticed that normalization actually works. For the images, we first trained logistic regression on raw gaussian pyramid HOG features (faces, eyes and nose), which yielded 82% accuracy on detected faces by itself. Later, we figured SVM works better by experiment (84%). However, when we ensemble the new model, it actually produced lower accuracy. We then found that the scores ranges produced by logistic regression and SVM were actually different. This may cause unbalanced weights across models. We then normalized all the scores with a sigmoid function with mean 0 and variance 2. This produced higher overall accuracy (92.3%).

1.5 Final Submission

Considering the time and space constraints, we finally submitted 5 models including logistic regression on words features, neural network, logitboost, kernel SVM and bagging of logistic regression models on HOG features. In this section, we analyze the over all performance of those 5 models.

1.5.1 LogitBoost and Feature Selection Using Information Gain

We want to use both words features and image features in order to enhance our accuracy, but we don't know which feature is most informative; therefore, we ranks all words and image features together using information gain and select the top features (In fact, by computing the information gain of image features, we found image feature 1 2 5 6 and 7 have roughly 0 information gain, but of course this does not mean all image features are useless).
We experiment with Logitboost based on the following considerations.
1. Words and the seven image features have different scales, we need a model that is scale invariant; boosting with decision stump trees in this case is a great choice.
2. The cost function using logistic regression is suitable for binary classification, it is a convex problem and minimizes the binomial deviance and gives less weights to misclassified observations.
In our final submission, we selected the top 1000 features ranked by information gain and used 300 decision stump trees. This model serves as a major classifier that contributes to our ensemble method.

1.5.2 Kernel SVM for Ensemble

The major motivation for introducing the Kernel SVM on words is to enhance our final ensemble accuracy. In order to do that, first we need to understand the contribution of each individual classifier to the ensemble. Below is the bar chart demonstration: (note when the ensembler only receives a single input, the outputs from Log-reg on words, our ensembler simply gives the predictions from the Log-reg classifier)
For our kernel SVM on words and image features, the Intersection Kernel is used for investigating the interactions between features (which we haven't done in the former models), and feature intersection might be a good way to obtain information about feature interactions.
We trained two Kernel-SVM on the top 1000 features based on information gain from the image feature and words training set: one on the original set; the other on the L2 normalized version of the data. We keep both models since in reality it's hard to say which model is less useful. Our goal is to enhance to the

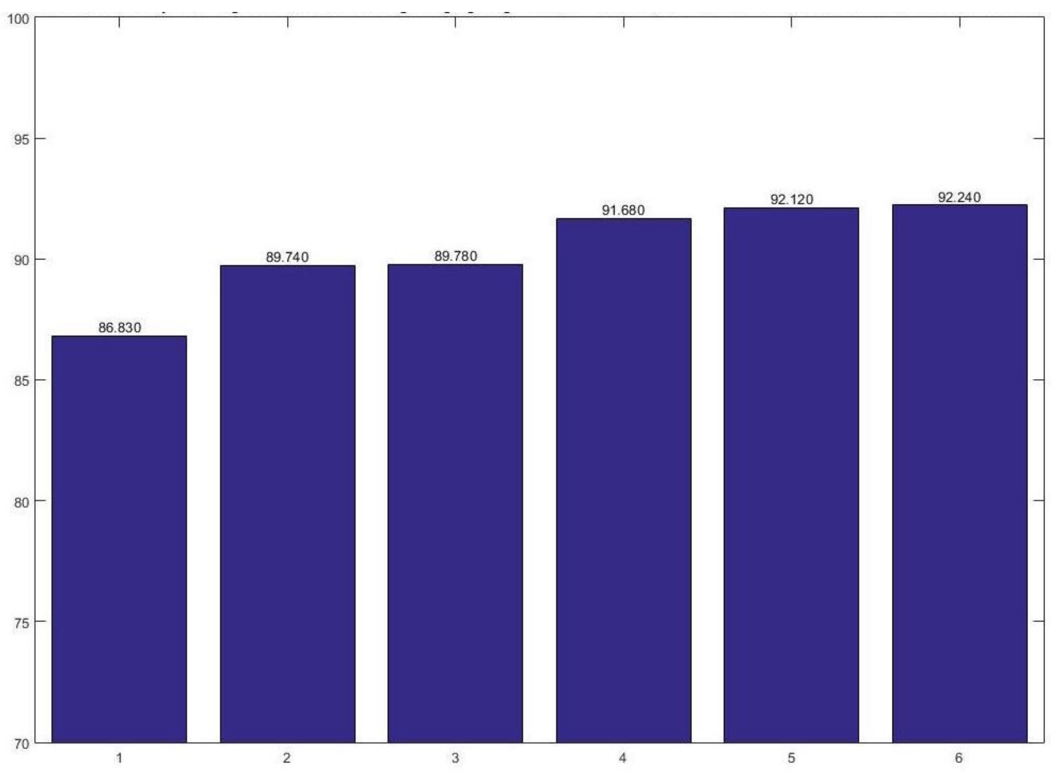


Figure 3: Cross-validation accuracy of the log ensembler after adding 1.Logistic Regression + LogitBoost 2.Neural Network 3.SVM+PCA-HOG 4.Kernel SVM 5.Kernel SVM-normalized

accuracy of our ensemble; therefore each model may give different but still useful information in different aspects. The bar plot below shows the improvement after adding the two models. (Note: all cross validation are performed on the same cross validation sets (5 sets in total)). This is the last ensemble combination we proposed one day before the submission deadline and unfortunately we haven't got a chance to obtain the accuracy on the testing dataset.

1.5.3 Bagging Logistic Regression on Raw HOG Features

To meet the time and space constraints of the competition, we dropped LBP model and and replaced the big SVM model with bagging logistic regression. Because the basis of PCs took 130Mb, which was too large for 50Mb space constraint, we didn't do PCA in the final submission either. Instead we used a bagging logistic regression with 6 logistic regression models, with each of them trained on 60 percent randomly resampled data from the original dataset. Bagging is another ensembling method we experiemented for thie project. In short, it trans multiple models on different subsets of original dataset, by doing so the model variance and overfitting can be reduced. We investigated on the bagging of Naive Bayes and Logistic regression. Bagging doesn't show significant improvements on Naive Bayes (in the writer's optinion, this makes sense, as Naive Bayes is just a simple categorical statistical model, a sum of multiple such models can been viewed as a big Naive Bayes model), but for logistic regression, as shown in our project demo, it can increase the accuracy of logistic regression on words and image features from 86.83% to 87.56%, if correcly configured; on HOG features, it increases the cross validation accuracy from 78.53% to 82.14%. After replacing SVM-HOG with baged logistic models, our cross validation accuracy dropped from 92.24% to 91.36%. The final model produced 91.04% accuracy on validation set. The models' accuracies are shown in figure 3

2 Discussion

Working on this gender-classification project gave our team a chance to reflect on what we have learned in class. Here is a short summary of things that have surprised us (or have taught us a lesson):

- With different feature sets (especially when they have various ranges and dimensions), feature selection and normalization have played an important role in improving the performances of our models.
- A huge part of this project is about trying various models, tuning their parameters and training with the right sets of features. Naive Bayes, for example, is around 40% more accurate when using the Bernoulli features (representing appearance/ absence) instead of its frequency. However, it does not improve the performance of our ensemble method and was not included in our final system. Methods like KNN, GMM and K-means require careful feature selections and we did not get the right feature sets to make it work well.
- Logistic Regression worked surprisingly well even without feature selection. It is both accurate and fast. Boosting using hundreds of weak learners also surprised us in terms of its accuracy. SVMs were expected to perform well and they did.
- Ensemble methods can really boost the performance with several classifiers with fair accuracies, while ideally theses classifiers disagree in many classifications (which means they capture different patterns).

3 Appendix

The table of single classifiers and their associated 5-fold cross-validation classification accuracies are shown in the Table 1. The approaches and 5-fold cross-validation classification accuracies of the ensemble method are shown in the Table 2.

References

[1] Rong-En Fan and Kai-Wei Chang and Cho-Jui Hsieh and Xiang-Rui Wang and Chih-Jen Lin, *LIBLINEAR: A Library for Large Linear Classification*, Journal of Machine Learning Research 9 (2008), 1871–1874.

[2] Chang, Chih-Chung and Lin, Chih-Jen, *LIBSVM: A library for support vector machines*, ACM Transactions on Intelligent Systems and Technology, 2:27:1–27:27, 2011.

[3] R. B. Palm, *Prediction as a candidate for learning deep hierarchical models of data*, 2012.

[4] Elkan, Charles, *Boosting and naive Bayesian learning*, Technical Report CS97-557, University of California, San Diego, 1997.

[5] Oza, Nikunj C, *Online bagging and boosting*, Systems, man and cybernetics, 2005 IEEE international conference, 3:2340 – 3:2345, IEEE, 2005.

Feature	Approach		Accuracy (%)
	Dimension Reduction	Classifier	
Words + Image features	PCA(500)	Ridge Regression + Sigmoid	$\approx$ 70%
Words + Image features	PCA(320)	Ridge Regression + Sigmoid	$\approx$ 79%
Words + Image features	PCA(2000)	Logistic Regression	$\approx$ 85%
Words	None	Logistic Regression	85.96%
Words	IG(1000)*	Logistic Regression	85.79%
Normalized-Words	None	Naive Bayes	72.25%
Words	IG(100)	multinomial Naive Bayes	77.95%
Words	None	Bernoulli Naive Bayes	79.59%
Words	IG(350)	Bernoulli Naive Bayes + EM	82.49%
Words	PCA(2000)	Artificial Neural Network	$\approx$ 86%
Words	IG(76)	K-Nearest Neighbor (L2)	72.89%
Words	IG(84)	K-Nearest Neighbor (Minkowski)	71.43%
Words	IG(95)	Random Forest	83.32%
Words	None	K-means	$\approx$ 60%
Words	IG(1000)	Decision stumps + LogitBoost	89.11%
Face-detected Image RGB	PCA(100)	Random Forest	$\approx$ 69%
Face-detected Image RGB	Auto-encoder(100)	Logistic Regression	75.17%
Raw HOG features over Face-detected Image RGB	None	Logistic Regression	$\approx$ 80%
Raw HOG features (Face/eyes/ nose-detected Image RGB)	None	Logistic Regression	$\approx$ 81%
Raw HOG features (Face/eyes /nose-detected Image RGB) + Gaussian Pyramid	None	Logistic Regression	$\approx$ 82%
Row HOG features (Face/eyes /nose-detected Image RGB) + Gaussian Pyramid	None	SVM (RBF kernel)	$\approx$ 83%
HOG features (Face/eyes /nose-detected Image RGB) + Gaussian Pyramid PCA(1500)	None	SVM (RBF kernel)	$\approx$ 84%
Dense LBP (Face-detected Image RGB)	None	SVM (RBF kernel)	$\approx$ 85%
*IG represents Information Gain			

Table 1: Experimental results of single classifiers

Approach		
Preprocessing	Classifier	Accuracy (%)
Features: Words + Image features		
IG(1000) on words and image features	Logistic (W) + Neural Network (W) + Ensemble trees (W+I) + stacking	91.11%
IG(1000) on words and image features	Logistic (W) + Neural Network (W) + Ensemble trees (W+I) + cascading	89.04%
Features: Words + Image features + Image RGB		
IG(1000) on words and image features; Face-detected image RGB PCA(100)	Logistic (W) + Neural Network (W) + Ensemble trees (W+I) + Logistic (PCA-ed RGB) + stacking	90.37%
IG(1000) on words and image features; Face-detected image HOG features	Logistic (W) + Neural Network (W) + Ensemble trees (W+I) Logistic (HOG) + stacking	91.55%
IG(1000) on words and image features; Face/eyes/nose-detected image HOG features	Logistic (W) + Neural Network (W) + Ensemble trees (W+I)+ Logistic (HOG) + stacking	91.74%
IG(1000) on words and image features; HOG features (Face/eyes /nose-detected Image RGB) Gaussian Pyramid	Logistic (W) + Neural Network (W) + Ensemble trees (W+I) Logistic (HOG) + stacking	91.9%
IG(1000) on words and image features; HOG features (Face/eyes /nose-detected Image RGB) Gaussian Pyramid + PCA(1500) Normalization	Logistic (W) + Neural Network (W) + Ensemble trees (W+I) Logistic (HOG) + stacking (SVM)	92.3%
IG(1000) on words and image features; HOG features (Face/eyes /nose-detected Image RGB) Gaussian Pyramid + PCA(1500) + Dense LBP + Normalization	Logistic (W) + Neural Network (W) + Ensemble trees (W+I) Logistic (HOG) + stacking (SVM)	92.42%
*For the classifiers: W: words; I: Image features		
*For the stacking method, we use logistic regression if not specified		

Table 2: Experimental results of ensemble classifiers