Realtime harassment detection on discussion comments using augmented moderation

Abhijit Pujare, A. Shukla
{apujare, *}@stanford.edu

Overview

Harassment is an unfortunate reality of online discussion, and the scale of forums makes them hard for humans alone to moderate. We devise a system which can **assist human moderation** using machine learning. The system starts with a baseline **logistic regression harassment comment classifier** with an 85% success rate. We then combine the baseline classifier with human moderation by (i) defining a notion of "classification confidence" and (ii) **passing low confidence classifications to a human** for final evaluation. The combined human and machine classifier has a success rate of 92.4%, higher than any publicly known classifier for this dataset.

Pipeline

Dataset

We used the "Detecting Insults in Social Commentary" dataset hosted on Kaggle and provided by Impermium software. The dataset is small: it contains 3947 comments of which 1049 are insults. As such, we must be careful to avoid overfitting.

Feature Extraction

We experiment with word and character n-grams with varying sequence lengths. Word n-grams perform poorly so we use character n-grams for all our final classifiers. We apply TF-IDF weighting on the n-gram vector in order to appropriately learn from uncommon sequences. To choose the appropriate hyperparameters for each classifier – n-gram type, n-gram length, regularization weight, and others – we implement random search where each combination of parameters is evaluated with k-fold cross validation with k set to 3.

Training

Our final classifier is trained using k-fold cross validation with k set to 3.

Models

Logistic Regression

We ran hyper parameter optimization on the regularization parameter in the logistic regression case. We use L2-regularization on the logistic regression loss function:

$$J_C = \frac{1}{m}\sum_{i=1}^{m} L(\theta^T x^{(i)}, y^{(i)}) + \frac{C}{2}||\theta||_2^2$$

Using the following set of possible values for the regularization parameter : `[0.1,1,5,50,100,1000,5000]`, the hyper parameterization optimization chooses a value of C=100.

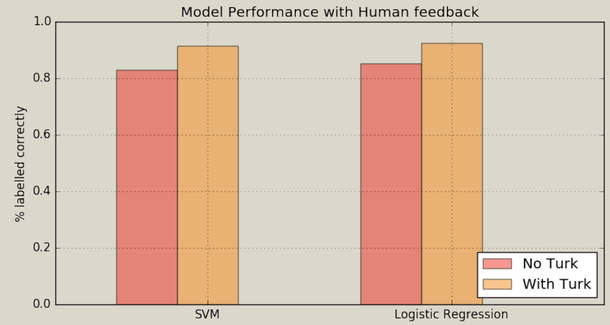
SVM

The SVM case uses the radial basis function as its kernel. We ran hyper parameter optimization on the regularization parameter and the gamma value in the rbf kernel. We use L1-regularization on the SVM optimization problem:

$$\min_{\gamma,w,b} \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\zeta_i$$

Discussion

As shown in the graph below, running SVM with a single human moderator improved classification by *8.56%* and for logistic regression classification improved by *7.25%*. More importantly, 78% of the gains - on average - were in precision so humans are helpful with identifying offensive comments.



Realtime Human Moderation

For a given classification by SVM and Logistic Regression, we define **confidence** as the distance to the separating hyperplane. We note that misclassifications are disproportionately also low-confidence classifications. As a result, we decide to set a **confidence threshold** below which we defer our classification to a human moderator. Because human moderators are capacity limited, our threshold is parameterized by the number of human moderators and set such that the expected number of comments selected for moderation is less than the total number which can be considered by humans.

We model a comment's confidence score as first being determined by the comment's type and then by a normal distribution for each given class, i.e.

$$S \sim pX_+ + (1-p)X_-; \quad X_+ \sim \mathcal{N}(\mu_+, \sigma_+), X_- \sim \mathcal{N}(\mu_-, \sigma_-).$$

We don't have the population value of the parameters so we estimate them from training data. This is the weighted sum of two Gaussians so can rewrite it as a single distribution. Our threshold t is found by inverting the capacity x:

$$t = \text{erf}^{-1}\left(\frac{x - p\mu_+ - (1-p)\mu_-}{p\sigma_+^2 + (1-p)\sigma_-^2 + p(1-p)(\mu_+ - \mu_-)^2}\right).$$