Hate Speech Detection using Semi Supervised Learning

Aditya Subash Rao
University of Illinois at Chicago
arao46@uic.edu

Dhwanit Sharma
University of Illinois at Chicago
dsharm37@uic.edu

1 Abstract

The important task of controlling hate speech on internet platforms begins with the task of identifying a piece of hate speech. Due to dearth of labeled data, we present two ways in which we can classify texts as hate speech with limited number of labeled samples and an unlabeled dataset. We also perform hate speech classification using a model trained with only positive and unlabeled samples.

2 Introduction

Internet has changed how we interact and connect with people. It allows people from different opinions and backgrounds to connect, share and discuss their personal views. Unfortunately,an opportunity to voice your opinion in a public forum freely can also give rise to the proliferation of negativity. An example of this is hate speech. Hate speech is defined as public speech that expresses hate or encourages violence towards a person or group based on something such as race, religion, sex, or sexual orientation. This behavior is prevalent in all kinds of platforms like social media and online gaming. The identification of hate speech can be done by handling it as a text classification problem, which requires us to have examples of hate speech and texts which don't contain hate speech. However, when we say try to find the opposite of hate speech, it contains every other type of speech. So, there is a paucity of negative examples of hate speech. The goal of the project is to train a model to identify instances of speeches in two scenarios, one when labeled data of both classes is available in limited quantity and when only positive and unlabeled data are available. We will compare results of both approaches. This will help reduce the rampant issue of hate speech and will be able to detect different types of hate speech. The scope will be limited to text-based conversations and posts.

3 Related Work

We have used the recent survey of semi-supervised learning by (Van Engelen and Hoos, 2020). Conceptually situated between supervised and unsupervised learning, it permits harnessing the large amounts of unlabeled data available in many use cases in combination with typically smaller sets of labeled data.
It present an up-to-date overview of semi-supervised learning methods, covering earlier work as well as more recent advances and focuses primarily on semi-supervised classification, where the large majority of semi-supervised learning research takes place.
The similar approach has been used for automatic fact checking which checks the worthiness of passing a text to a fact checking system. With applying this approach (Wright and Augenstein, 2020) have out-performed the state of the art models. The method is a unified approach using a variant of positive unlabelled learning that finds instances which were incorrectly labelled as not check-worthy. In addition to following Elkan and Noto's theorem for predicting using only positive examples, they also built a classifier that weights the unlabeled examples.r

4 Problem Statement

What constitutes hate speech? There is no definite set of rules that can reliably identify hate speech for what it is. At its core, hate speech tends to attack people for having certain characteristics, such as their race, skin color, ethnic group, religion, gender or sexual orientation – essentially harassing, intimidating or calling for violence against people for who they are.
There is a scarcity of annotated data for hate speech and no dearth of unlabeled data. This can be resolved by using a model which can leverage both types of data to accurately to classify hate speech.

Our main objective is to detect Hate Speech from a regular comment, posts and more importantly differentiate between a critique and hate speech using a semi-supervised deep learning model. Models that can learn using only positive examples and using limited number of limited samples.

5 Technical Approach

We are using the EM algorithm to train on the unlabeled data and the following deep learning models to classify the data.

5.1 PU Learning: Learning from Positive and Unlabeled Examples

The idea behind PU Learning is to learn from only positive examples and unlabeled examples alone.(Elkan and Noto, 2008) prove that the probability that a certain sample is positive [P(y=1|x)] equals the probability that the sample is labeled [P(s=1|x)] divided by the probability that a positive sample is labeled in our data set [P(s=1|y=1)].

5.2 Proof

Suppose the "selected completely at random" assumption holds. Then $p(y = 1|x) = p(s = 1|x)/c$ where $c = p(s = 1|y = 1)$.

Proof : Remember that the assumption is $p(s = 1|y = 1, x) = p(s = 1|y = 1)$. Now consider $p(s = 1|x)$. We have that

$$p(s = 1|x) = p(y = 1s = 1|x)$$

$$= p(y = 1|x)p(s = 1|y = 1, x)$$

$$= p(y = 1|x)p(s = 1|y = 1).$$

The result follows by dividing each side by $p(s = 1|y = 1)$.

So when we train our model, we give it a fraction of the positive data labeled as one and the rest of the unlabeled data as label 0. We keep a hold out from the positive set to calculate the average property that a an example is positive given it is labeled, i.e. c.

5.3 Expected Maximum Algorithm

The expectation-maximization algorithm is an approach for performing maximum likelihood estimation in the presence of latent variables. It does this by first estimating the values for the latent variables, then optimizing the model, then repeating these two steps until convergence.

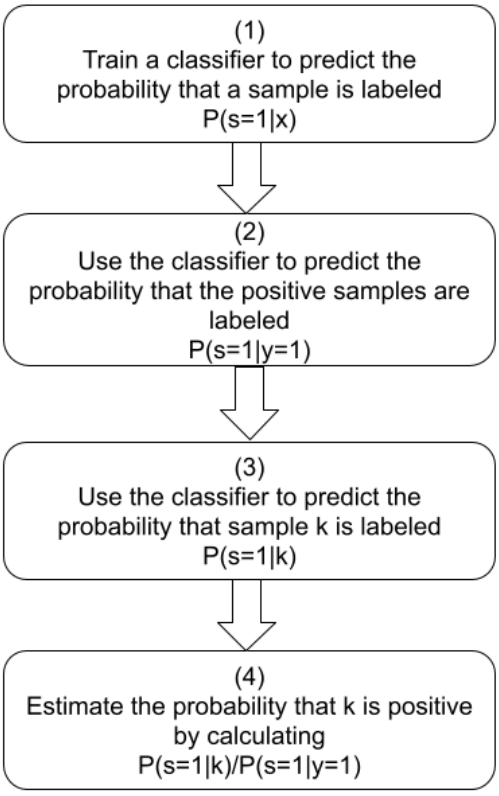

Figure 1: Steps to estimate the probability that unlabeled sample x is positive

It is an effective and general approach and is most commonly used for density estimation with missing data, such as clustering algorithms like the Gaussian Mixture Model.(Dellaert, 2002). For our approach we run the EM algorithm for only one iteration.

5.4 GloVe Word embeddings

We will train on GloVe embeddings with a dimension of 300.(Pennington et al., 2014). The main advantage of using GloVe is that, unlike Word2vec, GloVe does not rely just on local statistics (local context information of words), but incorporates global statistics (word co-occurrence) to obtain word vectors.

5.5 Long short-term memory(LSTM) model

LSTM layer consists of a set of recurrently connected blocks, known as memory blocks. Each one contains one or more recurrently connected memory cells and three multiplicative units – the input, output and forget gates – that provide continuous analogues of write, read and reset operations for

the cells(Graves and Schmidhuber, 2005). We are using an LSTM with 32 neurons.

5.6 Convolutional neural networks (CNNs)

A CNN typically has three layers: a convolutional layer, a pooling layer, and a fully connected layer. Convolutional layer performs a dot product between two martices, one is set of learnable parameters and other is the receptive field.
The pooling layer helps in reducing the spatial size of representation and decreases the required amount of computation and weights.
The Connected layer has full connectivity with all neurons in the preceding and succeeding layer.(Goodfellow et al., 2016)(Yamashita et al., 2018)

For both the models, we have added dropout layers, which randomly set 20% of the weights to zero, which prevents overfitting, which could be a common occurrence considering the small size of the data.

5.7 Bidirectional Encoder Representations from Transformers (BERT)

BERT is a transformer-based machine learning technique for natural language processing (NLP) pre-training developed by Google. BERT is at its core a transformer language model with a variable number of encoder layers and self-attention heads. The architecture is "almost identical" to the original transformer implementation in (Vaswani et al., 2017). There are 2 models of BERT that we have used, one with a 512 size representation and one with a length 768 word representattion, each with 12 transformer blocks. These are followed by a Dense fully connected layer and a Dropout Layer, followed by a sigmoid activated Dense neuron to display the classification output,

6 Experimental Setup

6.1 Data-Set Used

We have the used a public release of the dataset described in (Kennedy et al., 2020) consisting of 39,565 comments annotated by 7,912 annotators, for 135,556 combined rows.The primary outcome variable is the "hate speech score" which we have used as a deterministic factor to label the data as Hate Speech.
We have used a specific cutoff of 0 for labeling the data i.e. a sample with "hate speech score" 1.2 is labelled as "hate speech" and a sample with -0.5 score is labeled as "not hate speech" .

6.2 Semi Supervised Learning with EM algorithm

For initial approach, we will randomly sample different sizes of labeled data and make the rest unlabeled. We will train a model on the limited labeled set and predict the labels for unlabeled set.We will use the most confident examples from these prediction and retrain the model with a lower learning rate.

We used the Adam optimizer with learning rate 0.01 for the initial labeled set as we wanted the model to learn more from these examples. We used learning rate 0.001 when we included the newly labeled unlabeled samples. Following the retraining, we use the values of the sigmoid function to determine if the model is confident in its prediction or not.
Predictions outputs less than 0.05 are considered as negative, whereas outputs above 0.95 are considered positive. We started with 10% all the way to 50% of labeled train set. We monitored the binary cross entropy loss with a patience of 4 epochs and restored the best weights when loss was no longer decreasing.

6.3 PU Learning

For our approach of PU learning, we sample different percentages of the positive data and mark the rest of the data as unlabeled. We use the intervals 15%, 30% and 50% of positive data. Out of this data, 25% is reserved as a hold out set in order to estimate the probability of a labeled sample being positive, or as the proof refers to it, c. We use a bidirectional LSTM and 2 versions of the BERT Model with a dense neuron with sigmoid activation.

7 Results

In this section, we use the following evaluation metrics to evaluate the performance of our models.

Precision : Precision identifies the frequency with which a model was correct when predicting the positive class. That is:

$$Precision = \frac{TruePositive}{TruePositives + FalsePositives} \quad (1)$$

Recall : A metric for classification models that counts the correctly identified positive examples out of all the positive examples. That is:

$$Recall = \frac{TruePositive}{TruePositives + FalseNegatives} \quad (2)$$

F1 Score = This is the harmonic mean of precision and recall and captures the relationship between both metrics

	Before EM			After EM		
	Class 0	Class 1	Accuracy	Class 0	Class 1	Accuracy
Fraction of Dataset	f1-score	f1-score		f1-score	f1-score	
10%	0.73	0.74	0.73	0.71	0.75	0.74
20%	0.74	0.76	0.75	0.75	0.76	0.77
50%	0.75	0.78	0.76	0.75	0.78	0.78

Table 1: Results with LSTM on different sizes of labeled dataset

	Before EM			After EM		
	Class 0	Class 1	Accuracy	Class 0	Class 1	Accuracy
Fraction of Dataset	f1-score	f1-score		f1-score	f1-score	
10%	0.32	0.68	0.57	0.39	0.68	0.58
20%	0.40	0.66	0.57	0.37	0.67	0.58
50%	0.36	0.69	0.58	0.37	0.68	0.59

Table 2: Results with CNN on different sizes of labeled dataset.

	Class 0			Class 1			
Fraction of Dataset	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Accuracy
15%	0.68	0.81	0.74	0.77	0.62	0.69	0.72
30%	0.77	0.67	0.72	0.71	0.80	0.75	0.74
50%	0..78	0.73	0.75	0.75	0.79	0.77	0.76

Table 3: Results with BERT on different sizes of labeled data set.

	Class 0			Class 1			
Fraction of Dataset	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Accuracy
15%	0.71	0.68	0.72	0.74	0.74	0.74	0.71
30%	0.79	0.66	0.72	0.71	0.83	0.77	0.75
50%	0.80	0.67	0.73	0.72	0.83	0.77	0.75

Table 4: Results with BI-LSTM on different sizes of labeled dataset

	Class 0			Class 1			
Fraction of Dataset	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Accuracy
15%	0.75	0.71	0.73	0.73	0.77	0.75	0.74
30%	0.77	0.72	0.74	0.73	0.79	0.76	0.75
50%	0.84	0.62	0.71	0.70	0.88	0.78	0.75

Table 5: Results with BERT-768 on different sizes of labeled dataset

$$F-1Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (3)$$

7.1 Results of Semi Supervised Learning

For both the LSTM 1 and CNN 2, the accuracy and other performance metrics increase when we retrain the model with the most confident unlabeled data. The results show that the performance of the model increases by the most delta when you have 20% of labeled data in the beginning of training followed by retraining on the unlabeled data. The results can be found in Table 1 and Table 2. The performance plateaus with an increase of 2% across metrics.

7.2 Results of PU Learning

For the PU Learning part of the project, we observed that the performance increase is the most when we train with 30% of the positive examples and the rest be negative. The BERT 3 model performed the best on our data set on average. The Bidirectional LSTM 1 was not too far behind in performance. The BERT 768 5 model performance was better for smaller portions of the positive data.

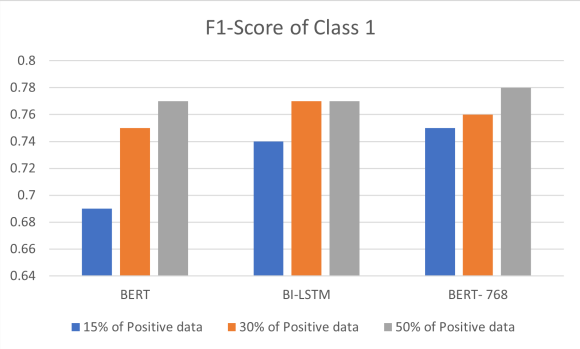

Figure 2: A Comparison of F1-scores with increasing percentage of labeled positive examples in PU learning

8 Future Work

In the future we can use the larger BERT models to see how far they improve the evaluation metrics and how that trades off with usage of computational resources.

References

Frank Dellaert. 2002. The expectation maximization algorithm. Technical report, Georgia Institute of Technology.

Charles Elkan and Keith Noto. 2008. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 213–220.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

A. Graves and J. Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm networks. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 4, pages 2047–2052 vol. 4.

Chris J Kennedy, Geoff Bacon, Alexander Sahn, and Claudia von Vacano. 2020. Constructing interval variables via faceted rasch measurement and multitask deep learning: a hate speech application. *arXiv preprint arXiv:2009.10277*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Jesper E Van Engelen and Holger H Hoos. 2020. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Dustin Wright and Isabelle Augenstein. 2020. Claim check-worthiness detection as positive unlabelled learning. *arXiv preprint arXiv:2003.02736*.

Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. 2018. Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, 9(4):611–629.