Hate Speech Detection on Social Media Data Using Deep Neural Networks

Muhammad M Rana

Advisor: Dr. Mehdi Allahyari

*Abstract*

*We address the problem of hate speech detection on online user contents and comments. Hate speech is defined as abusive speech targeting specific group characteristics, such as ethnic origin, religion, gender, or sexual orientation. Hate speech detection is crucial for applications like automated chatbots, product recommendation, sentiment analysing and many m0re. We reported series of experiments using the classic Bag of Words(BoW) approaches and using deep Neural Networks. Word level Convolutional Neural Network with pre-trained word vectors (Word2Vec, Glove) achieves excellent result. Long Short Term Memory (LSTM) networks has also shown great promises on detecting this problem.*

Motivation

With the increase of social media interaction there have also been a massive rise of hateful activities on social media. German government has already threatened to fine the social networks up to 50 million euros per year if they continue to fail to act on hateful postings (Thomasson, 2017). So, it has become a concerning issue for all of the world.

The manual way of filtering out hateful tweets is not scalable, motivating researchers to identify automated ways. The task is quite challenging due to the inherent complexity of the natural language constructs – different forms of hatred, different kinds of targets, different ways of representing the same meaning. Most of the earlier work revolves either around manual feature extraction or using representation learning methods followed by a linear classifier which is not accurate enough for this sort of task. Recent works on Natural Language Processing (NLP) using Deep Neural Networks (DNN) has shown impressive promises. So, we have experimented with a lot of possible combinations of DNN for rising up the accuracy on Hate Speech Detection Task.

Related Works

Nemanja et al. proposed a neural language model [2] where they have learnt the distributed representations of comments in a joint space using the continuous BOW (CBOW) model. This results in low-dimensional text embedding, where semantically similar comments and words reside in the same part of the space. Then, they used the embeddings to train a binary classifier to distinguish between hateful and clean comments. They have achieved upto 80% accuracy on their dataset.

Dennis et al. [4] has shown A Lexicon-based Approach for Hate Speech Detection using extreme lexicon analysis. Though their accuracy didn’t hit that

More recent work on Hate speech detection is done by Björn et al. [1] on their “Using Convolutional Neural Networks to Classify Hate-Speech” and by Pinkesh et al. [7] on their “Deep Learning for Hate Speech Detection in Tweets”. On the first one the authors have completed the experiment using two models. First model uses CNN and random word embeddings and the second model uses CNN and Word2Vec word embedding. On the later paper the authors have done an extensive work of classifying the hate speech using a variety of models. Besides CNN they have introduced LSTM for the first time for this job. They have also tried Glove word embeddings for training the embedding layer.

|  |
| --- |
| Word Embedding  A word embedding is a learned representation for text where words that have the same meaning have a similar representation. The most popular pre-trained word embedding vector model is Word2Vec which was developed by Tomas Mikolov, et al. at Google in 2013 as a response to make the neural-network-based training of the embedding more efficient and since then has become the de facto standard for developing pre-trained word embedding.  The Global Vectors for Word Representation, or GloVe, algorithm is an extension to the word2vec method for efficiently learning word vectors, developed by Pennington, et al. at Stanford.  DataSet Selection and Data Extraction  We have run our experiment on 16k annotated tweets labeled as hate speech (Racism, Sexism) and none. At the first step of the experiment we used the tweet ids for extracting the tweets using *tweepy* api. We have used batch extraction for reducing the http calls used on this work.  Experimented Methods   1. Bag of Words Approaches   As the benchmark method we started our experiment using the classic Bag of Words approach. After doing massive data cleaning, stop words removal and feature extraction we have run basic and Multinomial Naive Bayes, SGD (Stochastic Gradient Descent) and Linear SVM (Support Vector Machine) classifier on the dataset. We achieved upto 77% accuracy using these classic methods.    2. CNN with Random Word Embedding, inspired by Kim  At 2014, Yoon Kim has shown how to use CNN on text by transforming words into vector shapes. After this revolutionary work a lot of work has gone using his model for various aspects of classification. We have also used CNN with random initialized word embedding vector for classifying our hate speech data. We have experimented the CNN model both on *Tensorflow* and *Mxnet* framework. The later one is believed to be faster than the earlier one.    3. CNN using pre trained word embedding (Word2Vec, Glove)  On the next phase of our experiment we have experimented our dataset with a CNN model with pre-trained word embedding. And we found the interesting accuracy result. Glove word embedding gave us better accuracy than Word2Vec word embedding on our case.  4. LSTM with Glove word embedding  Then we tried the Glove word embedding on LSTM networks. The accuracy became better than the CNN networks. But the training time was more on some cases.  5. CNN + LSTM  At this level of our experiments we have tried some models that have never been used for detecting Hate Speech. We have tried a hybrid of CNN and LSTM model for reducing the training time and achieving better accuracy. The result comes out good and we are expecting more excel by tuning the hyperparameters.    *Fig : CLSTM (CNN+LSTM)*  6. Character Level CNN model  At 2015, Xhang et al. [12] have shown the promise of Character Level CNN for extracting more semantic information in text classification. At the final step of our experiment we have introduced this model in our case.  Results  Among the classic Bag of Words approaches Logistic Regression has given us the highest accuracy of ~77%. SVM has also achieved closer accuracy to that. CNN with Word2Vec embedding has achieved 82.9% and CNN with Glove Embedding has hit upto 83.6. LSTM has shown the most promising result. The basic LSTM model with 100 dimensional random vector embedding with 100 epochs has reached upto 85% accuracy. The hybrid of CNN and LSTM model has also reached closer to 85%. Unfortunately the Character Level CNN model hasn’t reached upto the expectation. We achieved near 78% accuracy using this approach. We are still experimenting with different hyper parameters in this Character level CNN model. |
|  |
|  |

Future Works

1. Bidirectional LSTM

LSTM in its core, preserves information from inputs that has already passed through it using the hidden state. Unidirectional LSTM only preserves information of the **past** because the only inputs it has seen are from the past. Using bidirectional will run the inputs in two ways, one from past to future and one from future to past and what differs this approach from unidirectional is that in the LSTM that runs backwards we preserve information from the **future** and using the two hidden states combined we are able in any point in time to preserve information from **both past and future**. What they are suited for is a very complicated question but BiLSTMs show very good results as they can understand context better. So, this could be a promising model for our case.

2. Hierarchical Attention Network

Hierarchical Attention Based neural networks have already shown great prospect for classifying large documents. This type of model have two levels of attention. — one at the word level and one at the sentence level — that let the model to pay more or less attention to individual words and sentences when constructing the representation of the document [13].

3. Dynamic Memory Networks

Besides these two we are also planning to experiment with this memory network model.

4. Multiple Dataset

For proving our model as a more generic one we need to test it on as many datasets as possible. We have another dataset containing 27k aggressive tweets that we are planning to run on our model for validating our dataset from a strong ground.

**Reference :**

1. **“Using Convolutional Neural Networks to Classify Hate-Speech”** [**http://www.aclweb.org/anthology/W17-3013**](http://www.aclweb.org/anthology/W17-3013)

# **Hate Speech Detection with Comment Embeddings**

<https://dl.acm.org/citation.cfm?id=2742760>

1. **Hateful Symbols or Hateful People? Predictive Features for Hate Speech Detection on Twitter**

[**http://www.aclweb.org/anthology/N16-2013**](http://www.aclweb.org/anthology/N16-2013)

**4. A Lexicon-based Approach for Hate Speech Detection**

[**https://preventviolentextremism.info/sites/default/files/A%20Lexicon-Based%20Approach%20for%20Hate%20Speech%20Detection.pdf**](https://preventviolentextremism.info/sites/default/files/A%20Lexicon-Based%20Approach%20for%20Hate%20Speech%20Detection.pdf)

**5. A Survey on Hate Speech Detection using Natural Language Processing**

[**http://www.aclweb.org/anthology/W17-1101**](http://www.aclweb.org/anthology/W17-1101)

6. **Automated Hate Speech Detection and the Problem of Offensive Language**

[**https://arxiv.org/abs/1703.04009**](https://arxiv.org/abs/1703.04009)

**7. Deep Learning for Hate Speech Detection in Tweets**

[**https://dl.acm.org/citation.cfm?id=3054223**](https://dl.acm.org/citation.cfm?id=3054223)

**8. Detecting hate speech on the world wide web**

|  |
| --- |
| [**https://dl.acm.org/citation.cfm?id=2390377**](https://dl.acm.org/citation.cfm?id=2390377)  **9. A C-LSTM Neural Network for Text Classification**  [**https://arxiv.org/abs/1511.08630**](https://arxiv.org/abs/1511.08630)  **10. Character-level Convolutional Networks for Text Classification**  [**http://papers.nips.cc/paper/5782-character-level-convolutional-networks-for-text-classifica**](http://papers.nips.cc/paper/5782-character-level-convolutional-networks-for-text-classifica)  **11. Convolutional Neural Networks for Sentence Classification by Kim**  [**http://www.aclweb.org/anthology/D14-1181**](http://www.aclweb.org/anthology/D14-1181)  **12.**[**http://papers.nips.cc/paper/5782-character-level-convolutional-networks-for-text-classifica**](http://papers.nips.cc/paper/5782-character-level-convolutional-networks-for-text-classifica)  **13.**[**http://www.cs.cmu.edu/~./hovy/papers/16HLT-hierarchical-attention-networks.pdf**](http://www.cs.cmu.edu/~./hovy/papers/16HLT-hierarchical-attention-networks.pdf) |

|  |
| --- |
|  |