NLP Applications

Final Project Report

Hate Speech Detection on Social Media Data

By
-Ravsimar Singh
-Vaibhav Bajaj

1 Overview

Our project consists of Hate speech detection on twitter data. We classified tweets into 3 categories - Hate Speech, Offensive and neither. For the baseline, we used a basic ML techniques such as SVMs, Logistic Regression to classify tweets. We also used with a basic single-layer LSTM neural network.

Extending this, we moved on to more complex models, multi-layer LSTMs, GRUs, CNNs. Various parameters were varied and tried out to find the optimal classifier for the given data.

2 Literature Review Dataset

We have referred to [Zhang and Luo, 2018], [Zimmerman et al., 2018], [Badjatiya et al., 2017], [Waseem and Hovy, 2016]. Our dataset is same as that of [Waseem and Hovy, 2016].

3 Working

3.1 Preprocessing and Tokenization

- Usernames indicate to whom the tweet is addressed to. These types of tokens are unique to social media and do not contribute much to the actual meaning being conveyed in the sentence, hence all usernames (@username) are removed.
- Hashtags are often used in tweets and can indicate a topic or sentiment. All hashtags in the tweets are preserved.
- All other special characters are removed. URLs are also replaced with a keyword 'URL'
- All remaining clean tokens are converted to lowercase. An in built tokenizer is used to tokenize the cleaned tweets.

Some specific techniques for feature extraction and preprocessing used were:

3.1.1 Stopword Removal

A stop word in a commonly used word (such as "the", "a", "an", "in"). These are usually removed as a part of preprocessing since they are only used to provide fluidity to the sentence.

3.1.2 Vectorization

The general process of turning a collection of text documents into numerical feature vectors is known as vectorization. In Bag of words or "Bag of n-grams" representation, documents are described by word occurrences while completely ignoring the relative position information of the words in the document.

We used the sklearn library for this function. It provides utilities for the most common ways to extract numerical features from text content, namely:

- *tokenizing* strings and giving an integer id for each possible token, for instance by using white-spaces and punctuation as token separators.

- *counting* the occurrences of tokens in each document.
- *normalizing* and weighting with diminishing importance tokens that occur in the majority of samples/documents.

In this scheme, features and samples are defined as follows:

- each *individual token occurrence frequency* (normalized or not) is treated as a *feature*.
- the vector of all the token frequencies for a given document is considered a multivariate sample.

In a large text corpus, some words will be very present (e.g. "the", "a", "is" in English) hence carrying very little meaningful information about the actual contents of the document. If we were to feed the direct count data directly to a classifier those very frequent terms would shadow the frequencies of rarer yet more interesting terms.

In order to re-weight the count features into floating point values suitable for usage by a classifier, it is very common to use the tf–idf transform.

Tf means term-frequency while tf–idf means term-frequency times inverse document-frequency.

3.2 Models

3.2.1 Logistic Regression

Initially we started with this basic technique as some of the papers that we were referring used Logistic Regression to classify the tweets. Accuracy in case of logistic regression along with wordVectorizer was pretty decent but recall was low. logistic regression along with TF-IDF didn't perform well. The specific scores are mentioned in the table.

3.2.2 SVM

The Accuracy in case of SVM was fairly low(lower than logistic regression as well) and recall was very bad.

3.2.3 LSTM

As we can see the recall in both the above cases was very low irrespective of accuracy. One of the reason is, the data was highly unbalanced (out of 25k tweets only 1.5k tweets were hate speech, 4.5k neither hate speech nor offensive and rest 19k were offensive language). So to improve the recall we started with techniques such as, LSTM. LSTM clearly outperformed the above methods and the recall was fairly decent considering the biased nature of data.

3.2.4 GRUs

The results were on par with the LSTM results, but significantly better than SVMs and Logistic Regression.

4 Results

We calculate the F-Score, Accuracy and the Recall for every model that is trained.

4.1 Logistic Regression

Table 1: Logistic Regression - Scores

Method	F-Score	Recall	Accuracy
Logistic Regression - Word - Count	0.884	0.666	0.898
Logistic Regression - Char - Count	0.884	0.666	0.898
Logistic Regression - Word - TF IDF	0.723	0.386	0.792
Logistic Regression - Char - TF IDF	0.752	0.418	0.809

4.2 SVM

Table 2: SVM - Scores

Method	F-Score	Recall	Accuracy
SVM - Word - Count	0.672	0.333	0.772
SVM - Char - Count	0.682	0.333	0.777
SVM - Word - TF IDF	0.677	0.333	0.775
SVM - Char - TF IDF	0.676	0.333	0.777

4.3 CNN

Table 3: CNN - Scores

Method	F-Score	Recall	Accuracy
CNN-2-layer	0.87	0.67	0.88
CNN-5-layer	0.88	0.69	0.88

4.4 LSTM

Table 4: LSTM - Scores

Method	F-Score	Recall	Accuracy
LSTM-128-adam	0.90	0.70	0.90
LSTM-128-rmsprop	0.92	0.77	0.91
LSTM-128-adam	0.89	0.70	0.89
LSTM-256-rmsprop	0.87	0.67	0.88
MultiLayer-LSTM-128-rmsprop	0.92	0.78	0.91

4.5 GRU

Table 5: GRU - Scores

Method	F-Score	Recall	Accuracy
GRU-128-adam	0.90	0.71	0.90
GRU-128-rmsprop	0.91	0.74	0.91
GRU-256-adam	0.90	0.70	0.89
GRU-256-rmsprop	0.91	0.74	0.90

References

[Badjatiya et al., 2017] Badjatiya, P., Gupta, S., Gupta, M., and Varma, V. (2017). Deep learning for hate speech detection in tweets.

[Waseem and Hovy, 2016] Waseem, Z. and Hovy, D. (2016). Hateful symbols or hateful people? predictive features for hate speech detection on twitter. In *SRW@HLT-NAACL*.

[Zhang and Luo, 2018] Zhang, Z. and Luo, L. (2018). Hate speech detection: A solved problem? the challenging case of long tail on twitter. *CoRR*, abs/1803.03662.

[Zimmerman et al., 2018] Zimmerman, S., Kruschwitz, U., and Fox, C. (2018). Improving Hate Speech Detection with Deep Learning Ensembles. In chair), N. C. C., Choukri, K., Cieri, C., Declerck, T., Goggi, S., Hasida, K., Isahara, H., Maegaard, B., Mariani, J., Mazo, H., Moreno, A., Odijk, J., Piperidis, S., and Tokunaga, T., editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).