

Week 13: Deliverables

Group Name: Awwab Ahmed

Name: Awwab Ahmed

Email: awwab.k.ahmed@gmail.com

Country: USA

College/Company: University of South Florida

Specialization: NLP

Problem description:

The term hate speech is understood as any type of verbal, written or behavioural communication that attacks or uses derogatory or discriminatory language against a person or group based on what they are, in other words, based on their religion, ethnicity, nationality, race, color, ancestry, sex or another identity factor. In this problem, We will take you through a hate speech detection model with Machine Learning and Python.

Hate Speech Detection is generally a task of sentiment classification. So for training, a model that can classify hate speech from a certain piece of text can be achieved by training it on data that is generally used to classify sentiments. So for the task of hate speech detection model, We will use the Twitter tweets to identify tweets containing Hate speech.

Data cleansing and transformation:

- 1) Remove Unnecessary Characters:
 - a) These include the @ symbol and other characters that aren't letters or numbers.
 - b) These characters have no effect on whether a tweet is considered hate speech and are therefore irrelevant to our model.
 - c) In fact, not removing them can hurt your model.
 - d) This will be done using the regex module.
- 2) Standardization:
 - a) This includes making everything lowercase, fixing misspellings, etc.
 - b) These factors could also cause noise that hurts the model.
 - c) These will be done using the spacy and NLTK modules.
- 3) Tokenization & Lemmatization:
 - a) This means splitting the tweets into smaller parts (ex: words) and lemmatization is getting the base of each word.
 - b) This is a form of standardization that leads to better model results.
 - c) This will also be done using spacy and NLTK modules.
- 4) Remove NA values:
 - a) This means getting rid of empty entries in the dataset if there are any using pandas dataframe functions.
- 5) Featurization:
 - a) TF-IDF or Word Embeddings

- b) Both methods will be tested.
- c) TF-IDF uses the number of occurrences of a word to draw connections between words while word embeddings considers word meaning and sentiment.

EDA:

Because this is an NLP project, the exploratory data analysis is different compared to the typical data analysis methods. We will explore:

- 1) Average length of tweet (in characters)
- 2) Average length of words (in characters)
- 3) The top most frequently occurring words
- 4) Word Clouds
 - a) We will do this for both positive and negative sentiment and it will show the more common words through size and correlated words through distance.

All the above is displayed in the jupyter notebook file in this folder.

Model Selection and Model Building

Selected Model: LSTM recurrent neural network

“LSTM stands for Long-Short Term Memory. LSTM is a type of recurrent neural network but is better than traditional recurrent neural networks in terms of memory. Having a good hold over memorizing certain patterns LSTMs perform fairly better. As with every other NN, LSTM can have multiple hidden layers and as it passes through every layer, the relevant information is kept and all the irrelevant information gets discarded in every single cell.”

Source: <https://www.analyticsvidhya.com/blog/2021/06/lstm-for-text-classification/>

- First, tokenization was applied.
- Using the 0 and 1 labels, it was time to classify.
- We utilized tensorflow and keras to create a rnn using the LSTM structure.
- We, then, trained the model.

TIME FOR DEPLOYMENT!
COMPLETE!