Individual Project – Free Topic

Anant Ashutosh Sharma
anantas2@illinois.edu

Topic

Text and Tweet Classification using Machine Learning

What is the task ?

The task involves classifying texts into their relevant categories using different machine learning techniques. Text classification is one of the standard applications in text mining. . The objective of our text classification task is to find appropriate labels for previously unlabelled data from a predictive model which has been trained on a pre-labelled dataset. A series of necessary subtasks are performed to identify and extract relevant features from a given text, which can be further applied to train a predictive model

The text to be classified can either be a sentence or a group of sentences i.e. a paragraph. Depending on the labels of the dataset, the text can be classified in binary labels or multiple labels. For instance, on training the models on SMS spam dataset, it will be able to accurately classify texts as Spam (spam) or Not Spam (ham), whereas on training the models on the Hate-Speech and Offensive Language Data, it will achieve the task of classifying text into hate speech, offensive language or neither

As of now, I plan to train models to

1. Classify emails in to Spam or Not Spam
2. Classify text into different hate-speech, offensive language categories
3. Classify political tweets using a newly created dataset

Why is it important or interesting ?

The internet is a hub of textual information. Users are mostly overwhelmed with the amount of information that they have to go through every day. Classifying the text which users encounters into different buckets can give a boost to their efficiency as well as understanding. For instance, classification of news into different topic allows the users to only focus on the topics relevant to their interest.

The core importance of textual classification lies in finding an appropriate representation of text data where interesting metrics (as measurements) can be used in order to compare different text data in accordance to their similarity to extract insights.

Further, classification of different texts from public social media platform can be tremendously useful in identifying the nature of a post / tweet. For instance, classifying tweets into different hate-speech categories can be used to automatically remove the tweet from the user's handle.

The opportunities of text classification are endless. The interesting thing not only lies in just classifying texts into different buckets, but the analysis which can be on the basis of classification obtained. For instance, using the classifier to classify Political tweets can be used to identify politically vocal users and the models can further be improvised to provide analytical parameters for their political inclinations.

What is your planned approach ?

I plan to implement multiple models in order to classify textual data from publicly available datasets initially. A comparison study between the different models will allow to identify the most accurate model for a particular dataset.

As of now, I plan to follow the following pipeline to categorize textual data

1. Pre-process the input textual data
    a. Cleaning of text
    b. Stop Words removal
    c. Stemming
    d. Removal of non-alphabetic characters

2. Extract features from the pre-processed data
   I plan to implement and compare the following vector representations
    a. Count Vectors
    b. TF-IDF Vectors
        i. Bag of words (word level)
        ii. Bag of n-grams (n-gram level)
        iii. Character level

3. Training Model – Learning
    a. Naïve Bayes
    b. Linear Classification
    c. SVM
    d. Random Forest
    e. Convolutional Neural Network

4. Classification of text using the trained model

5. Evaluation and Comparison of different models

I plan to use the above approach for different datasets, namely, SMS Spam and hate-speech. Further, I will using the above approach on my newly generate dataset to classify political tweet in the Indian context.

What tools, systems or datasets are involved?

I will primarily be using Python. Within python, I plan to use the following libraries sklearn, seasborn, pandas, numpy, Spacy, nltk for getting stopwords corpus, tweepy etc.

Datasets Involved

- SMS Spam Collection
  http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/
  The SMS Spam Collection v.1 is a public set of SMS labeled messages that have been collected for mobile phone spam research. It has one collection composed by 5,574 English, real and non-enconded messages, tagged according being legitimate (ham) or spam.
  Labels : spam / ham

- Hate-speech and Offensive Language Dataset
  https://github.com/t-davidson/hate-speech-and-offensive-language/tree/master/data
  The data are stored as a CSV and as a pickled pandas dataframe (Python 2.7). Each data file contains 5 columns:
  count = number of CrowdFlower users who coded each tweet (min is 3, sometimes more users coded a tweet when judgments were determined to be unreliable by CF).
  hate_speech = number of CF users who judged the tweet to be hate speech.
  offensive_language = number of CF users who judged the tweet to be offensive.
  neither = number of CF users who judged the tweet to be neither offensive nor non-*class* = class label for majority of CF users. 0 - hate speech 1 - offensive language 2 – neither
  Labels : hate speech / offensive language / neither

- Political Tweets Dataset – Custom (India)
  I plan to obtain the relevant political tweet (Indian context) from the publicly available dataset on Kaggle
  https://www.kaggle.com/codesagar/indian-political-tweets-2019-feb-to-may-sample

I will label them as political. Further, I will collect non-political tweets from publicly available twitter datasets and label them as non-political.

What is the expected outcome?

I expect to accurately classify the textual data in their respective categories using the trained models. Further, I intend to write a script which can automatically collect tweets based on some parameters and accurately classify them into political and non-political tweets.

Inputs : Textual data from the respective datasets
Expected outcomes as per the models trained on different datasets
SMS Spam Data : spam / ham (Not spam)
Hate Speech and Offensive Data : hate speech / offensive language / neither
Political Tweets Dataset : political / non political

How are you going to evaluate your work?

I intend to evaluate the models by comparing their predictions using parameters like precision, recall, f1 scores etc. This evaluation will help identify the most suitable model for textual classification, depending on the datasets. At the time of training, the dataset will be divided into testing, training and validation sets.

Which programming language do you plan to use?

I plan to implement this project in python. Further, I am also planning to use Jupiter notebook for making the code more presentable and easy to demonstrate during the project presentation.

Justification of Work Load

S. No.	Task	Estimated Hours
1.	Import and Pre-process the input textual data a. Cleaning of text b. Stop Words removal c. Stemming d. Removal of non-alphabetic characters	2 hrs
2.	Extract features from the pre-processed data a. Count Vectors (0.5 hrs) b. TF-IDF Vectors (1.5 hrs) i. Bag of words (word level)	1.5hrs

	ii. Bag of n-grams (n-gram level) iii. Character level	
3.	Training Model – Learning a. Naïve Bayes (0.5 hrs) b. Linear Classification (0.5) c. SVM (1 hr) d. Random Forest (1 hr) e. Convolutional Neural Network (4 hrs)	7 hrs
4.	Classification of text using the trained model	2 hrs
5.	Evaluation and Comparison of different models	4 hrs
6.	Create new data set for classifying political tweets in India	3 hrs
7.	Fetch tweets using different parameters and classify as Political / Non Political	3 hrs

This above table only gives a rough time estimate of the tasks which will be involved in completing the project. It fulfils the 20+ hours workload as mentioned in the requirements.