**TOXIC COMMENT CLASSIFICATION**

* 1. **Introduction**

According to the father of Artificial Intelligence, John McCarthy, it is “The science and engineering of making intelligent machines, especially intelligent computer programs”.

Artificial Intelligence is a way of making a computer, a computer-controlled robot, or a software think intelligently, in the similar manner the intelligent humans think.

AI is accomplished by studying how human brain thinks, and how humans learn, decide, and work while trying to solve a problem, and then using the outcomes of this study as a basis of developing intelligent software and systems.

**Philosophy of AI**

While exploiting the power of the computer systems, the curiosity of human, lead him to wonder, “Can a machine think and behave like humans do?”

Thus, the development of AI started with the intention of creating similar intelligence in machines that we find and regard high in humans.

**Goals of AI**

To Create Expert Systems − The systems which exhibit intelligent behavior, learn, demonstrate, explain, and advice its users.

To Implement Human Intelligence in Machines − Creating systems that understand, think, learn, and behave like humans.

**What Contributes to AI?**

Artificial intelligence is a science and technology based on disciplines such as Computer Science, Biology, Psychology, Linguistics, Mathematics, and Engineering. A major thrust of AI is in the development of computer functions associated with human intelligence, such as reasoning, learning, and problem solving.

Natural Language Processing (NLP) refers to AI method of communicating with an intelligent systems using a natural language such as English.

Processing of Natural Language is required when you want an intelligent system like robot to perform as per your instructions, when you want to hear decision from a dialogue based clinical expert system, etc.

The field of NLP involves making computers to perform useful tasks with the natural languages humans use. The input and output of an NLP system can be −

1.Speech

2.Written Text

3.Components of NLP

There are two components of NLP as given −

**Natural Language Understanding (NLU)**

Understanding involves the following tasks −

1.Mapping the given input in natural language into useful representations.

2.Analyzing different aspects of the language.

**Natural Language Generation (NLG)**

It is the process of producing meaningful phrases and sentences in the form of natural language from some internal representation.

It involves −

1.Text planning − It includes retrieving the relevant content from knowledge base.

2.Sentence planning − It includes choosing required words, forming meaningful phrases, setting tone of the sentence.

3.Text Realization − It is mapping sentence plan into sentence structure

**Python Programming**

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

 Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

* 1. **Objectives of research**

The background for the problem originates from the multitude of online forums, where-in people participate actively and make comments.

To build a multi-headed model that’s capable of detecting different types of toxicity like threats, obscenity, insults, and identity-based hate better than Perspective’s [current models](https://github.com/conversationai/unintended-ml-bias-analysis).  A research area of focus is the study of negative online behaviours, like toxic comments (i.e. comments that are rude, disrespectful or otherwise likely to make someone leave a discussion). So far they’ve built a range of publicly available models served through the [Perspective API](https://perspectiveapi.com/), including toxicity.

But the current models still make errors, and they don’t allow users to select which types of toxicity they’re interested in finding (e.g. some platforms may be fine with profanity, but not with other types of toxic content).

**1.3 Problem Statement**

Given a group of sentences or paragraphs, used as a comment by a user in an online platform, classify it to belong to one or more of the following categories — toxic, severe-toxic, obscene, threat, insult or identity-hatewith either approximate probabilities or discrete values (0/1).

As the comments sometimes may be abusive, insulting or even hate-based, it becomes the responsibility of the hosting organizations to ensure that these conversations are not of negative type.

The task was thus to build a model which could make prediction to classify the comments into various categories like toxic, severe-toxic, obscene, threat, insult or identity-hate.

**2. Review of Literature**

This is the existing solution overview to the [toxic comment classification](https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge) . The topic is to build a multi-headed model that is capable of detecting different types of toxicity like threats, obscenity, insult.

Solution is based on the following 3 parts:

1. Pre-processing
2. Models
3. Ensemble

**Pre-processing**

Importing Libraries

We are going to use below three most essential libraries for data pre-processing:

[numpy](http://www.numpy.org/) : numpy is a library which supports mutli-dimensional arrays in Python

[matplotlib](https://matplotlib.org/) : matplotlib is used for plotting charts in Python

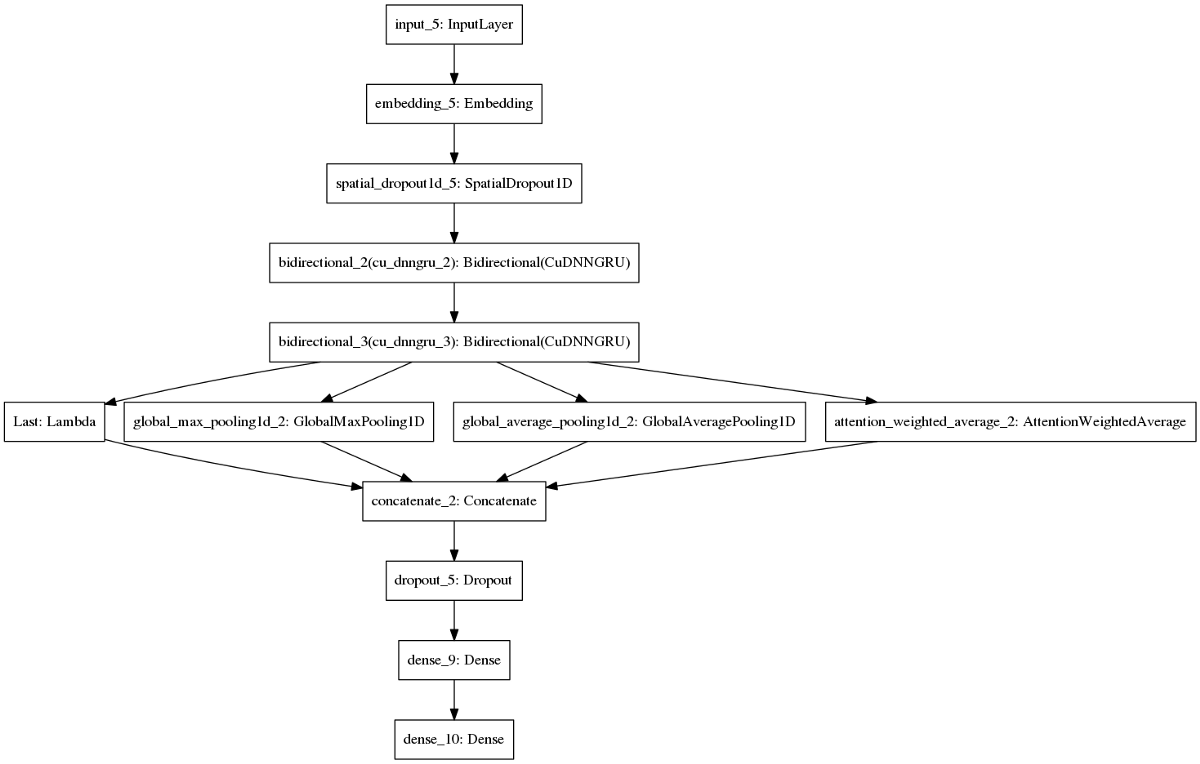
[pandas](https://pandas.pydata.org/pandas-docs/stable/) : pandas provides expressive data structures for relational data

Trained our models on 3 datasets with different pre-processing:

* original dataset with spellings correction: by comparing the Levenshtein distance and lots of regular expressions
* original dataset with post taggings: We generate the part of speech (POS) tagging for every comment by TextBlob and concatenate the word embedding and POS embedding as a single one. Since TextBlob drops some tokens and punctuations when generating the POS sequences, that gives our models another view.
* original dataset with very heavily data-cleaning (drop html/css and wikipedia template), spelling correction and non-English comment translation

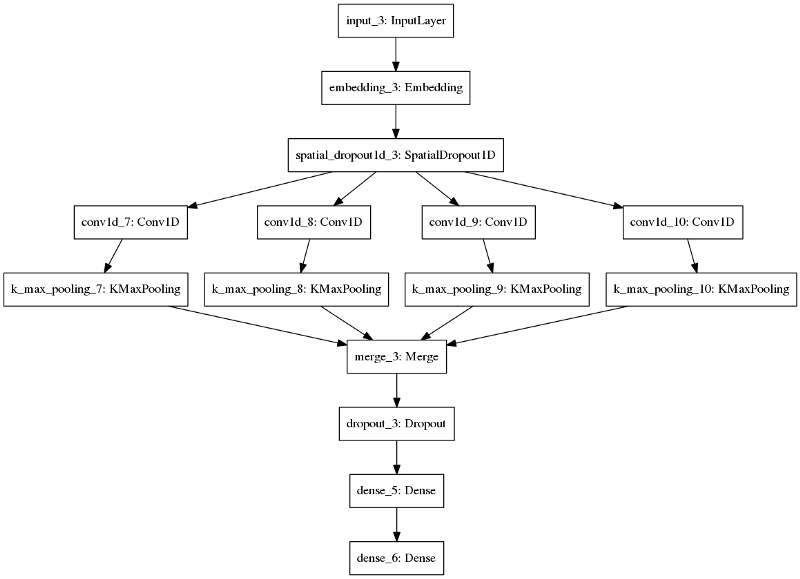
### Models

simple models always perform better. some complicated structures (RHN with recurrent dropout, DPCNN, VDCNN, HAN, Convolutional Attention Model). Most of them had performed very well locally but got lower AUC on the leader board. The models during the final stage are the following two:



It is a very classical Text-RNN structure with some tricks. I applied 4 pooling layers to summarize the text sequences:

1. Lambda layer, return the last hidden state
2. Global max pooling
3. Global average pooling
4. Dot attention



**3. Data Collection**

To categorize the toxic comments based on the types of toxicity. Examples of toxicity types can be toxic, severely toxic, obscene, threat, insult, identity hate. Different machine learning techniques like Logistic Regression, Support Vector Machines and Naive Bayes are implemented to determine the 6 types of toxic comments.

**Data**: The dataset we are using for toxic comment classification is taken from Kaggle competition which can be found at Kaggle. Dataset has a large number of comments from Wikipedia talk page edits. They have been labeled by human raters for toxic behavior.

| **toxic** | **severe\_toxic** | **obscene** | **threat** | **insult** | **identity-hate** |
| --- | --- | --- | --- | --- | --- |
| Count | 15294 | 1595 | 8449 | 478 | 7877 | 1405 |
| Percentage | 9.5% | 0.9% | 5.2% | 0.2% | 4.9% | 0.8% |

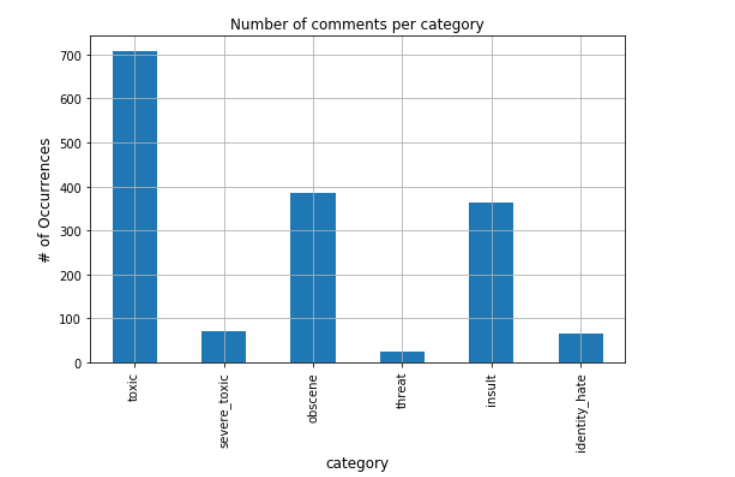
For this problem while under-sampling does not work well, higher-weight and over-sampling give us better results.

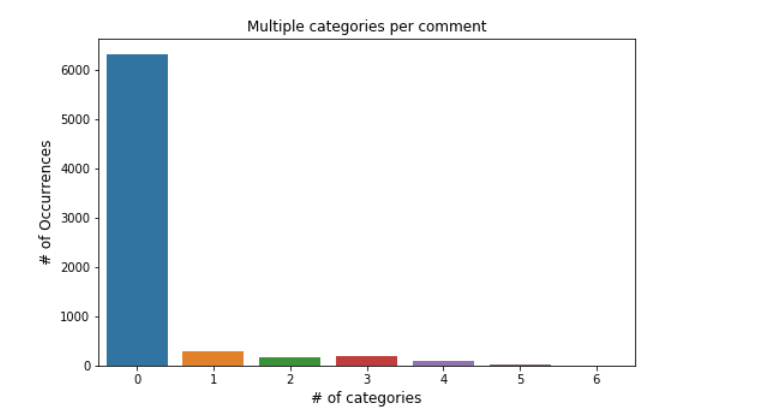
| **Label** | **Model** | **F1-score for class "1"** | **Method** |
| --- | --- | --- | --- |
| Toxic | Neural Network | 0.69 | higher weight |
| Severe Toxic | Naïve Bayes | 0.64 | over sampling |
| Obscene | Neural Network | 0.70 | higher weight |
| Threat | Neural Network | 0.23 | higher weight |
| Insult | Neural Network | 0.59 | higher weight |
| Identity Hate | Neural Network | 0.28 | higher weight |

**4 Methodology**

**4.1 Exploratory Data Analysis**

**4.1.1 Figures and Tables**





**4.2 Data Modelling**

To implement this classification we use the Natural Language Processing Algorithm

**NLP ALGORITHM:**

Nat­ur­al Lan­guage Pro­cessing is a field that cov­ers com­puter un­der­stand­ing and ma­nip­u­la­tion of hu­man lan­guage, and it’s ripe with pos­sib­il­it­ies for news ­gath­er­ing

NLP is a way for computers to analyze, understand, and derive meaning from human language in a smart and useful way. By utilizing NLP, developers can organize and structure knowledge to perform tasks such as automatic summarization, translation, named entity recognition, relationship extraction, sentiment analysis, speech recognition, and topic segmentation.

NLP algorithms are typically based on machine learning algorithms. Instead of hand-coding large sets of rules, NLP can rely on machine learning to automatically learn these rules by analyzing a set of examples (i.e. a large corpus, like a book, down to a collection of sentences), and making a statistical inference. In general, the more data analyzed, the more accurate the model will be.

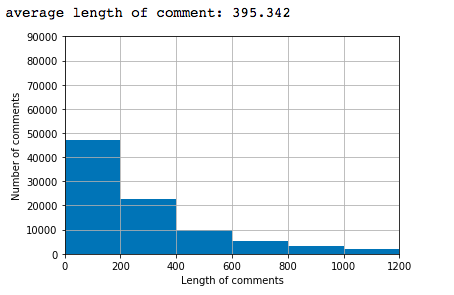
In multi-label classification, data can belong to more than one label simultaneously. For example, in our case a comment may be toxic, obscene and insulting at the same time. It may also happen that the comment is non-toxic and hence does not belong to any of the six labels.

**Studying data & identifying hidden patterns**

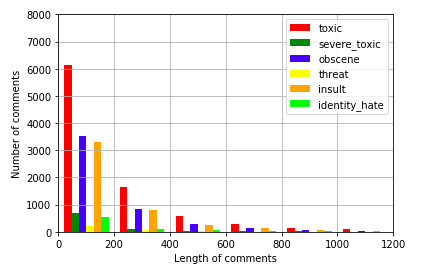
I had a dataset of 159572 samples of comments along with their labels. I observed that every 1 in 10 samples was toxic, every 1 in 50 samples was obscene and insulting, but the occurrences of sample being severe-toxic, threat and identity hate was extremely rare.

**creating some visualizations:**

Keeping length of comments on the independent axis with buckets of size 200, I counted the number of comments which had number of characters in that range. For instance, from the graph we can see there are 10000 comments which have 400 to 600 characters.

****

For the next visualisation, I had length of comments on the independent axis again similar to the previous plot. But instead of counting number of comments, I counted comments belonging to each of the different categories.



The second visualisation plots the number of comments belonging to various categories. Toxic comments were highest in number, followed by obscene, insult, severe-toxic, identity-hate and threat in decreasing order.

**Partitioning into testing and training**

Here X is the independent data and y is the dependent data

We partition independent and dependent data. In X the

comments are present which are independent and the y

variable contains the binary values of different categories.

from sklearn.feature\_extraction.text import CountVectorizer

cv = CountVectorizer(max\_features=1500)

X = cv.fit\_transform(data).toarray()

y=dataset.iloc[:,2:].values

Test and Train sets are partitioned as follows. In Cross-Validation the Train-Test-Split method is used.

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.3, random\_state = 0)

**Data Preprocessing**

import re

import nltk

#nltk.download('stopwords')

from nltk.corpus import stopwords

from nltk.stem.porter import PorterStemmer

ps = PorterStemmer()

data=[]

for i in range(0,len(dataset)):

review=dataset["comment\_text"][i]

review = re.sub('[^a-zA-Z]', ' ', review)

review = review.lower()

review = review.split()

review = [ps.stem(word) for word in review if not word in set(stopwords.words('english'))]

review = ' '.join(review)

**5. Findings and suggestions**

We get some help by using the following sources:

<https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge>

<https://medium.com/@nupurbaghel/toxic-comment-classification-f6e075c3487a>

<https://towardsdatascience.com/multi-label-text-classification-with-scikit-learn-30714b7819c5>

**6.Conclusion:**

Using the developed model we can predict and classify the comments into 6 categories namely, Toxic, Severe-Toxic, Obscene, Insult, Threat, Identity-hate.

Our model is developed using Natural Language Processing where it automatically classifies the data using training and test splitting and optimizer techniques which is adam

Since, many users are limiting their social communications because the threat on various online platforms, we can help them by taking certain action on the netizens who are making social media an abusive platform so that, people can use the social media without any misery.

We can also put our model into next phase by helping the cyber security workers to take severe action on the people with their unique ids like ip address or some other identities.