Identifying and Classifying Toxic Comments on the Internet

**Introduction to Problem:**

Being anonymous over the internet can sometimes make people say nasty things that they normally would not in real life. Let's filter out the hate from our platforms one comment at a time.

**Objective:**

To use Natural Language Processing (NLP) in Python to identify and classify toxic comments on the internet. The ability to recognize toxic comments on the internet would be of use to websites where comments are collected from the public. Therefore, the client would include online business directories such as Yelp and AirBnB and other directories where people provide comments for other users. By identifying the toxic comments, a decision can be made as to whether to delete harmful comment, block the user, etc.

**Data Overview:**

The dataset is from wiki corpus dataset which was rated by human raters for toxicity. The corpus contains 63M comments from discussions relating to user pages and articles dating from 2004-2015.

Different platforms/sites have different standards for their toxic screening process. For this project the comments are tagged in the following five categories:

* toxic
* severe\_toxic
* obscene
* threat
* insult
* identity\_hate

The tagging was done via crowdsourcing**.** The dataset was rated by different people and the tagging might not be 100% accurate. The same concern is being discussed [here](https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/discussion/46131): https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/discussion/46131

The [source paper](https://arxiv.org/pdf/1610.08914.pdf) (<https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/discussion/46131> ) contains more interesting details about the dataset creation.

**Approach:**

The approach for the problem will be as follows:

* Load data set, i.e. the training data and the test data
* Preprocessing
  + Remove stop words using Natural Language Tool Kit (NLTK)
  + Perform bigram collocation – words that often appear consecutively
  + Perform stemming and lemmatization - reduce words to their root
* Use algorithm for text classification – create a list of words in the cleaned text
* Perform grid search to tune and optimize parameters
* Visualize – create a word cloud

This approach is subject to change once the data has been uploaded and explored.