## Importing Libraries

Firstly necessary libraries are imported like numpy, pandas, spacy and others.

## Reading Dataset

* Then read the train and test csv files in python notebook with the help of pandas libraries.

## Texts Processing

* To remove any newline characters from the comments I used the lambda function to replaces all occurrences of the newline character ('\n') with a space (' ').
* Then I defined several tects Categories - 'toxic', 'severe\_toxic', 'obscene', 'threat', 'insult', 'identity\_hate'. These labels correspond to different types of toxic or offensive language that the model will be trained to detect.
* I created a empty list called 'train\_prepared\_data',
* Next, a function called 'format\_text\_spacy' is defined, which takes in a Spacy 'text' object as its argument. The function returns a tuple containing the text string itself and a dictionary called 'cats', which contains the categorical labels associated with the text (e.g., whether the text is toxic, severe\_toxic, obscene, etc.).
* The for loop iterates over the rows of 'train\_data' and retrieves the text data for each row using the 'iloc' method. The 'format\_text\_spacy' function is then applied to each text object, and the resulting tuple is appended to the 'train\_prepared\_data' list.

By the end of the loop, 'train\_prepared\_data' will contain a list of tuples, with each tuple representing a processed text sample along with its associated categorical labels.

This data can then be used to train a Spacy model for text classification or other NLP tasks.

* Then I showed categories of some of the 'train\_prepared\_data'.
* Next I created a new, blank Spacy pipeline using the 'blank' method from the Spacy library. Then created a new pipe in the pipeline for multi-label text classification, and assigns it to a variable called 'textcat'. The 'textcat\_multilabel' pipe is used as there are multiple labels to be assigned to a single text sample. Then labels for the different categories are added

## ****Training and Testing Data****

* I used 'train\_prepared\_data' to train the text classification model.
* only the 'textcat\_multilabel' pipe is trained during the training process.
* I choose 10 epochs (i.e., complete passes through the training data) to train the model. For each epoch, the it initializes an empty dictionary to store the training loss for that epoch.
* The training data is then divided into batches using the 'minibatch' method from Spacy and ‘compounding’ gradually increasing the batch size over time.
* Finally I print the training loss for the current epoch This process repeats for each epoch until training is complete.
* Finally I tested my model on test dataset.