# *Toxic Comments Classification Challenge*
# Machine Learning for Natural Language Processing 2021

**Lilia BEN BACCAR**
ENSAE, MS Data Science
lilia.benbaccar@ensae.fr

**Sarah LAUZERAL**
ENSAE, MS Data Science
sarah.lauzeral@ensae.fr

## Abstract

We address the problem of online abuse and harassment by building a model capable of detecting in comments different types of toxicity. Our approach is: (1) to compare 3 types of word embeddings with the same single-label neural model; (2) to compare two multi-label models using LSTM and BERT. Code available here: Github[1] and Google Colab[2].

## 1 Problem Framing

### 1.1 Context and motivation

With social media we can communicate quickly and easily, share experiences, our opinions and beliefs. Our conversations and comments can be closely targeted or widely broadcast to the point that they can go viral. It is also widely used by abusers that 'hide' behind the fact that they are anonymous. The threat of abuse and harassment online means that many people stop expressing themselves and give up on seeking different opinions. Platforms struggle to effectively facilitate conversations, leading many communities to limit or shut down user comments.

### 1.2 A Kaggle Challenge

Jigsaw and Google have founded Conversation AI team, a research initiative working on tools to help improve online conversation. One area of focus is the study of negative online behaviors. They've built a range of models and decided to create a Kaggle challenge[3] to help them improve the current models. We have to build a model that's capable of detecting different types of toxicity like threats, insults and identity-based hate to help online discussion become more productive and respectful.

### 1.3 Dataset

We have at our disposal a dataset of comments from Wikipedia's talk page edits. Each of these comments have been labeled by human raters. We used the train file which has 159 571 comments and 8 columns that are: comment id, comment text and toxic, severe_toxic, obscene, threat, insult, identity_hate (types of toxicity). A comment can belong to one or more classes and all comments are in english.

## 2 Experiments Protocol

### 2.1 Data cleaning

Due to the diversity of capitalization in a sentence, the first step is to put all the comments in lowercase. When we talk online, we rely on abbreviations so we decided to remove contractions. Then because text data could include various unnecessary characters, we decided to remove URLs, HTML tags, non-ASCII characters, emojis and punctuations. Finally, as stop words do not deliver meaningful information, we decided to remove them.

### 2.2 Baseline model

We build a baseline model to get a quick performance benchmark. When trying to classify natural language, logistic regressions usually give quick and solid results. So we used the standard vector space representation using TF-IDF weighting and fit a logistic regression to build single-label models. Indeed, because each comment can have multiple labels, as a baseline model, we decided to use 6 different binary classifiers for each toxicity type.

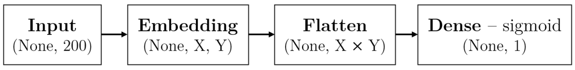
[1] https://github.com/lbenbaccar/Toxic-Comment-Classification-Challenge

[2] https://rb.gy/gz5nhi

[3] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

### 2.3 Single task models

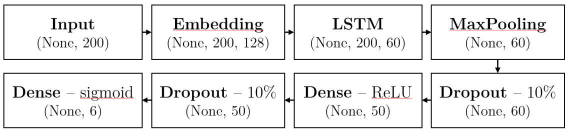
We used different text representation methods to compare them with the same neural model. First, we represented comments using the following word embeddings: (1) word2vec (provides direct access to vector representations of words as a combination of CBOW & Skip-gram model), (2) GloVe (one of the newest methods for calculating the vector representation of words), and (3) One-Hot (representation of categorical variables as binary vectors). We used the following model architecture:



### 2.4 Multi tasks models

#### 2.4.1 LSTM

We fed the comments into an LSTM as part of a neural network. As pre-processing steps, we did the following steps : tokenization (break down sentences into unique words), indexing (put the words in a dictionary and give them an index each), index representation (represent sequences of words in the comments in the form of index) and padding (max length = 200 to have inputs of the same size). We used the following model architecture:



#### 2.4.2 BERT

As pre-processing, we did the following steps : tokenization, truncation and padding. We also created efficient data pipelines using tf.data. We loaded the pretrained BERT base-model, took the first hidden-state from BERT output (CLS token) and fed it into a Dense layer with 6 neurons and sigmoid activation function so that the outputs of this layer will be probabilities for each of the 6 toxicity types. To train our model, we used the BinaryCrossentropy as our loss function that is calculated for each of the 6 output neurons, Adam as our optimizer and AUC as our evaluation metrics.

## 3 Results

### 3.1 Qualitative analysis

We tried to understand to what type of sentiment all the classes refers to. We plotted the more common words belonging to each class. It appears that ”fuck” is the word that appears the most in the classes ”toxic”, ”severe_toxic”, ”obscene” and ”insult”. The class ”identity_hate” is firstly against the black community, then jews, gays and Mexicans. The comments flagged as ”threat” mainly regroups death threat but also threat to ”block” someone on social media, this is a very specific toxicity type. ”Toxic”, ”obscene” and ”insult” classes regroup the majority of comments. Half of the severe toxic comments are also toxic. What differentiates both categories, is that the severe toxic comments include a sexual component in it and are highly degrading.

### 3.2 Single task models

| F1-SCORE | **OneHot** | **word2vec** | **GloVe** |
|---|---|---|---|
| ***toxic*** | 28% | 64% | 70% |
| ***severe_toxic*** | 30% | 32% | 65% |
| ***obscene*** | 31% | 66% | 73% |
| ***threat*** | 5% | 25% | 74% |
| ***insult*** | 32% | 55% | 65% |
| ***identity_hate*** | 5% | 27% | 63% |

### 3.3 Multi tasks models

| | **LSTM** | **BERT** |
|---|---|---|
| ***toxic*** | 90.4% | 99.0% |
| ***severe_toxic*** | 99.0% | 99.1% |
| ***obscene*** | 94.7% | 99.4% |
| ***threat*** | 99.7% | 97.3% |
| ***insult*** | 95.1% | 99.0% |
| ***identity_hate*** | 99.1% | 98.7% |

Table 1: Accuracy for LSTM and AUC score for BERT

## 4 Discussion / Conclusion

To compare single task models' embeddings, we fixed the neural networks architecture but the models performance have to be studied carefully since each embedding technique could require a different number of epochs to give similar results so we can't give an absolute hierarchy of embedding performances. We could also predict to which classes a comment would belong to by implementing 6 different single tasks models so that each model maximizes the $f_1$ score of the predicted class. Thus, 6 single task models could possibly have competitive results with those obtained by the multitasks model.