# Capstone Proposal:<br>Comment toxicity classification using Neural Networks and Machine Learning techniques

Fabio STEFFENINO

July 3, 2019

The argument of the research I would like to tackle and develop as final capstone project for my Machine Learning Engineer Nanodegree, it is related to the usage of Neural Networks and NLP techniques in comment toxicity classification. As a future machine learning engineer, I am really interested in understanding how NLP works and how to solve related problems, as a father, I am concerned about online comment cruelty and cyber-bullying. This will be my first approach to NLP problems and my goal is, first, to learn how to design a Neural Network for text processing and, second, to apply what I learned to the problem I will highlight hereafter.

**Background** The online world has become entirely part of our lives. One of the biggest problem of the cyberspace is the one identified with the term Online Disinhibition Effect, in other words the lack of restraint one person feels when communicating online in comparison to communicating in-person (Wikipedia definition). There are two Online Disinhibition categories: Benign and Toxic. The Toxic Online Disinhibition represent a situation where people, with the help on anonymity in certain cases, have an inappropriate behaviour on platforms like blogs or social networks where they feel free to comment using hostile or derisive language. Researchers spent a lot of time in the last years in Sentiment Analysis and Comment Toxicity classification and Natural language Processing with Neural Networks is the most used technique.

**Problem Statement** Google and Jigsaw co-founded a research initiative, called Conversation AI, that is working on tools to help improve online conversation. One area of focus is the one highlighted before: the study of online negative behaviour and toxic comment identification. In 2017 they release a free tool, Perspective API, that use machine learning to score the perceived impact a comment might have on a conversation. Their first version of the model identifies whether it could be perceived as "toxic" or not to a discussion. In order to improve their model performances, they hosted a Kaggle competition detailed at Toxic comment classification challenge. The scope of this competition is to build a multi-headed model that is capable of detecting different types of of toxicity like *threats, obscenity, insults, and identity-based hate* better than Perspective's current models and providing for each comment the probability that it falls under each class(*Multi-Label classification*).

**Datasets and Inputs** The dataset called "Wikipedia Talk Page Comments annotated with toxicity reasons" has been provided on Kaggle and can be found at the following link: Toxic comment classification challenge dataset. Both Train and Test dataset are CSV files containing approximately 160,000 comments each. They have been all manually labeled by human raters for toxic behaviour. The given toxicity categories are the following:

- Toxic
- `Severe_Toxic`
- Obscene
- Threat
- Insult
- `Identity_hate`

Train dataset has 8 columns: `id`, `comment_text` and one column for each category with value 0;1 (1 means that the comment falls under that category, each comment can be part of more than one category). Test dataset has 2 columns: `id` and `comment_text`. The labels for the test dataset have been provided separately in a file called `test_labels`.

*Disclaimer: the dataset for this competition contains text that may be considered profane, vulgar, or offensive.*

**Solution Statement** We do have a lot of available solutions already explored out there: From Naive Bayes, Logistic Regression, SVM(Support Vector Machine), MLP(Multi Layer Perceptron) and CNN(Convolutional Neural networks). The goal of my project is to demonstrate that RNN(Recurrent Neural Networks) can be a particular optimal candidate model for solving this kind of problems. In particular I will also explore the different RNN core architectures like LSTM(Long Short-Term Memory) or GRU(Gated Recurrent Unit) and the different methods of vectorizing the input comment text. The output will be, for a given comment, the probability that it falls under each single category.

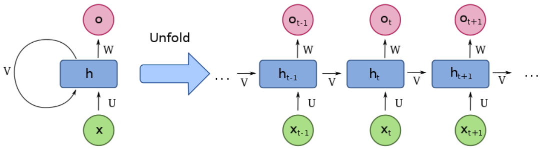

Figure 1: RNN generic architecture, Wikipedia

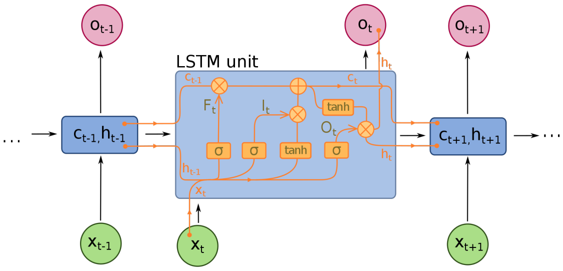

Figure 2: LSTM unit representation, Wikipedia

**Evaluation Metrics** The initial metric proposed by Kaggle was the `LogLoss` function but it has been later changed into `AUC-ROC`. So we will be using this last one in order to evaluate our model. `AUC-ROC` is the Area Under the Receiver Operating Characteristics Curve. It is a particular good metrics because it tells how much the model is capable of distinguish between the different classes. In a multi-label classification problem the global AUC-ROC score is calculated by taking the average of the individual AUCs of each predicted category column. AUC can have value between 0 and 1. A model with 100% of correct prediction has a AUC score value of 1, a model with 100% of wrong prediction has a AUC score value of 0.

**Benchmark Model** A first benchmark score I will be using to evaluate my model is the current competition leaderboard. I would like to obtain a score that could possibly place my model on the top 100 (ROC AUC greater or equal to 0.9874). Additionally, I would like to select one of the available solution present on the net and compare it with my RNN solution. (a possible candidate is the following logistic regression model )

**Project Design** I do not have yet in mind how i will design my pipeline and solve the problem in details but i will for sure follow the steps highlighted hereafter:

- EDA (Exploratory data analysis)
- Feature engineering
- Explore text to vector representations (Word2Vec/Glove)
- Prepare and split data for training validation and testing
- Prepare benchmark models
- Explore Neural Networks for Natural Language Processing (RNN, LSTM, GRU)
- Explore and apply different parameters in order to improve the model
- Classification of Test dataset and result comparison with benchmarks models
- Report preparation

I would like my project to be not only solution but also education oriented. That means, I will use this project to introduce myself to Natural Language Processing methods and so I will also include in the report some theoretical notes.