Initial Plan- Abusive language detection against immigrants and women

Author: Ross Singleton
Supervisor: Luis Espinosa-Anke
Moderator: David J Humphreys
CM3202 One Semester Individual Project – 40 Credits

Project Description

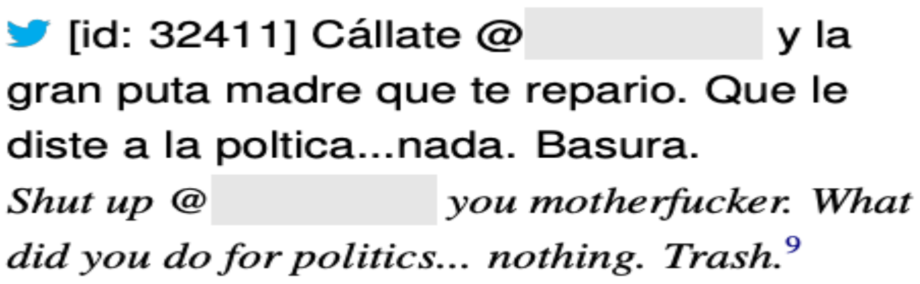
In this proposal a student will explore computational approaches for modelling and detecting biased and abusive language in social media. There are two broad topics: (1) Offensive language detection, which occurs when individuals take advantage of the perceived anonymity of computer-mediated communication and engage in behaviour that many of them would not consider in real life [1]; and (2) Hate Speech detection. Hate Speech is commonly defined as any communication that disparages a person or a group on the basis of some characteristic such as race, colour, ethnicity, gender, sexual orientation, nationality, religion, or other characteristics [2].

Both these types of abusive language are pervasive in social media. Projects will be carried out in a controlled environment, i.e., they will use the datasets provided in two current data science competitions (see references). The main goal of a project developed within this proposal is to build a system that works (i.e., which produces output that can be evaluated with the official scorer script provided in the task), to be able to describe the intuition behind the design of the model, and finally how these intuitions made it into the final model/code. Then, this system can be compared in a real-life scenario with the baseline system provided by the organizers of each competition, and with other systems submitted by research groups and companies.

A student may choose to develop a system for either of these competitions, or for both.

[1] https://competitions.codalab.org/competitions/20011
[2] https://competitions.codalab.org/competitions/19935

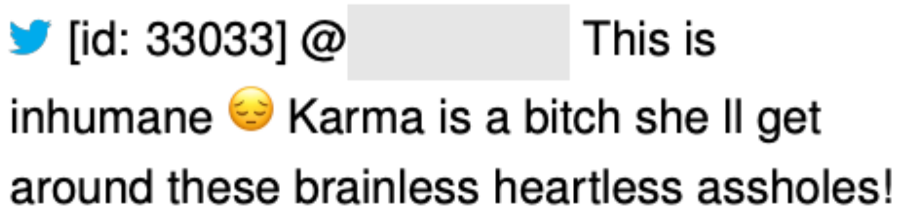
I have chosen to work on Sub Task A of the first project and build a system to process and normalise the data, then use a classifier to make predictions as to if the tweets are hateful or not hateful to women and immigrants. Once this is done, I will evaluate the performance of my classifier in comparison to the correct evaluations of the tweet, using a variety of evaluation metrics. Following on from this, I will perform some error analysis on the predictions that were produced by my classifier and attempt to understand and improve the classifier in terms of what it can and can’t detect in the data set. Please see the figures below as examples of tweets that are hateful and not hateful. These were provided in the original task summary publication.



[id: 32411] Cállate @[redacted] y la gran puta madre que te repario. Que le diste a la poltica...nada. Basura.

Shut up @[redacted] you motherfucker. What did you do for politics... nothing. Trash.[9]

Figure 1 - Hateful Tweet



[id: 33033] @[redacted] This is inhumane 😔 Karma is a bitch she ll get around these brainless heartless assholes!

Figure 2 - Non-Hateful Tweet

Project Aims and Objectives

- A method of processing and normalizing the provided data set in preparation for input into the classifier.
  - Data will need to be formatted for input into the classifier, for example being turned into numerical values for some methods of classification.
  - The data may also need to be all put into lower case for example.
- A classifier for generating predictions of hateful and non-hateful based on the data set fed into it.
  - The classifier will take the form of a machine learning algorithm, most likely making use of the scikit-learn library in Python.
  - I plan on using a decision tree as my chosen method of generating predictions from the data set.
- A set of predictions from the classifier, organised in such a way that performance analysis and error analysis can be performed on them.
  - These will be outputted in a uniform way, as a list of binary with 1 for hateful and 0 for non-hateful.
  - This is done in order to be in the correct format to compare to the correct detections.
- Performance evaluation of my predictions compared to the correct detections of abuse, based on certain evaluation metrics.
  - The correct detections of abuse are provided as part of the data set and will play a key role in the analysis and subsequent improvement of my model.
  - This analysis will cover metrics like accuracy, precision, recall, and speed.
- Error analysis of my predictions.

    - There could be things like irony that my classifier might struggle to pick up on, and this analysis will point out areas where the classifier can be improved to detect more abuse.

Work Plan

I intend to work in an Agile way, making use of a Kanban board to structure the work in such a way that the highest priority tasks are completed first. I have chosen to work in this way as I believe that it is the best way of visualising work, allowing me to see what needs to be done first and what tasks are blocking each other. As well as this, working in Kanban will allow me to make changes to the tasks, either expanding or reducing the scope of the project, quickly and easily if required.

I also intend to meet with my supervisor once a week, in order to gain feedback quickly on the work I have completed in the intervening time, allowing me to make changes if required. This will work well in conjunction with my plan to work with a Kanban board to structure the project. Please see the breakdown of deliverables week by week throughout the project:

- Week 1: Initial Report
- Week 3: Method for Data Processing and Normalizing
- Week 5: Working Classifier
- Week 6: First Review Meeting
- Week 7: Performance Analysis and Evaluation Metrics
- Week 9: Error Analysis
- Week 11: Second Review Meeting
- Week 12: Final Report

The above time frames are all based on what I think is the likely amount of time that they will each take. Obviously, there is potential for some of these tasks to take longer than expected, due to the risky and unknown nature of some of them. For example, designing and building a working classifier may end up taking longer due to unknowns such as how long it will take me to learn and properly make use of libraries that are completely new to me. For example, if in the event I cannot complete my plan and get the classifier working, I have a backup plan of making use of much less complex “if else” statements to generate predictions that I can complete the rest of the project with. This would not change the scope of the project overall and would still allow me to perform performance and error analysis.

Another potentially risky task in the project is the error analysis. Although I think this task is at less risk of running over or not being completed than developing the classifier, I think there is still a risk that I may struggle to analyse why tweets are not being characterised as offensive by the classifier when they are offensive. I may also not be able to suggest any way to improve the classifier, which would limit the usefulness of the error analysis. In the event this occurs, I think that the best course of action would be to leave the error analysis out of the project. This is because the error analysis task is going to be done near the end of the project, and I think I will be able to produce a higher quality report by focussing my time on writing the report as opposed to getting the error analysis correct.