Final Project Draft Report

Hayato Ishida

February 14, 2022

1 Introduction

These days, many human-to-human interactions are taking place on online platforms, mainly in social media. Many comments are being exchanged on a wide range of topics within social media, but those are not necessary all using appropriate languages [1]. Some of the contents may include offensive or hateful language, and exposure to those contents can affect users' mental health well-being [1].

In recent years, there has been increasing attention towards violence on social media platforms in the general public. In Japan, young TV star suicide was caused by offensive and violent comments targeting a specific person [2]. Following that incident, there was lots of focus on online insulting with offensive or abusive language. The Japanese government considers stricter law enforcement against those insulting on social media platforms [3]. COVID-19 pandemic was declared by WHO in March of 2020 [4]. Since then, many countries have placed various major restrictions to fight this disease, including social distancing and lockdowns. Social media plays a vital role in connecting people during that time. Although it brought lots of benefits to our society, there is also a dark side [5]. A study [6] shows an increase in abusive content or comments on Twitter and Reddit in several countries, specifically countries with strong restrictions.

A system for detecting offensive, abusive and hateful content attracting more interest from social media providers and regulators [7]. Specifically, there is growing concern about the effect of those content on the mental health of younger generations [6]. At the same time, due to ambiguous and diverse definitions of offensive content, it is a challenging topic for the research field of NLP and machine learning [7].

This project attempts to develop a machine learning model that has overcome several challenges: classifying diverse offensive content and short text. The model should return a degree of offensiveness in numerical value instead of a specific class. This model can be a solution to overcome the ambiguity and diverseness of offensive content. Also, it will use genetic algorithms to automate the optimization process of neural network architecture [8]. Genetic algorithm is a searching algorithm inspired by the theory of evolution developed by Charles Darwin in 1859. With this algorithm, the optimized neural network possibly developed automatically.

This project is based on the final project template of Machine Learning and Neural Network: deep learning on a public dataset. In addition, the project is partially inspired by the Artificial Intelligence project template: automated design using

evolutionary computation. Also, include knowledge and programming techniaues of Natural Language Processing.

2 Literatures Review

Since this is a well-established topic of research in Machine Learning and Natural Language Processing, several studies are related to offensive language detection.

Malmasi Shervin and Zampieri Marcos [9] demonstrate a text classification model to classify hate speeches in social media. This study developed a supervised classification model using a publicly published dataset of tweets from Twitter. The study used surface n-gram and word skip grams features, which have proven to work well with this task. The study uses support vector machines with LIBLINEAR package for a classifier model because of their high performance with similar word classification. The package, However, the study also demonstrates some problems with hate speech classification. First of all, each tweet of the dataset was labeled by individuals into three different classes: Hate, offensive, and ok. There is some ambiguity of definitions between 'hate' and 'offensive.' In the usual sense, different individuals would get different impressions from the same content. For example, with the same violent tweet, some might think it is 'offensive,' but others might think it is instead 'hate.' The result of this study exactly shows the ambiguity of those class definitions. The classification model successfully distinguishes the 'ok' tweet from the other two. However, the model performance was not good enough to distinguish between 'hate' and 'offensive'; also, some 'hate' or 'offensive' tweets are classified as 'ok.' This result demonstrates that there is an overlap of definition between those three classes because of ambiguity of language. The study shows that this is not practical to use specific classes to detect any hate or offensive content on social media platforms. It is necessary to overcome that ambiguity of language to build a practical and effective text classification model.

The study by Rishav Hada et al. [1] also points out those ambiguous definitions of different types of offensive languages. This study is similar to Malmasi Shervin and Zampieri Marcos [9] but approaches the problem differently. Classifying offensive language is quite complicated because there are many possible classes such as racist, sexist, hate speech, offensive, hateful, etc. In the real world, those classes can be ambiguous and overlap. In addition to those ambiguities, swear words are another problem that offensive text classification must consider. It is easy to classify them as offensive, but text with swear words does not necessarily mean offensive. For instance, 'Hell yes, and 'sure as hell love it' are not offensive. Those comments use swear words to express their feelings. However, those words are surely inappropriate languages. This study overcomes the ambiguity of offensive words by giving each data the best-worst scaling between -1 and 1, where -1 is maximally supportive, and 1 is maximally offensive. This approach allows classification tasks to be less ambiguous. Although each person will take an offensive text differently, it will not cause different classes to overlap. The study consider three different computational models: Bidirectional LSTM [10], BERT [11], and HateBERT [12]. As a result, HateBERT performs the best, and considering its publication date, it is a good result.

The previous two studies give a clear insight into distinguishing between different types of of-

fensive languages on social media or the internet platforms. The following study shows the negative effects of offensive words on young people.

The paper by Shi Xiaoqin, Yu Chao, and Wu Dongmei [13] study how those violent languages on the Internet affect young students' mental health. Most young students are users of the Internet, and it is hard to find a young student who does not have a social media account. The study focuses on young students because they are most vulnerable to that language violence on the Internet. Young people are in the middle of physical and mental development, and the study concern the negative effect of language violence during that development. A survey to each student collects data. The study shows that there are many different negative effects on young students' mental health, and an attempt of psychological intervention to improve their mental health did not bring significant positive results. All of those negative psychological effects lead to various problems for young students. It can cause problems in daily life activities such as sleeping, eating, and interpersonal relationships. It can also cause mental health-related problems, including feelings, emotions, and consciousness. Those problems can lead to other problems for young students, as well. To improve those psychological problems, the improvement of online platforms is necessary. Unfortunately, some online platforms ignore those language violences and gain some attention to maintain or increase traffic and for another benefit. This leads to more violence. Adults are likely to have the capability to avoid those contents or deal with those contents. However, young people are not capable of dealing with those content. Therefore, some sort of protection is necessary for them. Although the data sample of this study is small, it still shows the significance of those adverse effects caused by language violence on the Internet. An efficient violent language detecting system is the possible solution for this issue.

The study by Rishav Hada et al. [1] proposed some efficient text classification methods to detect offensive language. The study by Andersen Hayden et al. [8] shows the possibility of using a genetic algorithm to construct an efficient text classification model automatically. Genetic algorithm is a bio-inspired searching algorithm that can bring solutions beyond our imagination [14]. One of the main issues in developing text classification is the requirement of knowledge. Convolutional Neural Network is known for its high performance in text classification tasks. To develop text classification with a Convolutional Neural Network, deep knowledge of natural language processing and deep learning, a combination of machine learning and neural networks, is necessary. Also, constructing this type of text classification model is not easy to do in general. Considering those challenges, this study attempt to automate the construction process by genetic algorithm. Although it is still an initial stage of this area of research, this study shows various possibilities of applying genetic algorithms to text classification tasks. The study presents two similar methods but different slitty approaches. Genetic algorithm representation is essential, and it presents two different ways: vector and tree-like graphs. The vector representation is relatively simple and easier to work with. The downside of this study is computational cost. It requires quite a lot of computation cost to run a genetic algorithm. However, it can be controlled flexibly by adjusting population size and number of generations. As a result, the study developed well perform text classification

model. The result of the study provides evidence that the genetic algorithm can be used for machine learning model development, and it is pretty compelling.

A text on social media platforms or any other online platform can be pretty short. For instance, content on Twitter is relatively short. The platform has a word limit of 280 characters [15]. Classifying short text is not an easy task. The study by Wang Haitao [15] presents a method to tackle this difficult task. First of all, in-text classification task, the short text is harder to deal with for some reason. This is simply because of its lack of necessary information for text classification. The study uses Convolutional Neural Network because of its promised performance in a text classification task. It goes through four steps to do short text classification. First, it uses Jaro–Winkler similarity to detect any spelling error on text. Spell miss can lead to entirely different output in the short text. Secondly, it finds related words to extend the semantics of short text. Finally, conceptualize short text and then extract short text features. As a result, the study successfully improved the performance of short text classification. Combining this technique with a traditional text classification method in a real-world application can be pretty challenging. However, at the same time, it is a necessary technique to apply text classification to various content on online platforms.

Those studies consider various methods to construct well perform text classification model. Therefore, the results of those studies are a beneficial source to consider the methodology to this project of offensive language rating.

3 Design

3.1 Overview

The main objective of this project is to develop an offensive language detection model using Machine Learning and Neural Network knowledge. The project also involves some knowledge from Natural Language Processing and Artificial Intelligence. The model aims to detect various offensive languages on online platforms, including social media platforms and other platforms that allow users to leave a comment. The project only considers one language, which is English. Reasons for this will be explained in a later section. This area of research is one of the main focuses in the Natural Language Processing research community, mainly because of increasing online platforms. Offensive language detection is the necessary technology to keep online platforms healthy in many ways [7]. A conventional text detecting model classifies a text into a specific class. For instance, particularly to this project's topic, classes can be 'hate,' 'violence,' 'offensive.' 'ok,' etc. The language is often very ambiguous, and it is always a challenge that natural language processing face. Of course, this ambiguity applies to offensive language, as well. Categorization is a crucial part of modeling the Machine Learning model. Categorization is complex in offensive language because of its variety [7]. There are many possible ways to categorize them. The general approach would be to categorize them into simple classes such as 'hate,' 'offensive,' 'ok,' etc. However, different researchers use a different level of abstraction and can be pretty unclear [7]. There is research [9] that shows results that demonstrate how abstracted class categorization is complex for the Machine Learning model, particularly for this

area of research. To overcome this challenge, a study by a group of researchers from the Alan Turing Institute [7] proposed a different way of categorizing offensive language. Also, the study by Rishav Hada [1] proposes rating each text with a degree of offensiveness which is a pretty different approach compared to the conventional Machine Learning model. This project uses a second approach to construct a Machine Learning model. When constructing a Machine Learning model, tuning parameters can be quite challenging. Also, it will require much time and human resources, in this case, myself. This project will use a genetic algorithm to automatically tune parameters for building Machine Learning models and construct Neural Networks. A genetic algorithm can take a long time to run, but it can run without the presence of a human. Also, this will be interesting to see what level of creativity can genetic algorithm provide to construct a deep Machine Learning model. There is one more thing that the project needs to consider, which is short text classification. Since short text lacks the necessary contextual information to train the Machine Learning model [15]. Online platforms and social network platforms are full of short text. This part might become necessary to this project, but it is not the main focus.

3.2 Templeate of this project

The main template of this project is Deep Learning on a public dataset from CM3015 Machine Learning and Neural Networks. Also, this project overlaps with another template: Automated design using evolutionary computation from CM3020 Artificial Intelligence. In addition to these templates, some knowledge from CM3070 Natural Language Processing will contribute to this project.

3.3 Domain and users

The primary users for this project are young adult internet users and online platform providers such as social network platforms. Most young adults are users of various online platforms, and not all platforms have proper regulations and technologies against offensive languages. The various negative effects of offensive language from the online platform are concerning [6] [13]. Also, it became essential for online service providers to have a robust system to detect offensive languages to provide safe platforms [7]. This project can also contribute to various psychological research related to negative psychological effects on mental health, such as the study about young students' mental health [13].

Inappropriate language detecting application is the primary domain for this project. Protecting people's mental health from offensive language is also a part of the domain. This technology area is still considered a challenging area of research, and not all online platform providers can adopt those into their service. Also, big technology companies have already placed systems to reduce the amount of offensive language within service, but there is still room for improvement. Nevertheless, this area of research can contribute to protecting people's mental health against unhealthy online behaviors.

Today, the Internet is available in most places in the World and even in space. Therefore, hundreds of different languages are used on various online platforms. This project focuses on English because it is one of the most common languages in the World. However, it is possible to apply this project to other languages in further

works to reach a broader range of users. Offensives language detecting systems related works perform well. However, it sometimes detects regular comments as offensive or offensive language gets around the system. Also, there is a challenge with an unclear categorization of offensive language and overall ambiguity of offensive language. Overcoming those challenges can provide a better detecting system. Furthermore, it is automating tuning process can implementation process more accessible for those online platform providers and make this project adaptable to different topics of text classification.

3.4 Overall Structure

This project will follow the basic steps of deep learning model development. The first steps will be data installation and pre-processing. This process involves installing all necessary datasets into a machine's local environment. Then pre-process these datasets as necessary, including formatting, data cleaning, and split into train, test, validation datasets. In the second step, the project will carry out some data analysis on the dataset to better understand the dataset, which includes some fundamental Natural Language analysis. The actual deep learning model construction will take place next. The base model will be from the research [1], which shows the most effective and relates to this project the most. However, the project will consider different choices from other related works. As part of model construction, a genetic algorithm will search for the best-performing parameter combination.

In this project, there are various technologies involved. The most important technologies are the deep learning model, which consists of various knowledge, including Machine Learning and Neural networks. Also, knowledge of Natural Language is vital for this project to succeed. In addition, genetic algorithms play an essential role in automating some parts of the project and provide some creativity.

Another important aspect of this project is evaluation. The result of model performance will be analyzed with Pearson correlation, MSE, and error analysis as the study by Rishav Hada [1] analyzes their data. To evaluate this project, the result will be compared with other related works. Since rating offensive language with numerical values is somewhat subjective, testing will require a third party to see the result. However, this type of test can be carried out with some users.

3.5 Project Plan

Project planning is a crucial criterion for a successful project. Detail plan for this project is gathered on one Gantt chart below ((a) Gantt Chart on Figures section). The plan is set weekly, and the week count is the same as the UoL module scheduling system. It starts from Week 9, the following week of peer graded assignment submission for project design. There are some extra weeks of work plan in important parts of this project. Last 5 weeks of implementation are dedicate towards evaluation, testing and implementation of a model. Testing and evaluation are critical part of the project, and project can be improved based on result of those. For a write up task, additional draft is included to make sure its quality is high at the final submission. Also, for that purpose, the last 2 weeks of the project is dedicate for write up only. Also, those extra weeks are for cases where thighs did not go as planned. Putting some insurance in planning ensures that the project will be completed on time.

4 Implemementation

4.1 Overviwe

The main objective of this project is to develop an offensive language detection deep learning model with the genetic algorithm. The model aims to classify offensive languages between the scale of -1 (not offensive) and 1 (very offensive). In addition, the genetic algorithm will be used to improve the model automatically. By the date of this report, implementations are focused on developing the base model.

4.2 Dataset

Dataset for this project was obtained from Ruddit research paper [1]. The dataset is available on the GitHub page of the paper. However, the dataset only contains the id of each comment. Therefore, each comment has to retrieve using Reddit API. This process only requires a couple of lines of Python script ((d) on the figures section).

This Python code iterates through each row of the original dataset and converts comment id into actual comment using Reddit API. Some comments no longer exist on Reddit, which are removed from the dataset.

4.3 Model

The main components of the initial prototype of the model are convolutional layers and a pre-trained GloVe word vector. From the performance of the prototype, convolutional layers show some promise performance. Models from the Ruddit research paper [1] were considered in the first implementation. Although the Ruddit paper uses the PyTorch library, this project uses the TensorFlow library. Bert and HateBert models perform well on the Ruddit research; however, there is a downside. The size of both models are enormous, and there is no access to the hardware that can handle those model with a decent time. For that reason, a different mode was considered, the BiLSTM model. This model is one of the models that the Reddit paper used. There are two different implementations at this date. Both with the same model but different tokenizers. One with BERT tokenizer [11] and another with HateBERT [12] tokenizer. These two implementations have somewhat experimental elements. Models are implemented based on how it is implemented on Reddit but slightly different. The model is a combination of Convolutional layers and BiLSTM layers. This combination had a better result than a model with just BiLSTM layers. Please refer to the (b) on the figures section.

5 Evalutation

Evaluation is carried out with Mean Square Error, which is also used for the loss function on model training. Also, accuracy is calculated in part of the evaluation. Since the model is trained to classify each comment with a value, it classifies the result as accurate if the absolute difference between actual and predicted values is within a specific range. Finally, detailed results are shown in the table. Please refer to the (c) Result of model prediction on the figure section velow.

Overall, MSE is not low as the model from Reddit. Accuracy is pretty high with 0.3 differences. However, there is plenty of room to improve the model, and the genetic algorithm should

be a helpful tool. By implementations so far, the base model is constructed but still needs improvement. Also, more efficient evaluations, including unit testing, are necessary for further development.

6 Conclusion

The usage of offensive and inappropriate language is a significant problem that everyone needs to consider on the modern internet, especially where people can use their own words. The development of this project up to this date suggests that the deep learning model can be developed and used to evaluate the offensiveness of English sentences. However, it is a challenging task, and different considerations need to be taken.

There is still much room for improvement and development required in this project. Also, different approaches need to take place, including developing own tokenizers and developing the model using the genetic algorithms. Then, more experimental elements can take this project with more successful results.

References

[1] Rishav Hada, Sohi Sudhir, Pushkar Mishra, Helen Yannakoudakis, Saif M Mohammad, and Ekaterina Shutova. 2021. Ruddit: norms of offensiveness for english reddit comments. *arXiv preprint arXiv:2106.05664*.

[2] Borowiec Steven. 2021. Hana kimura death: man charged over cyberbullying of japanese reality tv star. (2021).

[3] Kyodo News. 2021. Japan eyes tougher jail sentence for insults to tackle cyberbullying. `https://english.kyodonews.net/news/2021/09/100ee4e7a81d-japan-eyes-tougher-jail-sentence-for-insults-to-tackle-cyberbullying.html`. (2021).

[4] World Health Organization. 2020. Who director-general's opening remarks at the media briefing on covid-19 - 11 march 2020. `https://www.who.int/director-general/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020`. (2020).

[5] Hongfei Liu, Wentong Liu, Vignesh Yoganathan, and Victoria-Sophie Osburg. 2021. Covid-19 information overload and generation z's social media discontinuance intention during the pandemic lockdown. *Technological Forecasting and Social Change*, 166, 120600.

[6] Pouria Babvey, Fernanda Capela, Claudia Cappa, Carlo Lipizzi, Nicole Petrowski, and Jose Ramirez-Marquez. 2021. Using social media data for assessing children's exposure to violence during the covid-19 pandemic. *Child Abuse & Neglect*, 116, 104747.

[7] Bertie Vidgen, Alex Harris, Dong Nguyen, Rebekah Tromble, Scott Hale, and Helen Margetts. 2019. Challenges and frontiers in abusive content detection. In *Proceedings of the third workshop on abusive language online*, 80–93.

[8] Hayden Andersen, Sean Stevenson, Tuan Ha, Xiaoying Gao, and Bing Xue. 2021. Evolving neural networks for text classification using genetic algorithm-based approaches. In *2021 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 1241–1248.

[9] Shervin Malmasi and Marcos Zampieri. 2017. Detecting hate speech in social media. *arXiv preprint arXiv:1712.06427*.

[10] Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[12] Tommaso Caselli, Valerio Basile, Jelena Mitrović, and Michael Granitzer. 2020. Hatebert: retraining bert for abusive language detection in english. *arXiv preprint arXiv:2010.12472*.

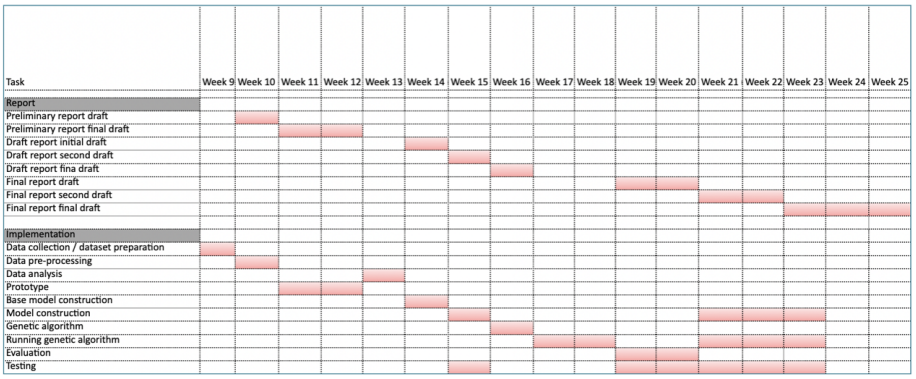
[13] Xiaoqin Shi, Chao Yu, and Dongmei Wu. 2021. Influence of internet language violence on young students' mental health and intervention countermeasures. *Journal of healthcare engineering*, 2021.

[14] Joel Lehman, Jeff Clune, Dusan Misevic, Christoph Adami, Lee Altenberg, Julie Beaulieu, Peter J Bentley, Samuel Bernard, Guillaume Beslon, David M Bryson, et al. 2020. The surprising creativity of digital evolution: a collection of anecdotes from the evolutionary computation and artificial life research communities. *Artificial life*, 26, 2, 274–306.

[15] Haitao Wang, Keke Tian, Zhengjiang Wu, and Lei Wang. 2021. A short text classification method based on convolutional neural network and semantic extension. *Int. J. Comput. Intell. Syst.*, 14, 1, 367–375.

7 Figures

(a) Gantt Chart



(b) BiLSTM model with Conv1D

```
Model: "sequential_9"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_9 (Embedding)     (None, 300, 256)          7813888

 conv1d_2 (Conv1D)           (None, 298, 20)           15380

 max_pooling1d (MaxPooling1D  (None, 149, 20)          0
 )

 bidirectional_8 (Bidirectio  (None, 149, 20)          2480
 nal)

 bidirectional_9 (Bidirectio  (None, 20)               2480
 nal)

 dropout_4 (Dropout)         (None, 20)                0

 dense_4 (Dense)             (None, 1)                 21

=================================================================
Total params: 7,834,249
Trainable params: 7,834,249
Non-trainable params: 0
_________________________________________________________________
```

(c) Result of model prediction

	MSE	Accuracy (> 0.3)	Accuracy (> 0.2)	Accuracy (> 0.1)
BiLSTM with Bert	0.046	0.905	0.763	0.47
BiLSTM with HateBert	0.048	0.895	0.741	0.464
BiLSTM from Ruddit [1]	0.035	n/a	n/a	n/a

(d) Code to convert comment ID to comment

```
df = pd.read_csv('sample_input_data.csv')
for index, row in df.iterrows():
    comment = get_comment(row['comment'])
    final_df = final_df.append({"id": row["k_id"], "comment":comment, "score":row["Score"]},
                    verify_integrity = True,
                    ignore_index =True,
                    )
    print(str(index) + comment)
```