_
_

成都理工大學中英合作办学
CDUT Sino-British Collaborative Education

OXFORD BROOKES UNIVERSITY

UNDERGRADUATE PROJECT PROGESS REPORT

Project Title:	**Toxic Comments Classification Based on Deep Learning**
Surname:	**Gao**
First Name:	**Ruiling**
Student Number:	**201918020301**
Supervisor Name:	**Joojo Walker**
Module Code:	**CHC 6096**
Module Name:	**Project**
Date Submitted:	**09/01/2023**

Table of Contents

1 Introduction

1.1 Background

With the advent of the digital age, the number of electronic documents is increasing. One of the more typical phenomena is that with the growth of online communities on the Internet, a large amount of cluttered textual information, such as posts, comments, etc., floods the entire online environment. Most normal and virtuous netizens are spontaneously maintaining the sustainability and usability of the Internet, but in contrast, there are some users who post anti-social and malicious comments on online platforms in an attempt to undermine the usability of the Internet [1]. The emergence of malicious comments violates the legitimate rights of netizens and can cause serious mental and psychological harm to them. Therefore, the efficient management and processing of large volumes of comment texts have become a target of interest for researchers [2].

Text classification is one of the effective ways to locate and triage information efficiently and accurately, solving the problem of information clutter as much as possible [3]. Text classification (TC), also known as text categorization, is an extensive area of current research in linguistic text mining and processing. TC is a process that uses deep learning algorithms to categorize text content into pre-given sets of labels [4]. Deep learning-based text classification techniques have been developed and matured since the 1990s. Compared to text classification systems based on knowledge engineering and expert systems, classification techniques using deep learning provide better classification results and flexibility and have become the main techniques used in related fields [5]. Among the techniques for text classification by different criteria, sentiment analysis (SA), also known as opinion mining, is a branch of text classification. Its main function is to identify and analyze the sentiment in a text by using pre-given labels with human sentiment colors and sentiment tendencies, such as positive, negative, neutral, etc [6].

The report is divided into five sections and the structure of the report, and the main content of each section is organized as follows. The first section introduces the basic concepts of text classification, the purpose and significance of this research, an analysis of the problems addressed by text classification, and an overview of the research on the topic. Finally, the overall structure of the report is given. Section 2 introduces the

research background of text classification and summarizes the current state of research and the main features of text classification. Section 3 presents the main research methods chosen for the topic, the techniques used, and the data and model testing strategy. Section 4 shows the project management plans including the activities, schedules, data and version management plans. The final section presents professional issues and risks that are relevant to the project.

1.2 Aim

The main goal of this project is to develop different deep learning-based models for the detection and classification of toxic comments automatically.

1.3 Objectives

The objectives of this text classification project are as follows:

- Conduct background research on text classification, understand the field of Natural Language Processing and the corresponding technologies.
- Collect usable dataset from the Internet.
- Clean and pre-process the data for modeling.
- Extract features from the text in the cleaned datasets.
- Train different models using datasets and assessing the quality of the models
- Analyze the quality of the models and compare the strengths and weaknesses of each model
- Develop data and model testing and evaluation strategy
- Risk analysis based on current progress

1.4 Project Overview

1.4.1 Scope

The project is designed to analyze the sentiment of comments made by users on f a business review website in the United State named Yelp, filtering out malicious comments and classifying them into different categories, such as hate speech, personal attacks, pornography, or violence, etc. The project helps social network staff to

automatically screen out therefore manage malicious comments, reducing labor and time costs, while also helping to clean up the online environment.

1.4.2 Audience

Text classification is one of the effective managements to helps to locate and triage information efficiently and accurately, solving the problem of information clutter as much as possible.[3]

2 Background Review

	Recall Ratio	Precision Ratio	F1	Data processing
Bi-LSTM + Word2Vec	97.00%	89.00%	92.00%	Word2Vec
Bi-LSTM	/	/	92.79%	Attention selection mechanism + Fine-grained text classification
AC-BiLSTM	87.81%	87.28%	86.45%	Attention mechanism + Convolutional Layer

Table 1: existing approaches and their feature

The table illustrates the features of existing approaches.

To date, several models have been used in research on sentiment classification. Among them, Kong Fand Chen G have presented a neural network model combining Word2Vec [7] with Bi-LSTM[8] to learn the spatial representation of word vectors through Word2Vec, transforming the text into a sentence representation in the input layer feature space, and improve the network using constant mapping covariance theory. The model using the improved Bi-LSTM is able to present excellent improvements in the dataset.

Ding Y have proposed a classification model based on an attentional mechanism called ON-LSTM [9]. The method is mainly based on a transfer learning approach through feature extraction, where the performance of the model is tuned to the best in the source dataset and then applied to the test set. A multi-level embedding model under the attentional selection mechanism is also proposed. Through the embedding representation at the character level and sentence level in addition to the word level, information that is more conducive to classification in the text can be extracted[10].

Li G and Guo J have proposed a new architecture of bidirectional LSTM (Bi-LSTM) [8] with attention mechanism and convolutional layer, which can more precisely extract text semantics and achieve better text classification results. In AC-BiLSTM, the convolutional layer is used to retrieve higher-level phrase representations from word embedding vectors, and Bi-LSTM is used to gain access to the forward and backward contexts. The attention mechanism is applied to put more focus on the important information in the output of the hidden layer. A SoftMax classifier is ultimately used to categorize the text information after processing. The strength of the AC-BiLSTM model lies in its ability to both extract partial features of phrases and to understand the semantics of phrases within a sentence [11].

3 Project Technical Progress

3.1 Methodology

3.1.1 Approach

This report is designed to investigate the effectiveness of different deep learning models for predicting malicious comments by modelling the same dataset and comparing the accuracy of different models. By modelling the same dataset and comparing the accuracy of different models, the most accurate model is selected as the most suitable machine learning model for malicious comment classification.

In this project, recurrent neural networks (RNN) are used as the main model. The use of RNN networks dated to the 1980s and has evolved into one of the main network models in the field of deep learning[12]. One thing that distinguishes RNN from CNN is that RNN can be applied to scenarios where the input can be a sequence of sequence type of data. In other words, the output of an RNN network is somewhat related to its previous output. Due to its recurrent feed-back connections, RNN model is able to establish relationships between the inputs of the preceding and following sequences, allowing the output values of the RNN at each moment to be influenced by the input values of multiple previous moments, yielding more accurate results[13]. Different RNN models were used in this project to compare their performance. The structure of the RNN is illustrated in Figure 1.

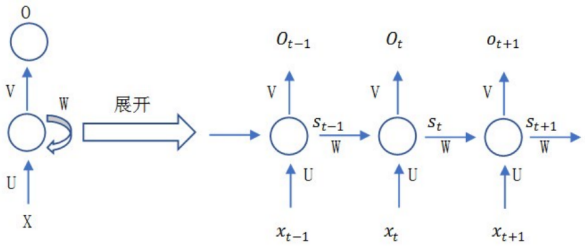

Figure 1[13] Structure of RNN

Long Short-Term Memory (LSTM) [14] is one of the RNN variants created to solve the RNN gradient vanishing problem. LSTM has a linear unit called constant error carousels (CECs) and is controlled by three gates that are used to store the input into the model from real time information entered the model[15]. The input gate controls whether information from the current moment is allowed to be added to the CEC, the output gate controls whether information from the previous moment's CEC will be output to affect the output of the next moment's node, and the forget gate controls whether information from the current moment's CEC will be formatted. Figure2 illustrates an architecture of a single LSTM unit, where c_j represents the entire memory cell. Where c_j represents the whole memory cell, in_j and out_j represent different gates respectively.

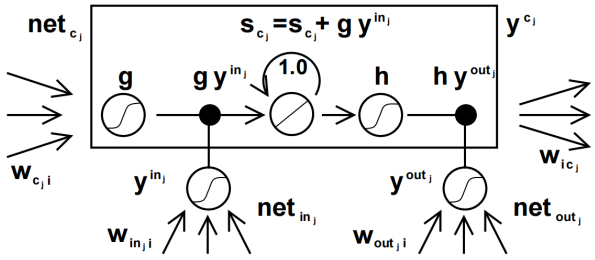

Figure 2[14] Architecture of LSTM Unit

Gated Recurrent Unit neural network (GRU) [16] improves on the LSTM by not only retaining the gate feature of the LSTM, but also simplifying it by reducing the three gates in the model to two gates: the update gate and the reset gate[13]. The update gate is used to control how much of the information in the CEC from the previous moment will be output and thus affect the output of the node at the next moment, and the reset gate is used to control how much information can be added to the CEC. Figure 3 shows a schematic diagram of the GRU structure.

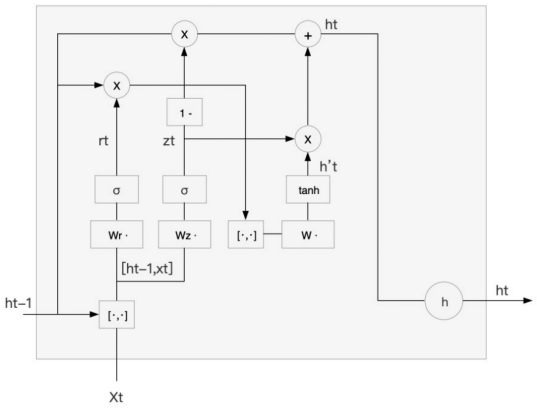

Figure 3[13] Structure of GRU

This paper focuses on training and tuning the parameters of these different models, and finally evaluating their performance in terms of accuracy, prediction rate, recall and other evaluation methods. The data used for this study is a 68.5M English comment dataset crawled from social media platforms 'Yelp' on the Internet, which contains the ids of the posters, the content of the comments posted, and the different types of toxicity. By using the same dataset with the same data pre-processing progress, controlling variables to ensure the type of model as single variable, so that the performance of different models can be analyzed as accurately as possible.

3.1.2 Technology

The experimental environment used in this paper: M1 ProM1 Pro integrates several different components, including CPU, GPU, unified memory architecture (RAM), neural engine, etc.

The 8-core M1 Pro is equipped with a 14-core GPU and a 16-core neural engine for machine learning.

Python 3.8, TensorFlow 2.7.0 with Jupyter Notebook are used to implement the methods

3.2 Testing and Evaluation

Two generic model-like tests, pre-train and post-train tests, are written[16].

3.2.1 Pre-train Test

Some tests can be used to test the data without adjusting the parameters.

- Check if the type of labels of the training and testing sets are the same one-hot code
- Check if the data of the training and testing sets are both containing id and comment_text
- Check if the output of the LSTM and GRU model matches all types in the label
- Check that the range of the LSTM and GRU model output matches the range of the label of 0 to 1
- Adding assertions to the model to control the operation of the model

3.2.2 Post-train Test

- Invariance Tests：Manually make changes (e.g.: change “a” to “A”) to the data entered the LSTM and GRU model to see if there is an impact on the model's predictions while ensuring that the output is not affected
- Directional Expectation Test: Manually make changes (e.g.: change “hate” to “like”) to the data entered the LSTM and GRU model to see if there is an impact on the model's predictions if it interferes with the model output
- Data Unit Test: Classify possible erroneous results in the model

Once these processes had been done, the evaluation and testing of the model can be used as a basis for modifying and refining the model.

3.2.3 Model Evaluation

Firstly, four criteria are used to evaluate the performance for each class in this experiment, namely the accuracy (P), the recall (R), the harmonic mean of recall and accuracy (F1-score). After calculating these criteria for each class, macro-F1, Accuracy and Area under Curve (AUC) are used to evaluate the performance of the whole model.

In particular, the accuracy represents the number of samples with positive predictions that are predicted correctly, and the recall refers to how many of the positive samples in the dataset are predicted correctly. The confusion matrix representing the actual and predicted values is shown in the table below.

Actual / Predict	Positive	Negative
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

Table 2 Confusion Matrix

In the table, TP: samples that are positive and predicted to be positive; FN: samples that are positive but predicted to be negative; FP: samples that are negative but predicted to be positive; TN: samples that are negative and predicted to be negative.

Therefore, the expressions of the evaluation criterion of the model are as (1) to (5):

$$Precision(P) = \frac{TP}{TP + FP} \tag{1}$$

$$Recall\ (R) = \frac{TP}{TP + FN} \tag{2}$$

$$F1 - score = \frac{2\ \times\ P \times R}{P\ + R} \tag{3}$$

$$macro - F1 = \frac{2\ \times \frac{1}{n}\sum_{i=1}^{n} P_i\ \times \frac{1}{n}\sum_{i=1}^{n} R_i}{\frac{1}{n}\sum_{i=1}^{n} P_i\ \times \frac{1}{n}\sum_{i=1}^{n} R_i} \tag{4}$$

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{5}$$

3.3 Design and Implementation

The data for this experiment was taken from the Kaggle "toxic comment classification challenge" dataset and was divided into a training set with labels and a test set with labels. Firstly, a pre-processing phase was applied to the dataset to ensure a better feature extraction and model building phase. In Jupyter Notebook, a data walkthrough of the training set and the test set is implemented. From this stage it can be roughly seen that the dataset is divided into user_ids, the content of the user's comments, and the classification of one-hot labels of the content. After getting a rough idea of the data, some pre-processing operations were implemented. Firstly, all letters were converted to lower case for better feature extraction. Then data visualization was used to get an overview of the content of the dataset used for this experiment. The most used word

segmentation package is the python Natural Language Toolkit (NTLK) developed at the University of Pennsylvania. NLTK library was called to remove distracting characters such as punctuations, common stop words, etc. from the comments. After removing these stop words, the textual semantics stay the same. In addition, the number of words per comment and the word cloud that has been classified as malicious and non-malicious comments in the entire training set are displayed separately. Below shows the word frequency and word cloud for different datasets.

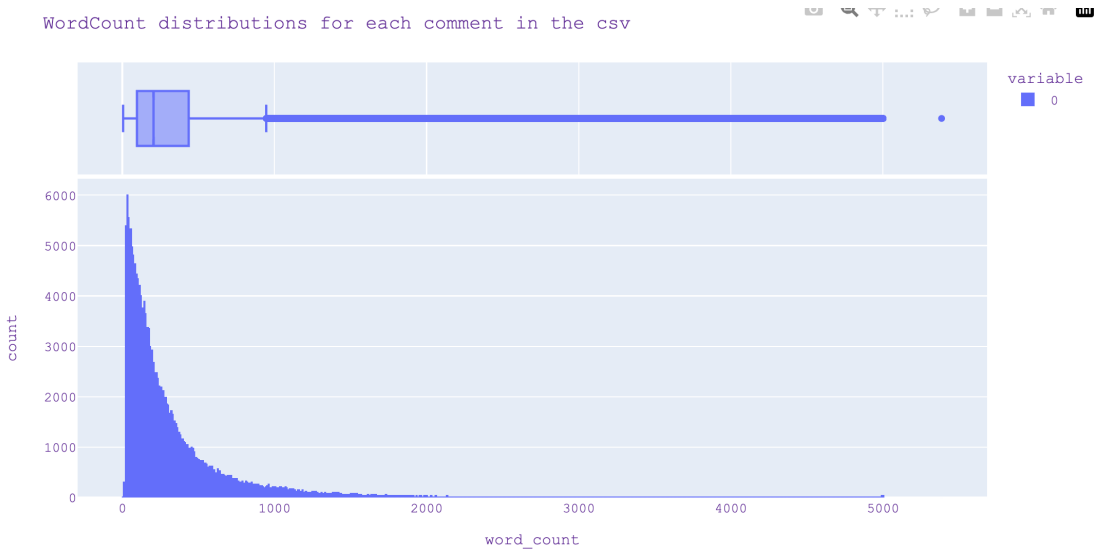

Figure 4 Word Frequency

Figure 5 WordCloud for Toxic Comments

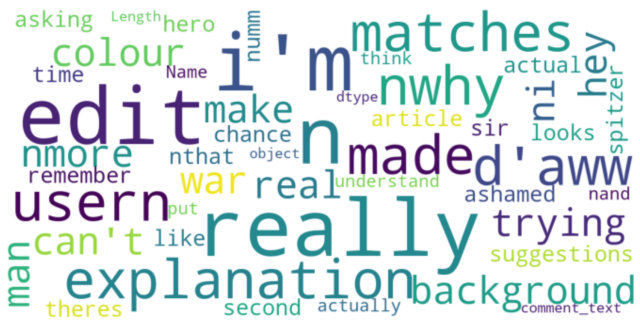

Figure 6 WordCloud for non-Toxic Comments

After the overview of the data and the initial processing, most of the comments in the entire dataset are concentrated in less than 500 characters, so data pre-processing operations are applied to retain the important information in the comments for better word separation operations: The comment text was converted into processed text input by limiting its length and size. Below shows the word length after the pre-processing.

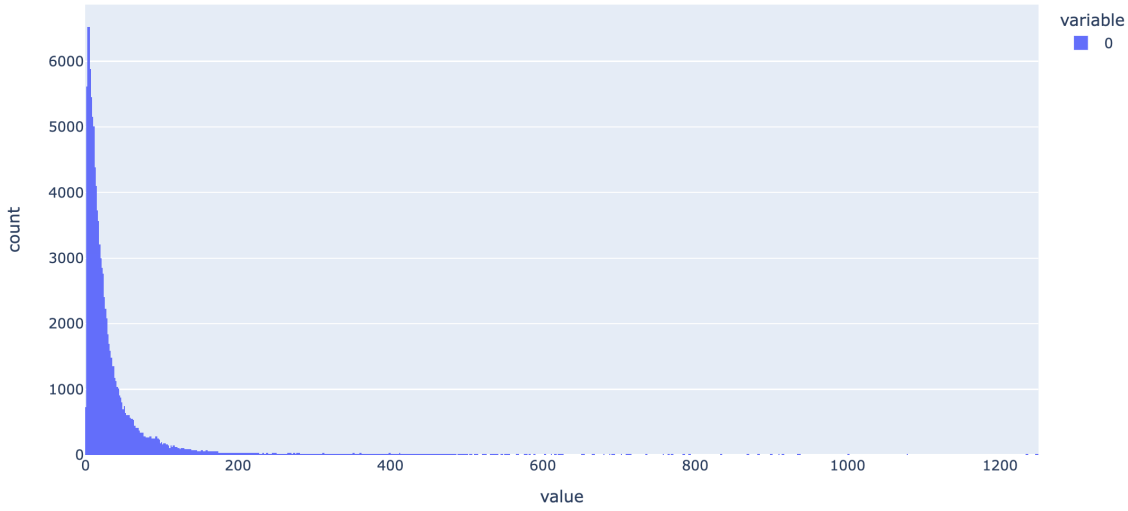

Figure 7 Word Length

After the pre-processing operation, the base models of LSTM and GRU are built first. The model of LSTM is built in four layers: an embedding layer for data dimensionality reduction, a recurrent LSTM layer, a Dropout layer to prevent overfitting and finally a dense layer. The hyperparameters of the model are selected as 12 batch sizes, 2 epochs, 50 LSTM layers units and the dropout rate of 0.5. The model is shown in the figure below.

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_2 (Embedding)     (None, 200, 13)           390000

 lstm_2 (LSTM)               (None, 50)                12800

 dropout_2 (Dropout)         (None, 50)                0

 dense_2 (Dense)             (None, 6)                 306

=================================================================
Total params: 403,106
Trainable params: 403,106
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 8 LSTM Model

The test set was applied to the model after training data was trained by the LSTM model. The training results and evaluation with the test results are shown below.

```
Epoch 1/2
2023-01-11 17:48:45.395050: I tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:112] Plugin opti
mizer for device_type GPU is enabled.
2023-01-11 17:48:45.927666: I tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:112] Plugin opti
mizer for device_type GPU is enabled.
2023-01-11 17:48:46.645920: I tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:112] Plugin opti
mizer for device_type GPU is enabled.
3990/3990 [==============================] - ETA: 0s - loss: 0.1381 - accuracy: 0.9504
2023-01-11 17:53:20.085655: I tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:112] Plugin opti
mizer for device_type GPU is enabled.
2023-01-11 17:53:20.195875: I tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:112] Plugin opti
mizer for device_type GPU is enabled.

Epoch 00001: val_loss improved from inf to 0.12407, saving model to model_toxic/cp.ckpt
3990/3990 [==============================] - 298s 74ms/step - loss: 0.1381 - accuracy: 0.9504 - val_loss: 0.1241 - va
l_accuracy: 0.9933
Epoch 2/2
3990/3990 [==============================] - ETA: 0s - loss: 0.1139 - accuracy: 0.9889
Epoch 00002: val_loss improved from 0.12407 to 0.10220, saving model to model_toxic/cp.ckpt
3990/3990 [==============================] - 317s 79ms/step - loss: 0.1139 - accuracy: 0.9889 - val_loss: 0.1022 - va
l_accuracy: 0.9916
```

Figure 9 Train Result

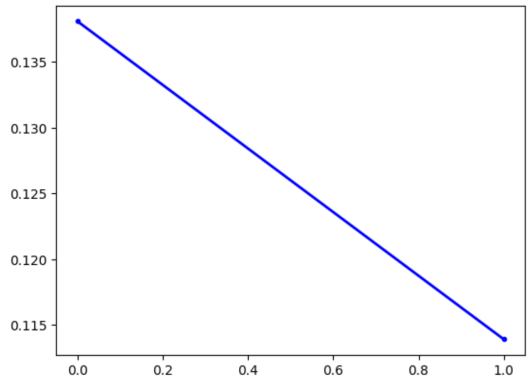

Figure 10 LSTM Train Loss

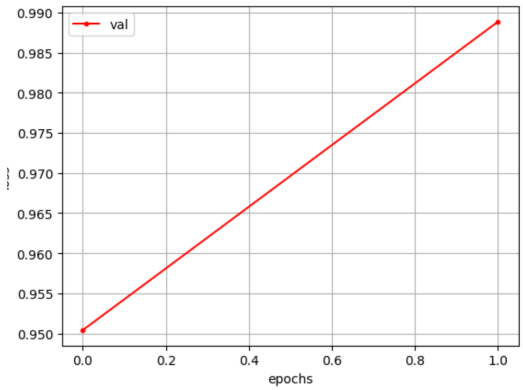

Figure 11 LSTM Train Accuracy

```
LSTM_result
```

executed in 50ms, finished 19:01:01 2023-01-11

	id	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	00001cee341fdb12	0.010622	0.000032	0.002704	4.686829e-06	0.002007	0.000069
1	0000247867823ef7	0.056379	0.000982	0.020023	2.931993e-04	0.019213	0.001890
2	00013b17ad220c46	0.049274	0.003632	0.033501	7.919517e-04	0.025712	0.003386
3	00017563c3f7919a	0.003306	0.000003	0.000594	3.038785e-07	0.000456	0.000008
4	00017695ad8997eb	0.040610	0.000371	0.014727	5.493087e-05	0.011600	0.000493
...	...	...	...	...	...	...	...
153159	fffcd0960ee309b5	0.191648	0.009699	0.104382	2.439720e-03	0.094987	0.011140
153160	fffd7a9a6eb32c16	0.035808	0.000496	0.011523	1.699445e-04	0.010624	0.001185
153161	fffda9e8d6fafa9e	0.020560	0.000182	0.005705	6.214662e-05	0.005346	0.000517
153162	fffe8f1340a79fc2	0.035868	0.000528	0.011663	1.924248e-04	0.010924	0.001328
153163	ffffce3fb183ee80	0.264404	0.016192	0.147551	4.138482e-03	0.145079	0.017195

153164 rows × 7 columns

Figure 12 LSTM Test Result

A GRU model was then compared with the same hyperparameters. The model itself, the model training results and evaluation with testing results are shown below.

```
Model: "sequential_8"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_8 (Embedding)     (None, 200, 13)           390000

 gru_5 (GRU)                 (None, 50)                9750

 dropout_8 (Dropout)         (None, 50)                0

 dense_8 (Dense)             (None, 6)                 306

=================================================================
Total params: 400,056
Trainable params: 400,056
Non-trainable params: 0
_________________________________________________________________
None
```

Figure 13 GRU Model

```
Epoch 1/2
2023-01-11 18:36:00.640222: I tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:112] Plugin opti
mizer for device_type GPU is enabled.
2023-01-11 18:36:00.995253: I tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:112] Plugin opti
mizer for device_type GPU is enabled.
2023-01-11 18:36:01.667049: I tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:112] Plugin opti
mizer for device_type GPU is enabled.
3990/3990 [==============================] - ETA: 0s - loss: 0.1031 - accuracy: 0.9397
2023-01-11 18:40:38.761192: I tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:112] Plugin opti
mizer for device_type GPU is enabled.
2023-01-11 18:40:38.929872: I tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:112] Plugin opti
mizer for device_type GPU is enabled.

Epoch 00001: val_loss improved from inf to 0.07801, saving model to model_toxic/cp.ckpt
3990/3990 [==============================] - 306s 76ms/step - loss: 0.1031 - accuracy: 0.9397 - val_loss: 0.0780 - va
l_accuracy: 0.9681
Epoch 2/2
3990/3990 [==============================] - ETA: 0s - loss: 0.0809 - accuracy: 0.9492
Epoch 00002: val_loss did not improve from 0.07801
3990/3990 [==============================] - 299s 75ms/step - loss: 0.0809 - accuracy: 0.9492 - val_loss: 0.0808 - va
l_accuracy: 0.9773
```

Figure 14 GRU Train Result

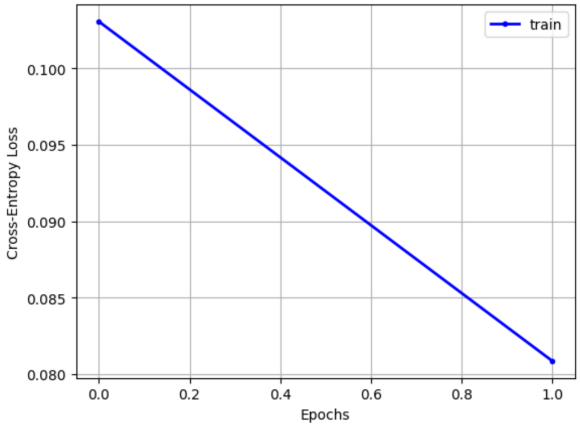

Figure 15 GRU Train Loss

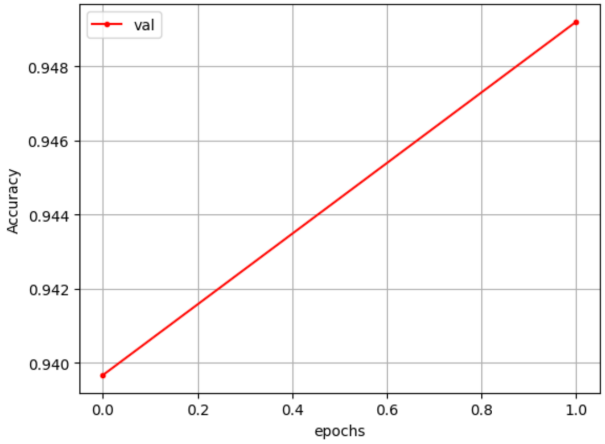

Figure 16 GRU Train Accuracy

```
GRU_result
```

executed in 86ms, finished 19:00:04 2023-01-11

	id	toxic	severe_toxic	obscene	threat	insult	identity_hate
0	00001cee341fdb12	0.918934	0.123065	0.927919	0.008091	0.768234	0.045929
1	0000247867823ef7	0.013001	0.000264	0.000773	0.000442	0.002512	0.001449
2	00013b17ad220c46	0.155038	0.016310	0.080393	0.004487	0.086675	0.019778
3	00017563c3f7919a	0.004974	0.000049	0.000218	0.000080	0.000720	0.000367
4	00017695ad8997eb	0.036303	0.002944	0.006995	0.001408	0.010426	0.004962
...	...	...	...	...	...	...	...
153159	fffcd0960ee309b5	0.014423	0.000415	0.001356	0.000408	0.002763	0.001376
153160	fffd7a9a6eb32c16	0.062318	0.000614	0.006438	0.000620	0.012461	0.003022
153161	fffda9e8d6fafa9e	0.028932	0.000171	0.002493	0.000203	0.005181	0.001204
153162	fffe8f1340a79fc2	0.025963	0.000097	0.001985	0.000107	0.004452	0.000716
153163	ffffce3fb183ee80	0.148063	0.001285	0.037176	0.000790	0.042000	0.003786

153164 rows × 7 columns

Figure 17 GRU Test Result

4 Project Management

4.1 Activities

All the activities related to the objectives are shown in the table below.

Objects	Activity	Completed
Ob1: Conduct background research on text classification, understand the field and the corresponding technologies.	A1.1 Identify subject keywords	Completed
	A1.2 Search for relevant essays	Completed
	A1.3 Read the relevant literature	Completed
	A1.4 Summary the advantages and limitations of different technologies	Completed
	A1.5 Perform a literature review	Completed
Ob2: Collect usable dataset from the Internet.	A2.1 Search for social media comment datasets on Kaggle	Completed
	A2.2 Download the datasets	Completed
	A2.3 Identify the structure of the datasets	Completed
Ob3: Clean and pre-process the data for modeling.	A3.1 Search for methods to clean the data	Completed
	A3.2 Apply methods on datasets	Completed
	A3.3 Evaluate the process	Completed
Ob4: Extract features from the text in the cleaned datasets.	A4.1 Search for methods to extract data	Completed
	A4.2 Implement methods on the datasets	Completed

Objects	Activity	Completed
	A4.3 Evaluate the feature extracting methods	Completed
Ob5: Train different models using datasets and assessing the quality of the models	A5.1 Search for documentation of different models	Completed
	A5.2 Apply models in Python language	Completed
	A5.3 Adjust the parameters until the model performs optimally	Uncompleted
Ob6: Analyze the quality of the models and compare the strengths and weaknesses of each model	A6.1 Search for different methods to evaluate the model	Uncompleted
	A6.2 Apply multiple rubrics to different models	Uncompleted
	A6.3 Put the results into a table	Uncompleted
Ob7: Develop data and model testing and evaluation strategy	A7.1Set up test plan for the dataset and model	Uncompleted
	A7.2 Put the test plan on the dataset and model	Uncompleted
	A7.3 Evaluate how well did the data and model perform	Uncompleted
Ob8: Risk analysis based on current progress	A8.1 Search for risk analysis for deep learning projects	Completed

Objects	Activity	Completed
	A8.2 Modify the risk analysis based on the used models	Completed
	A8.3 Implement the analysis on the models	Completed

Table 3 Activities of Objects

4.2 Schedule

The time schedule with the accomplished work and future work is shown below. The green color represents the accomplished work, and the orange color represents the undone work.

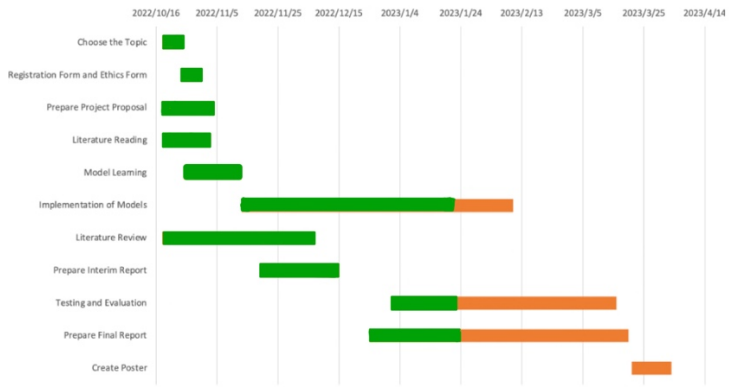

Figure 18 Schedule of the project period

4.3 Project Version Management

Whenever a new version of the code or related electronic documentation is updated, it will be uploaded to the Baidu Cloud in order to keep track of all project progress.

4.4 Project Data Management

Data of the project is planned to be stored in the GitHub:

Progress of the project can be seen in the sharing folder with the URL https://github.com/Ivvvvvvvy/OBU_Project

4.5 Project Deliverables

Throughout the execution of this project, the following items will be submitted for assessment:

- Project proposal with ethical forms, showing detailed description of the work to be done. (Submitted)
- Project weekly report containing planned objectives for each week. (Submitted)
- Progress report providing justification of the project. (Submitted)
- Project presentation illustrated by a poster and a practical demonstration. (Submitted)
- Final report which comprises a complete and clear explanation of the problem to be solved. (In process)

5 Professional Issues and Risk:

5.1 Risk Analysis

Possible risks that may appear through the process of accomplishing the project are listed below.

The table is arranged as:

Potential Risk: the possible risks that may appear in the process of the project

Potential Causes: The reason of having the risk

Severity: The impact degree potential causes may influence the project

Likelihood: The probability of the situation happening

Risk: The score of the Potential Causes

Mitigation: The method to prevent risk from happening

Risk ID	Potential Risk	Cause ID	Potential Causes	Severity	Likelihood	Risk	Mitigation ID	Mitigation
R1.1	Missed deadline	C1.1.1	Illness	1	6	6	M1.1.1	Inform the supervisor in time
		C1.1.2	Cannot choose topic	1	1	1	M1.1.2	Conduct research early and meet supervisor
		C1.1.3	Poor time management	3	4	12	M1.1.3	Follow the Gantt Chart strictly
R1.2	Unable to finish tasks	C.1.2.1	Overconfidence in ability	3	2	6	M1.2.1	Dicuss with supervisor in time about which is the unneccessary part of the tasks.
R1.3	Loss of data	C1.4.1	Poor version control	4	3	12	M1.4.1	Put every version of data on the cloud

Figure 19 Risk Analysis

5.2 Professional Issues

5.2.1 Legal Issue

This project strictly follows the regulations outlined in the General Data Protection Regulation (GDPR) [17] and is used to ensure that the data used for training and testing is legitimate, particularly with regard to transparency and purpose limitation.

5.2.2 Ethical Issue

The ethical issue raised by this project is that when the project is applied to real social media platforms, the accuracy of the model cannot be correct as many of the corpus cannot be updated in a timely manner due to the proliferation of new online words in today's online society. The misjudgment of non-malicious comments that results from this situation can lead to problems such as the banning of accounts of non-offending users, which may cause some legal concerns.

5.2.3 Social Issue

Most deep learning models are black box models. Because of the lack of intrinsic explanations of black box models, and the inevitably use of unbalanced sample data for model training, the use of such models can lead to a range of social problems such as gender bias and racial bias when the problem is severe.

5.2.4 Environment Issue

The main purpose of this model is to save human and time resources in social media platforms by automatically screening malicious comments by machines, which is environmentally friendly and greatly reduces the consumption of resources.

6 References

[1] N. Fan, Y. An, and H. Li, “Research on Analyzing Sentiment of Texts Based on K-nearest Neighbor Algorithm,” *Computer Engineering and Design*, vol. 33, no. 3, pp. 1160–1164, 2012, doi: 10.16208/j.issn1000-7024.2012.03.053.

[2] C. Shang, M. Li, S. Feng, Q. Jiang, and J. Fan, “Feature selection via maximizing global information gain for text classification,” *Knowl Based Syst*, vol. 54, pp. 298–309, 2013, doi: 10.1016/j.knosys.2013.09.019.

[3] Li R, “Research on Text Classification and Its Related Technologies,” Apr. 2005.

[4] A. Dhar, H. Mukherjee, N. S. Dash, and K. Roy, “Text categorization: past and present,” *Artif Intell Rev*, vol. 54, no. 4, pp. 3007–3054, Apr. 2021, doi: 10.1007/s10462-020-09919-1.

[5] J. S. Su, B. F. Zhang, and X. Xu, “Advances in machine learning based text categorization,” *Ruan Jian Xue Bao/Journal of Software*, vol. 17, no. 9, pp. 1848–1859, Sep. 2006, doi: 10.1360/jos171848.

[6] G. Wei and K. Wu, “Sentiment Analysis Based on Word Vector Model,” vol. 26, no. 3, 2016, doi: 10.15888/j.cnki.csa.005655.

[7] T. Tang, X. Tang, and T. Yuan, “Fine-Tuning BERT for Multi-Label Sentiment Analysis in Unbalanced Code-Switching Text,” *IEEE Access*, vol. 8, pp. 193248–193256, 2020, doi: 10.1109/ACCESS.2020.3030468.

[8] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997, doi: 10.1109/78.650093.

[9] Y. Shen, S. Tan, A. Sordoni, and A. Courville, “Ordered Neurons: Integrating Tree Structures into Recurrent Neural Networks,” Oct. 2018.

[10] Ding Y, “Research on Multi-granular Text Classification of Hate Speech and Abusive Language based on RNN.” https://www.cnki.net (accessed Jan. 09, 2023).

[11] G. Liu and J. Guo, “Bidirectional LSTM with attention mechanism and convolutional layer for text classification,” *Neurocomputing*, vol. 337, pp. 325–338, Apr. 2019, doi: 10.1016/j.neucom.2019.01.078.

[12] H. Li, “A review of natural language processing based on RNN and Transformer models,” *Information Recording Material*, vol. 22, no. 12, pp. 7–10, 2021, doi: 10.16009/j.cnki.cn13-1295/tq.2021.12.081.

[13] Y. Xia, “A Review of the Development of Recurrent Neural Network,” *Computer Knowledge and Technology*, vol. 15, pp. 182–184, Jul. 2019, doi: 10.14004/j.cnki.ckt.2019.2379.

[14] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.

[15] J. Koutník, K. Greff, F. Gomez, and J. Schmidhuber, “A Clockwork RNN,” Feb. 2014, Accessed: Jan. 08, 2023. [Online]. Available: http://arxiv.org/abs/1402.3511

[16] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling,” Dec. 2014.

[17] F THE EUROPEAN PARLIAMENT AND OF THE COUNCIL, “General Data Protection Regulation,” *Official Journal of the European Union*, 2016.