HATE SPEECH DETECTION IN CHINESE LANGUAGE USING DEEP LEARNING

BY

HAZEL LIM BENIN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JUNE 2024

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: HATE SPEECH DETECTION IN CHINESE LANGUAGE USING DEEP LEARNING

Academic Session: JUNE 2024

I HAZEL LIM BENIN

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

__Hazel________ vicky

(Author's signature) (Supervisor's signature)

Address:
1171, JALAN SEKSYEN 1/2
31900 KAMPAR, PERAK.
____
____

Ts Dr Vikneswary a/p Jayapal
Supervisor's name

Date: 13/9/2024__ **Date**: ____13/9/24____

FACULTY/INSTITUTE* OF INFORMATION AND COMMUNICATIO TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 13/9/24

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that *HAZEL LIM BENIN* (ID No: *22ACB00107*) has completed this final year project/ dissertation/ thesis* entitled "Hate Speech Detection in Chinese Language using Deep Learning" under the supervision of Ts Dr Vikneswary a/p Jayapal (Supervisor) from the Faculty/Institute* of Information and Communication Technology.

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

____Hazel________
(*HAZEL LIM BENIN*)

*Delete whichever not applicable

DECLARATION OF ORIGINALITY

I declare that this report entitled "**Hate Speech Detection in Chinese Language using Deep Learning**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature	:	Hazel
Name	:	HAZEL LIM BENIN
Date	:	13/9/2024

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisors, Ts Dr Vikneswary a/p Jayapal who has given me this bright opportunity to engage in a machine learning design project. It is my first step to establish a career in machine learning design field. A million thanks to you.

Finally, I must say thanks to my parents and my family for their love, support, and continuous encouragement throughout the course.

ABSTRACT

In recent years, the rise of cyberbullying and online sexism has had devastating consequences, with Chinese social media platforms such as Sina Weibo and Zhihu seeing increased incidents of online harassment, leading to severe outcomes like suicide. To combat this, the project aims to develop deep learning models that effectively classify sexist content in Chinese social media. Despite extensive research on English-language cyberbullying detection, there is limited focus on Chinese contexts, particularly regarding sexism. This study utilizes the Sina Weibo Sexism Review (SWSR) dataset, evaluating several recurrent neural network (RNN) architectures, including RNN, LSTM, GRU, Bi-LSTM, Bi-GRU, RNN-LSTM, and RNN-GRU. These models were tested on balanced and imbalanced datasets, yielding accuracy rates between 74.2% and 76.8%. Precision, recall, and F1 scores ranged from 0.6818 to 0.7447, indicating strong classification performance. Moreover, incorporating emoji embeddings and English-Chinese translation further improved model accuracy and sensitivity in identifying sexist content. This research provides a significant contribution toward addressing online harassment in Chinese text, offering actionable insights for future cyberbullying detection systems.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

β	beta
α	alpha

LIST OF ABBREVIATIONS

CB	Cyberbullying
RNN	Recurrent neural network
LSTM	Long short-term memory
GRU	Gated recurrent unit
Bi-LSTM	Bidirectional Long short-tern memory
Bi-GRU	Bidirectional Gated recurrent network
NLP	Natural language processing
SWSR	Sina Weibo Sexism Review
TextCNN	Text Convolutional neural network
TextRNN	Text Recurrent neural network
TextRCNN	Text Region Convolutional Neural Network
TextRNN_Att	TextRNN combined with attention
DPCNN	Deep Pyramid Convolutional Neural Network
NB	Naïve Bayes
SVM	Support Vector Machine
LR	Logistic Regression
RF	Random Forest
XLNet	Extra-Large Neural Network
BERT	Bidirectional Encoder Representation from Transformers
RoBERTa	Robustly optimized BERT approach
ALBERT	A Lite BERT
ERNIE	Enhanced Representation through knowledge integration
LERT	Linguistically-motivated bidirectional Encoder Representation form Transformer
MacBERT	MLM as correction Bidirectional Encoder Representation from Transformers
ELECTTRA	Efficiently Learning an Encoder that Classifies Token Replacements Accurately
wwm	Whole Word Masking
TF-IDF	Term frequency-Inverse Document Frequency

ET	Extra tree
XLM	Cross-lingual language Model
GBDT	Gradient Boosting Decision Tree
SWSR	Sina Weibo Sexism Review Dataset
COLD	Chinese Offensive Language Dataset
SMOTE	Synthetic Minority Over-Sampling Technique

CHAPTER 1: Introduction

1.1 Problem Statement

1.1.1 Complexity of Analyzing Cyberbullying Due to Emojis and English Text in Chinese Comments.

The analysis of cyberbullying in Chinese comments is complicated by the integration of emojis and English text. Emojis in Chinese social media are often used to convey a range of emotions, reactions, or humor, but their meanings can vary greatly depending on context, cultural nuances, and individual interpretation. For instance, the “upside-down face” emoji might denote sarcasm or playfulness, while the “tearing face with joy” emoji typically signifies laughter. Additionally, English words and phrases are frequently interspersed within Chinese comments, driven by trends, brevity, and globalization. This mixing introduces challenges for automated analysis, as models must effectively handle code-switching between languages. For example, the comment “这个 idea 很 awesome!” (Translation: “This idea is awesome!”) includes the English word “awesome” alongside Chinese text, complicating the interpretation and analysis of such comments.

1.1.2 Limited Diversity in Detection Models for Chinese Sexism Dataset

The current approaches to detecting sexism in Chinese text suffer from a lack of model diversity. Different models possess distinct strengths and weaknesses; for instance, LSTMs are proficient at capturing long-term dependencies, while GRUs excel at handling short-term patterns. Employing a limited variety of models restricts the ability to fully address the challenges presented by the Chinese sexism dataset. A more diverse set of models is needed to provide a comprehensive view of the dataset and to capture various aspects of the data that a single model might miss. This diversity is crucial for developing robust detection mechanisms that can effectively identify and address sexist content in Chinese comments.

1.2 Research Objectives

The primary objective of this project is to investigate techniques for translating emojis and English text into Chinese to address the complexity introduced by multilingual content in cyberbullying analysis. The project also aims to develop and evaluate various deep learning models specifically designed for detecting sexism in Chinese. This involves enhancing the diversity of detection models applied to the Chinese sexism dataset and improving model performance. Additionally, the project seeks to integrate English-Chinese translation within the dataset to enable effective cross-lingual analysis. Finally, the project will incorporate emoji embeddings into deep learning models to refine the accuracy and effectiveness of detecting nuances related to sexism and cyberbullying in Chinese online content.

Based on the objective abovementioned:

1. To investigate techniques to translate Emoji and English into Chinese text.
2. To develop various deep learning models to detect Chinese sexism.
3. To evaluate the performance of various models in detecting Chinese sexism.

1.3 Motivation

In the vast landscape of Chinese social media, Sina Weibo stands as a prominent platform where millions of users engage in discussions, share opinions, and express themselves. However, beneath the seemingly innocuous posts lies a darker reality: cyberbullying. In early 2024, the nation mourned the loss of a young college student named Zhèng Llínghuá [26]. She fell victim to relentless online bullying, which ultimately led to her tragic suicide. The incident sparked outrage and highlighted the urge to address CB on social media platforms [7]. The awakening of feminism in Chinese society has intensified discussions around gender bias. On Sina Weibo, sexism emerges as a potent form of cyberbullying. Women are disproportionately targeted, facing attacks related to their appearance, relationships, and personal lives. Meanwhile, men encounter judgment based on morality and public opinions. The internet provides a protective layer for users to express themselves freely. Many individuals publish sexist comments online without fear of physical confrontation. This anonymity emboldens cyberbullies, allowing them to perpetuate harmful narratives. To combat sexism in online discourse, a groundbreaking project combines state-of-the-art deep learning algorithms with pre-trained word embeddings. By unmasking sexism and empowering accurate detection, this project aims to create a safer online environment—one where empathy prevails over cruelty.

1.4 Project Scope and Direction

The project's objective is to develop a model capable of classifying messages into two distinct categories: sexist and non-sexist. This will be achieved by employing a combination of neural networks for binary sentiment analysis and deep learning algorithms to categorize labeled comment texts. Binary sentiment analysis assigns numerical values (1 or 0) to classify text as positive or negative, respectively. Given that the task involves binary classification, the deep learning model is specifically trained to recognize negative sentiment as a potential indicator of sexist messages. Negative sentiment within a message is often associated with a higher likelihood of cyberbullying or sexist content.

The model will be trained using a substantial corpus of annotated text data, representing a supervised learning approach. Natural Language Processing (NLP) techniques will be used to convert text data into numerical vectors, making it comprehensible for the model and facilitating effective training. After training on the labeled data, the model will be capable of autonomously identifying the sentiment of new or unseen text inputs. By exposing the model to a diverse training set, it will learn to detect subtle textual cues, ensuring accurate classification.

The performance of the trained model will be evaluated using precision, accuracy, recall, and F1-measure. In the first phase, a reviewed model will be trained and deployed to serve as a benchmark for comparing methods not previously applied to the dataset. In the second phase, additional learning models will be thoroughly analyzed and tested to identify the best-performing model for detecting sexist hate speech. Finally, the hyperparameters of the models will be fine-tuned to determine the optimal values for enhancing performance.

1.5 Contributions

In this project, we employed various deep learning models, including RNN, LSTM, GRU, Bi-LSTM, and Bi-GRU, to detect sexism in Chinese text. Our integration of emoji embeddings and English-to-Chinese translation significantly advanced the accuracy of cyberbullying detection. The latest models demonstrated improved performance metrics, with the Bi-GRU model achieving an accuracy of 0.8198 and the RNN_GRU model reaching 0.8188. These results reflect the effectiveness of our enhancements in capturing the subtleties of sentiment and context, leading to a more robust detection system for Chinese online content.

1.6 Report Organization

The report is structured into seven chapters. Chapter 1, "Introduction," provides an overview of the project, detailing the problem statement, motivation, background, objectives, contributions, timeline, and structure. Chapter 2, "Literature Review," delves into existing research and methodologies related to cyberbullying detection, with a focus on machine learning models and relevant studies. In Chapter 3, "Proposed Approach," the report outlines the techniques for emoji translation, the deep learning models for detecting Chinese sexism, and the integration of English text into the Chinese dataset. Chapter 4, "System Design," describes the system architecture and design, including architecture diagrams and technical frameworks. Chapter 5, "Experiment/Simulation," covers the experimental setup, data preparation, and simulation processes, detailing model training, testing, and evaluation. Chapter 6, "System Evaluation and Discussion," analyzes model performance based on various metrics and discusses the results, effectiveness, and limitations of the methods. Finally, Chapter 7, "Conclusion," summarizes the findings, draws conclusions, and offers recommendations for future research. The appendices include weekly logs, code snippets, additional materials, the project poster, and plagiarism assessment results..

CHAPTER 2: Literature Reviews

2.1 Previous Works on Cyberbullying Detection

2.1.1 SWSR: A Chinese dataset and lexicon for online sexism detection

In this article, the authors aimed to create a Chinese Mandarin language Weibo dataset and lexicon focusing on sexism, which is a important task for natural language processing (NLP) and social media analysis [1],[33]. They used posts and comments form Sina Weibo that is China's most well-known microblogging platform. A total of 10,000 posts and comments from Sina Weibo were collected and annotated by using a fine-grained annotation scheme that covers different types of sexism such as, benevolent sexism, hostile sexism, sexual objectification and homophobia, and targets which can be either individual or collective. Sexism is defined as "any form of discrimination, prejudice or stereotyping based on gender or sexual orientation"[1] . They found that collective targets are more common than individual targets and gender wise women are more likely to be the victim of sexism than men.

` The authors used a crowdsourcing platform called Zhubajie to gather and annotate the data prior to training and testing the model [1]. It reported a high inter-annotator agreement of 0.87 using Krippendorff's alpha [1]. An exploratory analysis of the dataset characteristics such as the distribution of sexism labels, the linguistic features of sexist language, and the correlation between sexism and sentiment is provided by the paper too. The researchers conducted three sexism classification experiments using cutting-edge machine learning algorithms, including BERT and RoBERTa. These classifiers were evaluated across binary, multi-class, and multi-label classification tasks. The findings revealed that RoBERTa consistently outperformed BERT and other baseline models.

They also analyses the errors made by the models, and identifies some challenges and limitations of their approach. For example, the paper notes that some sexist expressions are subtle and implicit, and that some posts contain sarcasm or irony that are hard to detect. One of the limitation is that their dataset may not cover all forms of sexism, and that their lexicon may need to be updated frequently to capture the nuances of sexist language. To sum up, it can be claimed to be the first of its kind in the Chinese language and inspired more research in Chinese NLP.

2.1.2 A review of Deep Learning with Recurrent Neural Network

The paper provides a comprehensive overview of Recurrent Neural Networks (RNNs), highlighting their unique capabilities and challenges. RNNs are a type of deep learning model designed for handling sequential data by maintaining context over time[2]. This paper emphasizes that while traditional machine learning algorithms are well-defined, deep learning, with its hierarchical structure, manages more complex patterns, especially in sequential contexts. RNNs are particularly noted for their ability to process data through time, unlike feedforward networks[2]. However, they come with training difficulties and numerous parameters. Recent advancements, including Long Short-Term Memory (LSTM) and Bidirectional RNNs (BRNNs), have significantly improved performance in tasks such as image captioning, language translation, and handwriting recognition[2].

The paper contrasts RNNs with Artificial Neural Networks (ANNs) and Convolutional Neural Networks (CNNs)[2]. RNNs are superior for modeling sequences where data points are dependent on previous ones, while ANNs are more suited for independent data points. RNNs handle varying-length sequences naturally, unlike CNNs. Although MLPs can theoretically approximate any function, they struggle with long-term dependencies, which RNNs address more effectively[2].

2.1.3 Improving Sentiment analysis accuracy with emoji embedding

The paper addresses the challenge of sentiment analysis in Chinese text, focusing on the difficulty of accurately identifying and distinguishing emotions due to the complexity and variability of Chinese syntax and semantics[24]. To enhance sentiment analysis accuracy, the study incorporates emojis as a supplementary feature. Emojis, with their vast range of graphic symbols, are increasingly used in online conversations to express emotions[24]. The paper evaluates various sentiment analysis algorithms, including rule-based methods, classification algorithms, and deep learning models, to assess the impact of integrating emojis.

The study explores the effectiveness of different approaches, such as translating emojis into sentiment words and using emojis directly as features. It also considers the influence of emoji usage patterns on algorithm performance. The results indicate that incorporating emojis significantly improves sentiment analysis accuracy[24]. Specifically, the Bi-LSTM-based emoji-embedding model, named CEmo-LSTM, achieved the highest accuracy of around 0.95 in analyzing online Chinese texts[24].

2.1.4 COLD: A Benchmark for Chinese Offensive Language Detection

The paper introduces COLD, a benchmark for detecting offensive language in Chinese social media, addressing the challenge of scarce reliable datasets[4]. It presents COLDATASET, a dataset specifically curated for this purpose, and COLDETECTOR, a baseline model based on BERT architecture, trained on this dataset. COLDETECTOR achieves an 81% accuracy rate, outperforming other methods, including a public API from Baidu and a model trained on translated English data[10]. The study demonstrates that incorporating the COLD benchmark significantly enhances the accuracy of offensive language detection[10].

The research also explores various detection methods, including prompt-based self-detection and translated data models, and highlights the challenges faced by current models, such as misclassification of anti-bias content[10]. Despite its strong performance, COLDETECTOR struggles with specific text patterns that can lead to lower precision[10]. The paper calls for further research to improve detection capabilities, including manual annotation of datasets and refinement of models to handle covert offensiveness and anti-bias content more effectively[10].

2.1.5 When the Timeline Meets the Pipeline: A Survey on Automated Cyberbullying Detection

According to the survey paper, cyberbullying is characterized as “the utilization of digital communication to intimidate or threaten an individual, often through the transmission of intimidating or threatening messages.”[31]. Some common elements that are used to define cyberbullying are: intentionality, repetition, power imbalance, harm, and anonymity[17]. However, these elements are not always clear or consistent in online interactions, which makes cyberbullying detection a challenging task. This paper covered the definition, motivation, challenges, methods, datasets, evaluation metrics, applications, and future directions[17]. They concluded that not many datasets are publicly available for cyberbullying detection. Most of the existing datasets are collected from specific platforms such as Twitter or Instagram or from specific domains such as schools or universities[17]. Moreover, most of the datasets are imbalanced in terms of class distribution or language diversity[17].

Cyberbullying detection methods can be classified into three main categories:

1. Rule-Based Methods
2. Machine Learning-Based Methods
3. Deep Learning-Based Methods

There are some popular deep learning models for cyberbullying detection are networks CNNs, RNNs, LSTMs, GRUs, attention mechanisms, transformers, etc[3],[7],[17],[27],[29],[31].

Besides that, the efficacy and performance of cyberbullying detection techniques are assess using metrics [3],[17]. However, there is no standard or agreed-upon metric for this task. Different metrics may have different advantages and disadvantages depending on the attributes of the information and of the task [17].

2.2 Summary

<table>
<tr><th>Year (ref)</th><th>Title</th><th>Sexism</th><th>Platform</th><th>Language</th><th>Classifier</th><th>Performance Metrics</th></tr>
<tr>
<td>2021
[1]</td>
<td>SWSR: A Chinese dataset and lexicon for online sexism detection</td>
<td>Yes</td>
<td>Weibo</td>
<td>Chinese</td>
<td>1. LR + ngram
2. Char-LR+ ngram
3. SVM + ngram
4. Char-CNN+ft
5. Bert
6. Bert-wwm
7. RoBerta</td>
<td>1. Accuracy score
2. Macro F1 score
<table>
<tr><th></th><th>1</th><th>2</th></tr>
<tr><td>1</td><td>0.785</td><td>0.737</td></tr>
<tr><td>2</td><td>0.790</td><td>0.746</td></tr>
<tr><td>3</td><td>0.781</td><td>0.739</td></tr>
<tr><td>4</td><td>0.774</td><td>0/749</td></tr>
<tr><td>5</td><td>0.787</td><td>0.753</td></tr>
<tr><td>6</td><td>0.806</td><td>0.776</td></tr>
<tr><td>7</td><td>0.792</td><td>0.763</td></tr>
<tr><td>8</td><td>0.792</td><td>0.764</td></tr>
</table>
</td>
</tr>
<tr>
<td>2024
[3]</td>
<td>Chinese Cyberbullying Detection Using XLNet and Deep Bi-LSTM Hybrid Model [3]</td>
<td>No</td>
<td>Weibo and Zhihu</td>
<td>Chinese</td>
<td>Machine Learning:
1. NB
2. SVM
3. LR
4. RF
Deep Learning:
1. TextCNN
2. RNN
3. GRU
4. LSTM
5. Bi-GRU
6. Bi-LSTM
Pre-trained Models:
1. BERT
2. XLNet
3. RoBERTa
4. ALBERT
5. ERNIE3.0
6. LERT
7. MacBERT
8. ELECTRA
9. Proposed</td>
<td>1. Precision
2. Recall
3. F1-score
<table>
<tr><th></th><th>1</th><th>2</th><th>3</th></tr>
<tr><td>1</td><td>0.7742</td><td>0.7328</td><td>0.8067</td></tr>
<tr><td>2</td><td>0.8517</td><td>0.8752</td><td>0.8623</td></tr>
<tr><td>3</td><td>0.8669</td><td>0.8392</td><td>0.8528</td></tr>
<tr><td>4</td><td>0.8669</td><td>0.8392</td><td>0.8528</td></tr>
<tr><td>5</td><td>0.8417</td><td>0.8572</td><td>0.8494</td></tr>
<tr><td>6</td><td>0.8762</td><td>0.9060</td><td>0.8909</td></tr>
<tr><td>7</td><td>0.6398</td><td>0.1080</td><td>0.1848</td></tr>
<tr><td>8</td><td>0.8778</td><td>0.9048</td><td>0.8911</td></tr>
<tr><td>9</td><td>0.8658</td><td>0.9084</td><td>0.8866</td></tr>
<tr><td>10</td><td>0.8696</td><td>0.9096</td><td>0.8891</td></tr>
<tr><td>11</td><td>0.8754</td><td>0.8884</td><td>0.8819</td></tr>
<tr><td>12</td><td>0.8848</td><td>0.9036</td><td>0.8914</td></tr>
<tr><td>13</td><td>0.8837</td><td>0.9180</td><td>0.9005</td></tr>
<tr><td>14</td><td>0.8945</td><td>0.8956</td><td>0.8951</td></tr>
<tr><td>15</td><td>0.8735</td><td>0.8676</td><td>0.8706</td></tr>
<tr><td>16</td><td>0.8777</td><td>0.9096</td><td>0.8933</td></tr>
<tr><td>17</td><td>0.8734</td><td>0.9192</td><td>0.8957</td></tr>
<tr><td>18</td><td>0.8989</td><td>0.8964</td><td>0.8977</td></tr>
<tr><td>19</td><td>0.8765</td><td>0.9116</td><td>0.8937</td></tr>
<tr><td>20</td><td>0.8988</td><td>0.9112</td><td>0.9050</td></tr>
</table>
</td>
</tr>
<tr>
<td>2023
[3]</td>
<td>Research on Chinese Text Classification Based on Improved RNN</td>
<td>No</td>
<td>THUCNews</td>
<td>Chinese</td>
<td>1. TextRNN
2. TextCNN
3. TextRCNN
4. TextRNN_Att
5. Transformer
6. DPCNN</td>
<td>1. Precision
2. F-measure
3. Recall
4. Accuracy
<table>
<tr><th></th><th>1</th><th>2</th><th>3</th><th>4</th></tr>
<tr><td>1</td><td>0.9035</td><td>0.9028</td><td>0.9026</td><td>0.9026</td></tr>
<tr><td>2</td><td>0.9102</td><td>0.9101</td><td>0.9102</td><td>0.9102</td></tr>
<tr><td>3</td><td>0.9120</td><td>0.9108</td><td>0.9108</td><td>0.9108</td></tr>
<tr><td>4</td><td>0.9084</td><td>0.9076</td><td>0.9078</td><td>0.9078</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td><table><tr><td>5</td><td>0.899
0</td><td>0.897
6</td><td>0.897
1</td><td>0.897
1</td></tr><tr><td>6</td><td>0.915
8</td><td>0.915
4</td><td>0.915
5</td><td>0.915
5</td></tr></table></td></tr>
<tr><td>2021[4
]</td><td>Abusive Language detection from social media comments using conventional machine learning and deep learning approaches</td><td>No</td><td>YouTube</td><td>Urdu and Roman Urdu</td><td>Machine Learning
1. NB
2. SVM
3. IBK
4. Logistic
5. JRip
Deep learning
6. CNN
7. LSTM
8. BLSTM
9. CLSTM</td><td>1. F-measure<table><tr><td></td><td>1</td></tr><tr><td>1</td><td>0.88</td></tr><tr><td>2</td><td>0.901</td></tr><tr><td>3</td><td>0.807</td></tr><tr><td>4</td><td>0.846</td></tr><tr><td>5</td><td>0.867</td></tr><tr><td>6</td><td>0.962</td></tr><tr><td>7</td><td>0.0.911</td></tr><tr><td>8</td><td>0.921</td></tr><tr><td>9</td><td>0.943</td></tr></table></td></tr>
<tr><td>2020
[5]</td><td>Cyberbullying Detection using Pre-Trained BERT Model</td><td>No</td><td>Formspring and Wikipedia</td><td>English</td><td>1. Pre-Trained BERT model with a Single Linear Neutral Network Layer(1,2,3 times)</td><td>1. Precision
2. Recall
3. F1-score<table><tr><td></td><td>1</td><td>2</td><td>3</td></tr><tr><td>1</td><td>0.64</td><td>0.55</td><td>0.59</td></tr><tr><td>2</td><td>0.82</td><td>0.91</td><td>0..86</td></tr><tr><td>3</td><td>0.90</td><td>0.99</td><td>0.94</td></tr></table></td></tr>
<tr><td>2023
[6]</td><td>A Comparative Study of Cyberbullying Detection in Social Media for Last Five Years</td><td>No</td><td>Twitter, Facebook, YouTube, Instagram</td><td>Arabic and English</td><td>Machine Learning:
1. SVM
2. NB
3. RF
4. XG-Boost
Deep learning:
1. CNN
2. RNN
3. GRU
4. BiLSTM</td><td>1. Accuracy
2. F1-Score</td></tr>
<tr><td>2023
[7]</td><td>A Review on Deep-Learning Based Cyberbullying Detection</td><td>No</td><td>-</td><td>-</td><td>1. DNN
2. BMs
3. DBN
4. DAE
5. GAN
6. RNN
7. LSTM
8. Bo-LSTM
9. CNN
10. RBFNs
11. MLPs
12. SOMs
13. RBMs
14. GRU
15. Attention-based Model</td><td>1. Accuracy
2. Precision
3. Recall
4. F1-score
5. MCC
6. AUC-ROC</td></tr>
<tr><td>2022
[8]</td><td>An Application to Detect Cyberbullying Using Machine Learning and Deep Learning Techniques</td><td>No</td><td>Twitter</td><td>English, Hindi, Hinglish</td><td>1. CNN-BiLSTM
2. CNN_BIGRU
3. BILSTM+BIGRU</td><td>1. Accuracy<table><tr><td></td><td>1</td></tr><tr><td>1</td><td>0.9512</td></tr><tr><td>2</td><td>0.9369</td></tr><tr><td>3</td><td>0.8853</td></tr></table></td></tr>
<tr><td>2022
[9]</td><td>COLD: A Benchmark for Chinese Offensive Language Detection</td><td>Racial, gender, and regional Bias</td><td>Weibo and Zhihu</td><td>Chinese</td><td>1. COLDetector
2. Baidu Text censor
3. Prompt-based Self-Detection
4. TranslJigsaw Detector
5. Random Setting</td><td>1. Accuracy
2. Precision
3. Recall
4. F1-score<table><tr><td></td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>1</td><td>0.81</td><td>0.80</td><td>0.82</td><td>0.81</td></tr><tr><td>2</td><td>0.63</td><td>0.61</td><td>0.56</td><td>0.54</td></tr><tr><td>3</td><td>0.59</td><td>0.58</td><td>0.57</td><td>0.57</td></tr><tr><td>4</td><td>0.60</td><td>0.62</td><td>0.62</td><td>0.60</td></tr></table></td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td><table><tr><td>5</td><td>0.50</td><td>0.50</td><td>0.50</td><td>0.49</td></tr></table></td></tr>
<tr><td>2023
[10]</td><td>Cyberbullying Detection on Social Media Using Stacking Ensemble Learning and Enhanced BERT</td><td>Offensiveness, racism, discrimination, abusive language, threathening.</td><td>Twitter, Facebook</td><td>English</td><td>1. Conv1DLSTM
2. BiLSTM
3. LSTM
4. CNN
5. BERT
6. Tuned-BERT
7. Proposed Stacked</td><td>1. Accuracy
2. F1-score
3. Precision
4. Recall<table><tr><td></td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>1</td><td>0.8649</td><td>0.8317</td><td>0.8146</td><td>0.8919</td></tr><tr><td>2</td><td>0.7795</td><td>0.8041</td><td>0.8373</td><td>0.8130</td></tr><tr><td>3</td><td>0.8011</td><td>0.8281</td><td>0.8142</td><td>0.7281</td></tr><tr><td>4</td><td>0.8496</td><td>0.8720</td><td>0.8836</td><td>0.7908</td></tr><tr><td>5</td><td>0.921</td><td>0.9149</td><td>0.915</td><td>0.915</td></tr><tr><td>6</td><td>0.9384</td><td>0.92</td><td>0.92</td><td>0.91</td></tr><tr><td>7</td><td>0.974</td><td>9.964</td><td>0.950</td><td>0.92</td></tr></table></td></tr>
<tr><td>2020
[11]</td><td>Revisiting Pre-trained Models for Chinese Natural Language Processing</td><td>No</td><td>Chinese Wikipedia</td><td>Chinese</td><td>1. BERT
2. BERT-wwm
3. BERT-wwm-ext
4. RoBERTa-wwm-ext
5. ELECTRA-base
6. MacBERT-base
7. ELECTRA-large
8. RoBERTa-wwm-ext-large
9. MacBERT-large</td><td>1. EM
2. F1-score<table><tr><td></td><td>1</td><td>2</td></tr><tr><td>1</td><td>70.0</td><td>87.0</td></tr><tr><td>2</td><td>70.5</td><td>87.4</td></tr><tr><td>3</td><td>71.4</td><td>87.7</td></tr><tr><td>4</td><td>72.6</td><td>89.4</td></tr><tr><td>5</td><td>73.1</td><td>87.1</td></tr><tr><td>6</td><td>73.2</td><td>89.5</td></tr><tr><td>7</td><td>73.9</td><td>87.1</td></tr><tr><td>8</td><td>74.2</td><td>90.6</td></tr><tr><td>9</td><td>74.8</td><td>90.7</td></tr></table></td></tr>
<tr><td>2022
[12]</td><td>Factor Detection Task of Cyebrbullying Using the Deep Learning Model</td><td>No</td><td>PTT forum</td><td>Chinese</td><td>1. BERT_α
2. BERT_β
3. SVM
4. RFC</td><td>1. Macro F1
2. Macro Precision
3. Macro Recall</td></tr>
<tr><td>2023
[13]</td><td>The Use of Large Language Model for Cyberbullying Detection</td><td>No</td><td>Formspring and Twitter</td><td>English</td><td>Case 1:
1. TF-IDF+RF
2. TF-IDF+SVM
3. BERT
4. XLNet
5. RoBERTa
6. XLM-RoBERTa
Case2:
1. BERT
2. XLNet
3. RoBERTa
4. XLM-RoBERTa
5. SBERRT+SVM
6. SBERT+RF
7. TF-IDF+SVM
8. TF-IDF+RF</td><td>1. Accuracy
2. Precision
3. Recall
4. Macro F1-Score<table><tr><td></td><td>1</td></tr><tr><td>1</td><td>0.80</td></tr><tr><td>2</td><td>0.84</td></tr><tr><td>3</td><td>0.96</td></tr><tr><td>4</td><td>0.95</td></tr><tr><td>5</td><td>0.95</td></tr><tr><td>6</td><td>0.94</td></tr><tr><td>7</td><td>0.85</td></tr><tr><td>8</td><td>0.86</td></tr><tr><td>9</td><td>0.87</td></tr><tr><td>10</td><td>0.86</td></tr><tr><td>11</td><td>0.85</td></tr></table></td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td><table><tr><td>12</td><td>0.81</td></tr><tr><td>13</td><td>0.84</td></tr><tr><td>14</td><td>0.84</td></tr></table></td></tr>
<tr><td>2021
[14]</td><td>Cyberbullying detection: advanced preprocessing techniques & deep learning architecture for Roman Urdu data</td><td>NO</td><td>Twitter</td><td>Roman Urdu</td><td>1. RNN-LSTM
2. RNN-BiLSTM
3. CNN</td><td>1. Accuracy
2. Macro average
3. Weighted Average
4. Approximate Loss<table><tr><td></td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>1</td><td>0.85</td><td>0.80</td><td>0.85</td><td>0.6</td></tr><tr><td>2</td><td>0.84</td><td>0.78</td><td>0.84</td><td>1.2</td></tr><tr><td>3</td><td>0.79</td><td>0.69</td><td>0.78</td><td>1.7</td></tr></table></td></tr>
<tr><td>2021
[15]</td><td>A Multichannel Deep Learning Framework for Cyberbullying Detection on Social Media</td><td>Offensive, non-offensive</td><td>Twitter</td><td>English</td><td>1. Proposed Multichannel method
2. CNN
3. BiGRU
4. Transformer Block
5. linear SVC
6. Bagging
7. LR
8. RF
9. ET</td><td>1. Accuracy<table><tr><td></td><td>1</td></tr><tr><td>1</td><td>0.8799</td></tr><tr><td>2</td><td>0.8743</td></tr><tr><td>3</td><td>0.8728</td></tr><tr><td>4</td><td>0.8699</td></tr><tr><td>5</td><td>0.5033</td></tr><tr><td>6</td><td>0.6830</td></tr><tr><td>7</td><td>0.5427</td></tr><tr><td>8</td><td>0.6929</td></tr><tr><td>9</td><td>0.6562</td></tr></table></td></tr>
<tr><td>2021
[24]</td><td>Improving sentiment analysis accuracy with emoji embedding</td><td>-</td><td>Weibo</td><td>Chinese</td><td>1. LR
2. SVM
3. NB
4. GBDT
5. LSTM
6. BERT
7. Cemo-LSTM</td><td>1. Accuracy<table><tr><td></td><td>1</td></tr><tr><td>1</td><td>0.727</td></tr><tr><td>2</td><td>0.681</td></tr><tr><td>3</td><td>0.709</td></tr><tr><td>4</td><td>0.720</td></tr><tr><td>5</td><td>0.790</td></tr><tr><td>6</td><td>0.762</td></tr><tr><td>7</td><td>0.811</td></tr></table></td></tr>
<tr><td>2021
[25]</td><td>Chinese Text Classification Based on ERNIE-RNN</td><td>-</td><td></td><td>Chinese</td><td>1. DPCNN
2. FastText
3. TextCNN
4. TextRNN
5. TextRCNN
6. Bert
7. Bert_RNN
8. BERT_RCNN
9. ERNIE_DPCNN
10. ERNIE_RNN</td><td>1. Loss
2. Accuracy<table><tr><td></td><td>1</td><td>2</td></tr><tr><td>1</td><td>0.31</td><td>92.16</td></tr><tr><td>2</td><td>0.26</td><td>93.79</td></tr><tr><td>3</td><td>0.25</td><td>93.73</td></tr><tr><td>4</td><td>0.29</td><td>93.47</td></tr><tr><td>5</td><td>0.25</td><td>93.82</td></tr><tr><td>6</td><td>0.27</td><td>94.11</td></tr><tr><td>7</td><td>0.35</td><td>93.12</td></tr><tr><td>8</td><td>0.30</td><td>93.73</td></tr><tr><td>9</td><td>0.32</td><td>93.44</td></tr><tr><td>10</td><td>0.24</td><td>94.63</td></tr></table></td></tr>
</table>

Table 2.1: Comparison of methodologies

2.3 Inspiration from related work

Recent research into offensive language detection and sentiment analysis in Chinese social media has made significant progress, exemplified by the development of datasets and models such as SWSR for sexism detection and COLD for offensive language analysis. These studies have utilized advanced techniques like RoBERTa and BERT-based models to improve classification accuracy, though challenges remain, such as detecting subtle biases and updating lexicons. Additionally, the integration of emoji embeddings, as seen with the CEmo-LSTM model, highlights the potential for enhancing sentiment analysis by incorporating contextual features. Despite these advancements, ongoing research is needed to address the complexities of language, handle nuanced expressions, and refine detection models to better capture the intricacies of online communication.

CHAPTER 3: Proposed Approach

The project's development process involved several distinct phases, including data pre-processing, data splitting, word embedding, model architecture building, and data training and testing. In this chapter, the methodology is first discussed, then continuing to the tools, system performance definition and system design.

3.1 System Design Diagram

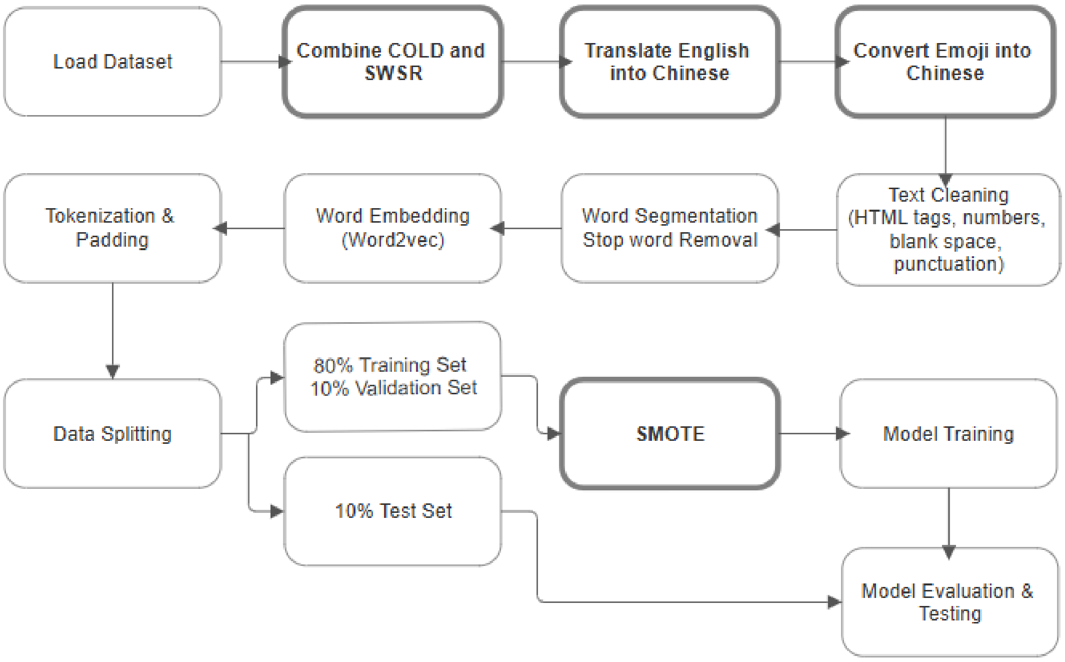


Figure 3.1 Illustration of System Design Diagram

As illustrated in Figure 3.1, the outlined boxes highlight the various combinations explored in the proposed system. These include comparisons between using both the COLD and SWSR datasets versus the SWSR dataset alone, applying language and emoji translation versus no translation, and utilizing SMOTE versus not using it. The system design begins with loading the datasets, followed by combining the two datasets to address class imbalances. To ensure the relevance of the comment texts, English words are detected and translated into Mandarin, while emojis are converted into corresponding Chinese descriptions. This process enriches the data by incorporating additional context from non-Chinese comments.

Moving forward with data preprocessing, the text undergoes cleaning to eliminate irrelevant elements such as HTML tags, numbers, extra spaces, and punctuation. If these unnecessary components are not removed, they could introduce noise during classification. After cleaning, the text is prepared for word segmentation and stop-word removal. The stop-word removal process ensures that only significant information is retained by filtering out common, insignificant words.

After stop word removal, the cleaned text proceeds to word segmentation, a critical step in Chinese language processing where continuous text is split into individual words. This is particularly important in languages like Chinese, where words are not inherently separated by spaces. Once segmented, the text is prepped for word embedding using Word2Vec, which transforms words into numerical vectors that capture semantic meaning, making them usable for machine learning algorithms.

Next, the text undergoes tokenization and padding, converting sentences into sequences of tokens (i.e., numerical representations) and padding them to ensure consistent input lengths for the model. After this step, the data is split into training (80%), validation (10%), and testing (10%) sets to allow for proper evaluation of the model's performance.

To address the issue of class imbalance in the datasets, SMOTE (Synthetic Minority Over-sampling Technique) is applied to oversample the minority class. By generating synthetic examples, SMOTE helps prevent the model from being biased towards the majority class.

Finally, the model training phase begins, where the dataset is used to train the classification model. Once trained, the model's performance is evaluated using the validation and testing datasets to ensure accuracy and generalizability. The evaluation involves metrics such as precision, recall, and F1 score, providing insight into how well the model performs on unseen data.

This proposed approach introduces variability in the data preparation phase by experimenting with combining datasets, incorporating language and emoji translation, and applying SMOTE or not. The goal is to find the most effective configuration for detecting offensive or sexist content in Chinese comments.

3.1.1 Data Acquisition and Labelling

The SWSR (Sina Weibo Sexism Review) dataset in consists of Chinese comments that have been categorized into various types of sexism, including Sexual Assault (SA), Sexual Coercion and Blackmail (SCB), Misogynistic Abuse (MA), and Sexual Objectification (SO)[1].It encompasses both binary and multi-class classifications, distinguishing between sexist and non-sexist texts, as well as categorizing texts into one of five sexism categories or three target classifications (Individual and group). In total, the dataset includes 8,969 comments associated with 1,527 Weibo's. The SWSR dataset comprises two tables containing Weibo and comment data. These tables are supplemented with anonymized user details, including gender and location. The dataset statistics reveal that 34.5% (3093) of the data corresponds to sexist content, while 65.6% (5876) represents non- sexist content.[1]. To ensure privacy, all personally identifiable information, including usernames and mentions, has been removed and remains undisclosed [1].

The COLD dataset (Chinese Offensive Language Dataset) is a comprehensive collection designed to identify and categorize offensive language, including hate speech, profanity, and abusive content[4]. It contains textual data derived from various Chinese social media platforms, such as Weibo, and has been labeled into categories that align with offensive language taxonomy. This dataset aims to capture both binary and multi-class classifications, differentiating offensive from non-offensive comments, and further classifying them into subcategories such as hate speech, vulgar language, and targeted insults[4].

Similar to the SWSR dataset, the COLD dataset also undergoes thorough anonymization to safeguard user identities. All personal data, including usernames and direct references, is removed to maintain user privacy. This dataset is especially valuable for research aimed at examining the frequency of hate speech and offensive language in online conversations. It comprises thousands of comments, with a significant portion marked as offensive. Researchers can utilize this resource to analyze toxic language patterns in Chinese social media and develop models for automatically detecting such content. We specifically extract comments related to the topic of gender from the COLD dataset and combine them with the SWSR dataset. Both datasets are crucial for advancing natural language processing (NLP) technologies, particularly in sentiment analysis and detecting offensive language in Chinese contexts. The COLD dataset contains a total of 9787ows, while the SWSR dataset has 8,969 rows ending up a total of 18756 comments.

3.1.2 Data Evaluation

The primary sexism dataset, known as SWSR, is imbalanced, which can affect the accuracy of models trained on it. To address this issue, we extract sexism-related comments from the COLD dataset and combine them with the SWSR dataset, creating a larger and more balanced Chinese sexism dataset. The COLD dataset contributes 9,787 sexism-related comments, bringing the combined dataset to a total of 18,756 entries.

In addition to this, we apply a technique called SMOTE (Synthetic Minority Over-sampling Technique) to further enhance the dataset. SMOTE generates synthetic examples for the minority class in an imbalanced dataset by creating new samples between existing data points, rather than duplicating them. This method helps balance the dataset and improves model performance by providing more representative data for the minority class, allowing for a more effective comparison against the original, imbalanced dataset. As shown in Figure 3.2 and 3.3, before combining both dataset the initial dataset consist of 65.5% of non-sexist comments and 34.5% of sexist comments. After combining them, there are 57.9% of non-sexist comments and 42.1% sexist comments. A more balanced dataset is obtained.

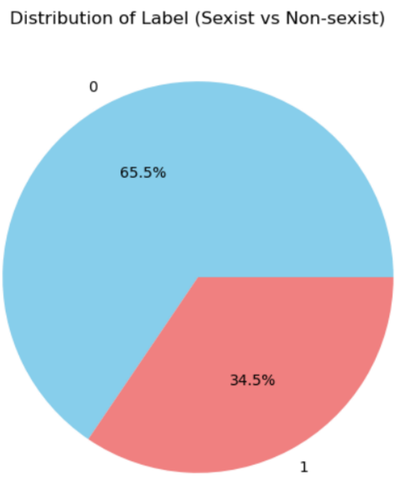


Figure 3.2 Before combining COLD with SWSR

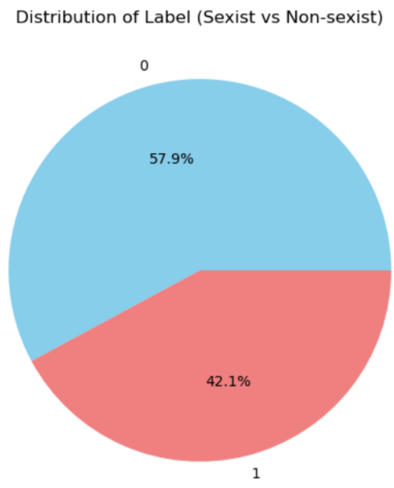


Figure 3.3 After Combining COLD and SWSR

3.1.3 Data Preprocessing

Data preprocessing in deep learning is a crucial step that involves transforming raw data into a format that can be effectively used by neural networks. The quality of data preprocessing can significantly impact the model’s performance, ensuring that the model can learn from the data efficiently.

Translate English words into Chinese meaning using

The technique used for language translation in this project involves the GoogleTranslator class from the deep_translator library. This method leverages Google's translation API to convert detected English words into Chinese. The GoogleTranslator function provides a straightforward interface for translating text, allowing for efficient and accurate language conversion. By utilizing this tool, the translation process benefits from Google's extensive language database and translation capabilities, ensuring high-quality translations for effective multilingual text processing.

```
Original Text: 孩子改母姓，还有在学校被bully的风险？这种合理化bully的言论是不是有点失智？
Translated Comment: 孩子改母姓，还有在学校被欺负的风险？这种合理化欺负的言论是不是有点失智？
Processed Comment: 孩子改母姓，还有在学校被欺负的风险？这种合理化欺负的言论是不是有点失智？
Cleaned Comment: 孩子改母姓还有在学校被欺负的风险这种合理化欺负的言论是不是有点失智
Segmented Text: 孩子 改母 姓 学校 欺负 风险 这种 合理化 欺负 言论 是不是 有点 失智
```

Figure 3.4 Example of English Translation

The figure provided displays the original text is in Chinese and contains the English word "bully" within the sentence. The text is asking about the risks associated with a child taking their mother's surname and whether such risks include being bullied at school. The next step, the English word "bully" in the original text has been translated to the Chinese word "欺负" (which means "bully" or "harass"). The translated comment replaces the English term with its Chinese equivalent, making the entire text consistent in language.

```python
# Function to process the dataset in parallel
def process_large_dataset(dataset):
    with concurrent.futures.ThreadPoolExecutor() as executor:
        results = list(executor.map(process_comment, dataset))
    return results
```

Figure 3.5 Code to handle parallel processing

The function also highlights the use of parallel processing for efficiency when handling large datasets. By utilizing a thread pool executor, the comments are processed simultaneously, allowing the program to handle a larger volume of data in less time. Each comment is processed individually by detecting English words, translating them, and then replacing the original words with their Chinese translations. This method ensures that only English words are translated, preserving the integrity of the original Chinese content and providing a seamless integration of the translated elements.

Convert Emojis into Chinese Description

The technique used for emoji translation in this project involves converting emojis into their corresponding Chinese descriptions. This is achieved using the demojize function from the emoji library, which translates emojis into text-based representations. This approach is essential for ensuring that emojis, which can convey significant meaning in text, are accurately understood and processed in the context of Chinese language data. By replacing emojis with descriptive text, the model can better analyze and classify text messages, leading to more accurate and comprehensive results in text classification tasks.

```
Original Text: 对，郑英俊就是放到现在也是理想型啊。一直以为他是演技派，却没想到他居然做了这样的事😔😔😔😔
Translated Comment: 对，郑英俊就是放到现在也是理想型啊。一直以为他是演技派，却没想到他居然做了这样的事😔😔😔😔
Processed Comment: 对，郑英俊就是放到现在也是理想型啊。一直以为他是演技派，却没想到他居然做了这样的事:忧郁的脸::忧郁的脸::忧郁的脸::忧郁的脸:
Cleaned Comment: 对郑英俊就是放到现在也是理想型啊一直以为他是演技派却没想到他居然做了这样的事忧郁的脸忧郁的脸忧郁的脸忧郁的脸
Segmented Text: 郑 英俊 放到 现在 理想 型 一直 演技派 没想到 居然 事 忧郁 脸 忧郁 脸 忧郁 脸 忧郁 脸
```

Figure 3.6 Example of Converting Emojis into Chinese Description

In the provided example, the original comment in Chinese, which includes both text and emojis. The emojis used here are "😔", representing a sad or melancholic face. Later on, the processed comment demonstrates the replacement of the emojis in the original text with their Unicode emoji descriptions. For instance, the sad face emoji "😔" is replaced with ":忧郁的脸:" (meaning "sad face"). This transformation aims to provide a descriptive representation of emojis, which can be useful for textual analysis or ensuring that all emojis are consistently described in text data.

The benefits of this conversion extend to improved sentiment detection and overall text comprehension. Emojis, when translated into their Chinese descriptions, provide deeper insight into the emotional tone of the text, helping models make more accurate predictions. This is particularly useful in user-generated content, where emojis are frequently used to express feelings. Furthermore, converting emojis into text ensures that they are processed uniformly with the rest of the input, avoiding the risk of them being ignored or misinterpreted by the model. Ultimately, this step enhances the model’s ability to understand and analyze complex text data that includes both language and symbols.

Text Cleaning

Text cleaning is an essential preprocessing step in deep learning, especially for natural language processing (NLP) tasks. The purpose of text cleaning is to remove noise from the input data, ensuring that the model can focus on the meaningful aspects of the text. In the provided code, cleaning involves eliminating HTML tags, numbers, and special characters, which may not contribute to the task at hand. This process helps standardize the text, making it easier for deep learning models to recognize patterns and extract relevant features.

```
-------------------------------------------------
Original Text: 哪里像段子????????你们自己心态不正，觉得性骚扰就像段子吧!
Translated Comment: 哪里像段子????????你们自己心态不正，觉得性骚扰就像段子吧!
Processed Comment: 哪里像段子????????你们自己心态不正，觉得性骚扰就像段子吧!
Cleaned Comment: 哪里像段子你们自己心态不正觉得性骚扰就像段子吧
Segmented Text: 段子 心态 不正 觉得 性骚扰 段子
-------------------------------------------------
```

Figure 3.7 Example of Punctuation Removal

For instance, in Figure 3.7, punctuation removal is demonstrated by stripping out characters like question marks, which are not essential for understanding the text's meaning.

```
Original Text: 才几岁哦，找28岁的男的? 还没毕业，我的天呐，这种女孩子能不能少一点。真的想不
通，就没想过这房子是不是你的，还帮他还钱，让她还吧，别后悔就行，甜甜的爱情真是上头。
Translated Comment: 才几岁哦，找28岁的男的? 还没毕业，我的天呐，这种女孩子能不能少一点。真的
想不通，就没想过这房子是不是你的，还帮他还钱，让她还吧，别后悔就行，甜甜的爱情真是上头。
Processed Comment: 才几岁哦，找28岁的男的? 还没毕业，我的天呐，这种女孩子能不能少一点。真的
想不通，就没想过这房子是不是你的，还帮他还钱，让她还吧，别后悔就行，甜甜的爱情真是上头。
Cleaned Comment: 才几岁哦找岁的男的还没毕业我的天呐这种女孩子能不能少一点真的想不通就没想过
这房子是不是你的还帮他还钱让她还吧别后悔就行甜甜的爱情真是上头
Segmented Text: 几岁 找 岁 男 没 毕业 天呐 这种 女孩子 不能 少 一点 真的 想不通 没想 房子
是不是 帮 钱 后悔 就行 甜甜的 爱情 真是 上头
```

Figure 3.8 Example of Number Removal

Similarly, Figure 3.8 illustrates number removal, where numerical digits are eliminated to avoid interference with text-based analysis.

```
Original Text: <username>法国女性独立性一直是比较强的，没有可比性
Translated Comment: <用户名>法国女性独立性一直是比较强的，没有可比性
Processed Comment: <用户名>法国女性独立性一直是比较强的，没有可比性
Cleaned Comment: 法国女性独立性一直是比较强的没有可比性
Segmented Text: 法国 女性 独立性 一直 比较 强 没有 可比性
```

Figure 3.9 Example of Tag Removal

Figure 3.9 showcases tag removal, which involves removing HTML or XML tags that might be present in the text, ensuring that the focus remains on the actual content rather than formatting details.

By applying these cleaning techniques, the text data is simplified, making it more uniform and easier for machine learning algorithms to process. Removing punctuation, numbers, and tags helps in reducing noise and standardizing the text, which can enhance the accuracy of models used for natural language processing tasks. This preprocessing step is crucial because it ensures that the data fed into the deep learning models is clean and relevant, leading to more effective and reliable outcomes.

Word Segmentation & Stop Word Removal

Figure 3.10 Word Cloud Before Stop Word Removal

Figure 3.11 Word Cloud After Stop Word Removal

Chinese stopword removal and word segmentation are critical preprocessing steps in natural language processing (NLP) for Chinese text. Unlike English, where words are typically separated by spaces, Chinese text lacks such clear delimiters. This absence of spaces makes word segmentation essential, as it involves breaking down a continuous string of characters into discrete words or phrases. Tools like jieba employ algorithms and dictionaries to identify these word boundaries, transforming the text into a format that is easier to analyze and process.

After segmentation, stopword removal further refines the text. Stopwords are common words, such as "的" (de), "了" (le), or "是" (shi), that, while grammatically significant, do not contribute meaningful information to the analysis. Removing these stopwords helps in focusing on the more informative parts of the text. This process not only improves the accuracy of subsequent NLP tasks—such as sentiment analysis and topic modeling—but also enhances computational efficiency by reducing the amount of noise in the dataset. By filtering out irrelevant words, the analysis can reveal more precise insights and patterns, making the results more actionable and relevant. Unlike English, where word boundaries are naturally indicated by spaces, Chinese text requires additional processing steps to achieve similar outcomes, highlighting the unique challenges of handling Chinese language data.

3.1.4 Word Embedding

In natural language processing (NLP), preparing text data for machine learning models often involves transforming raw text into numerical representations that algorithms can understand. For Chinese text classification, this process includes training a Word2Vec model, which is a popular technique for word embedding. Word2Vec converts words into dense vector representations, capturing semantic relationships and contextual meanings based on their usage in large text corpora. The model's parameters, such as the vector size, context window size, minimum word count, and sampling rate, are essential for controlling the quality and efficiency of the embeddings. For instance, a larger vector size can capture more nuanced meanings but requires more computational resources, while a smaller context window might miss broader contextual relationships between words.

```
Original text: 我觉得这就是对一类人所有的特点进行形容吧，“爹味”也是贬义词，为什么不觉得有性
别歧视意味呢?
Tokenized text: 觉得 一类 特点 进行 形容 爹味 贬义词 觉得 性别歧视 意味
```

Figure 3.12 Tokenization

(I) (have) (a) (camera)(.) (我)(有)(一台)(相机)(。)			
Token 1	Token 2	Token 3	Token 4
我	有	一台	相机

Table 3.1 Tokenized Chinese Words

Training a Word2Vec model on segmented Chinese text involves first tokenizing the text, which breaks it down into individual words or phrases. This tokenization is crucial for the model to learn from meaningful units rather than arbitrary sequences. After tokenization, the Word2Vec model is trained using parameters like vector size and context window to generate word embeddings that encode semantic information about words. The embeddings can then be used as features in text classification tasks, such as sentiment analysis or topic classification. This approach is particularly beneficial for Chinese text due to the language's unique characteristics, such as the lack of spaces between words and the importance of context in understanding meaning. By converting text into dense vectors, Word2Vec allows machine learning models to leverage semantic relationships between words, improving their ability to classify and analyze Chinese text effectively.

In addition, word segmentation is done by an open source package call Jieba. The advantage of using Jieba is its fast processing and lightweight. By referring Table 3.4, there is word structure differences between English and Chinese text messages. In Chinese more than one character can represent a word, while in English most of the times one word represents its meaning itself.

After segmenting the texts, they are ready to be tokenized. Tokenization involves breaking up text data into smaller units known as tokens. This process is crucial in Natural Language Processing (NLP) aspect because its reduction of the size of raw text and represents it numerically for algorithmic comprehension. In our proposed approach, we prepare tokens using TensorFlow’s Tokenizer tool.

Subsequently, to ensure consistent input sizes, padding is applied. We define a maximum number of words for each sentence, and then add additional zeros to shorter sentences while truncating sentences that exceed the maximum word count.

3.1.5 Data Splitting

Data splitting is a crucial step in preparing datasets for machine learning, ensuring that the model is trained, validated, and tested on distinct subsets of the data to evaluate its performance effectively and avoid overfitting. In the provided code, the data splitting process involves two key stages to achieve this.

The date is split into a training set and a temporary set, with 20% of the data allocated for the temporary set. This temporary set is further divided into validation and testing sets, each containing 50% of the remaining data. Specifically:

1. Training Set: Comprising 15,004 samples with a feature shape of (15004, 100) and labels of shape (15004,).
2. Validation Set: Consisting of 1,876 samples with a feature shape of (1876, 100) and labels of shape (1876,).
3. Testing Set: Comprising 1,876 samples with a feature shape of (1876, 100) and labels of shape (1876,).

This method, with a random state set to 42 for reproducibility, ensures that the model is trained on one subset and validated and tested on separate, non-overlapping subsets. This approach is essential for assessing the model's generalization ability and performance on unseen data, which is critical for developing a reliable machine learning model.

3.1.6 Model Selection and Training

Model Selection

Metrics	**RNN**	**LSTM**	**GRU**	**Bi-LSTM**	**Bi-GRU**	**RNN-LSTM**	**RNN-GRU**
Loss	**0.5256**	**0.5767**	**0.4987**	**0.7953**	**0.7702**	**0.4460**	**0.4350**
Accuracy	**0.7559**	**0.7414**	**0.7692**	**0.7480**	**0.7581**	**0.7737**	**0.7871**

Table 3.2 Previous Models Trained and Its Performance

The project aims to classify sexism in text messages using deep learning techniques, focusing on evaluating various neural network models. The models are assessed based on their basic configurations to ensure uniformity across different architectures. This includes using consistent hyperparameters such as batch size, epochs, activation functions, and the number of layers. The exploration includes natural language processing techniques like embeddings and recurrent neural networks (RNNs), with a particular focus on enhancing selected RNN models based on their performance in previous projects. The goal is to determine how different model architectures—ranging from standard RNNs to more complex variants like Bidirectional LSTM (Bi-LSTM) and GRU—impact processing time and accuracy. Model selection and training are critical stages in the deep learning pipeline, and the research involves training, fine-tuning, and testing various models using initial and combined datasets. These include basic RNNs, LSTMs, Bi-LSTMs, GRUs, and their bidirectional and hybrid forms, such as RNN-LSTM and RNN-GRU. The models are compiled with binary cross-entropy loss and evaluated using accuracy, precision, and recall metrics.

The models will be explained in the further subchapter.

Fine Tuning

Keras Tuner is a powerful tool used for hyperparameter tuning in deep learning models, optimizing their performance by systematically searching for the best combination of hyperparameters. Fine-tuning with Keras Tuner involves experimenting with various hyperparameter configurations to identify the set that yields the highest validation accuracy.

The process begins with defining a model-building function, such as build_model, that allows for flexible hyperparameter adjustments. The tune_model function instantiates Keras Tuner's RandomSearch, which explores different hyperparameter combinations to find the optimal setup. In this case, max_trials=2 indicates that the tuner will try up to two different configurations, and executions_per_trial=1 specifies that each configuration will be tested once. The tuner evaluates each configuration's performance on the validation set and selects the best-performing model based on the validation accuracy (val_accuracy).

The advantages of using Keras Tuner for fine-tuning include its efficiency in searching a wide range of hyperparameters and its ability to automate the tuning process, saving time and computational resources compared to manual tuning. By leveraging Keras Tuner, the best hyperparameter values can be identified systematically, ensuring that the model performs optimally for the given dataset. This approach allows for comprehensive experimentation with different model types, such as RNN, LSTM, Bi-LSTM, GRU, Bi-GRU, RNN-LSTM, and RNN-GRU, providing a robust foundation for selecting the most effective model architecture. The tune_model function, applied to each model type, helps in comparing their performance and determining which configuration offers the best balance between accuracy and efficiency.

3.1.7 Model Evaluation

After training is complete, the model's performance is evaluated using several key metrics to ensure its effectiveness in classifying text messages for sexism detection. The evaluation metrics include the Confusion Matrix, Precision, Recall, F1-Score, and ROC AUC.

Confusion Matrix: This matrix provides a detailed breakdown of the model's predictions compared to the actual labels. It helps identify how many true positives, true negatives, false positives, and false negatives the model has produced. This is crucial for understanding the types of errors the model makes and for further refinement.

Precision: Precision measures the proportion of true positive predictions among all positive predictions made by the model. It is important in this project as it indicates the accuracy of the model when it predicts a text message as sexist, which helps in minimizing false positives and ensuring that flagged messages are genuinely sexist.

$$Precision = \frac{TP}{TP + FP}$$

Figure 3.13 Formula of Precision

Recall: Recall assesses the proportion of true positives among all actual positive cases. It is vital for this project to ensure that the model identifies as many sexist messages as possible, reducing false negatives and ensuring thorough detection.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Figure 3.14 Formula of Recall

F1-Score: The F1-Score is the harmonic mean of Precision and Recall, providing a single metric that balances the two. This metric is especially useful when there is an imbalance between precision and recall, offering a balanced view of the model's performance in classifying sexist messages.

$$F1 = \frac{TP}{TP + \frac{1}{2}(FP + FN)}$$

Figure 3.15 Formula of F1 score

ROC AUC: The ROC AUC score measures the area under the Receiver Operating Characteristic curve, reflecting the model's ability to distinguish between the classes. A higher AUC value indicates better performance in separating sexist messages from non-sexist ones, making it crucial for evaluating the overall effectiveness of the model.

These metrics are crucial for assessing different aspects of model performance, helping in choosing the best model for the classification task and ensuring that it meets the project's requirements effectively.

CHAPTER 4: Model Architecture Design

In chapter 4, the work done and the results obtained based Chapter 3 will be documented. In summary, the system design and implementation about data preprocessing, word embedding and model training will be reported.

4.1 Model Architecture Diagram

4.1.1 RNN

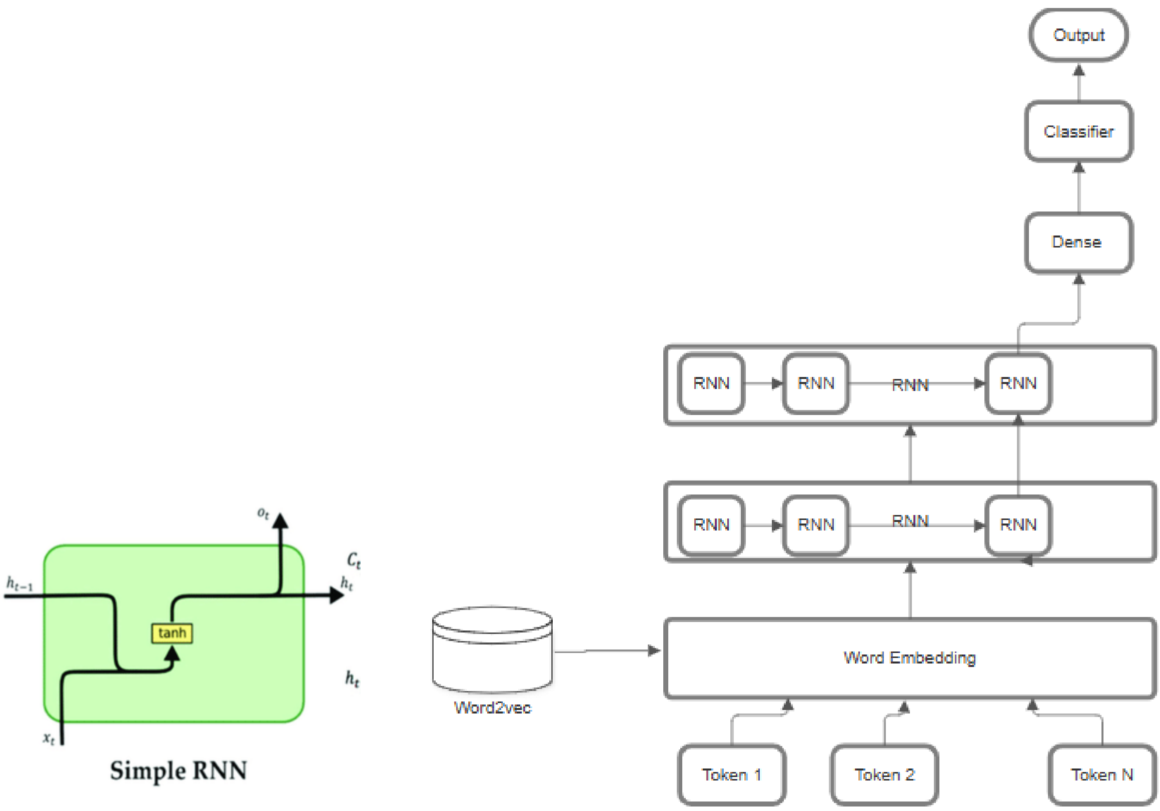


Figure 4.1 Architecture Diagram of RNN model

The structure of an RNN cell is straightforward, consisting of an input gate and a recurrent connection. Output from the prior timestep is fed becoming part of input for current timestep, enabling the cell to retain a form of memory across the sequence. During each timestep of sequence processing, the RNN cell receives two inputs: the current input ((x_t)) and the previous hidden state ((h_{t-1})). It combines these inputs to update its current (h_t), which then be used in generating an output or passed onto the next timestep. The hidden state (h_t) acts as the cell's memory, capturing information about what has been processed so far in the sequence. This state is updated at each timestep, allowing the RNN to carry forward information through the sequence.

The build_model function defines a Recurrent Neural Network (RNN) model using Keras, specifically designed for text classification tasks. It starts with an embedding layer that transforms input text sequences into dense vectors of fixed size, using pre-trained word embeddings that are not updated during training. This layer helps convert the text into a format suitable for the model. Following the embedding layer, two stacked SimpleRNN layers are added if the model type is set to 'RNN'. The number of units (neurons) in these RNN layers, as well as the dropout and recurrent dropout rates, are tunable hyperparameters, allowing for flexibility in model optimization. The first SimpleRNN layer returns the entire sequence of outputs (return_sequences=True), enabling the second layer to capture deeper temporal dependencies by processing the full sequence. The second SimpleRNN layer condenses the information into a single vector, making it suitable for classification. This architecture is designed to effectively model sequential patterns in text data while preventing overfitting through dropout regularization.

4.1.2 LSTM

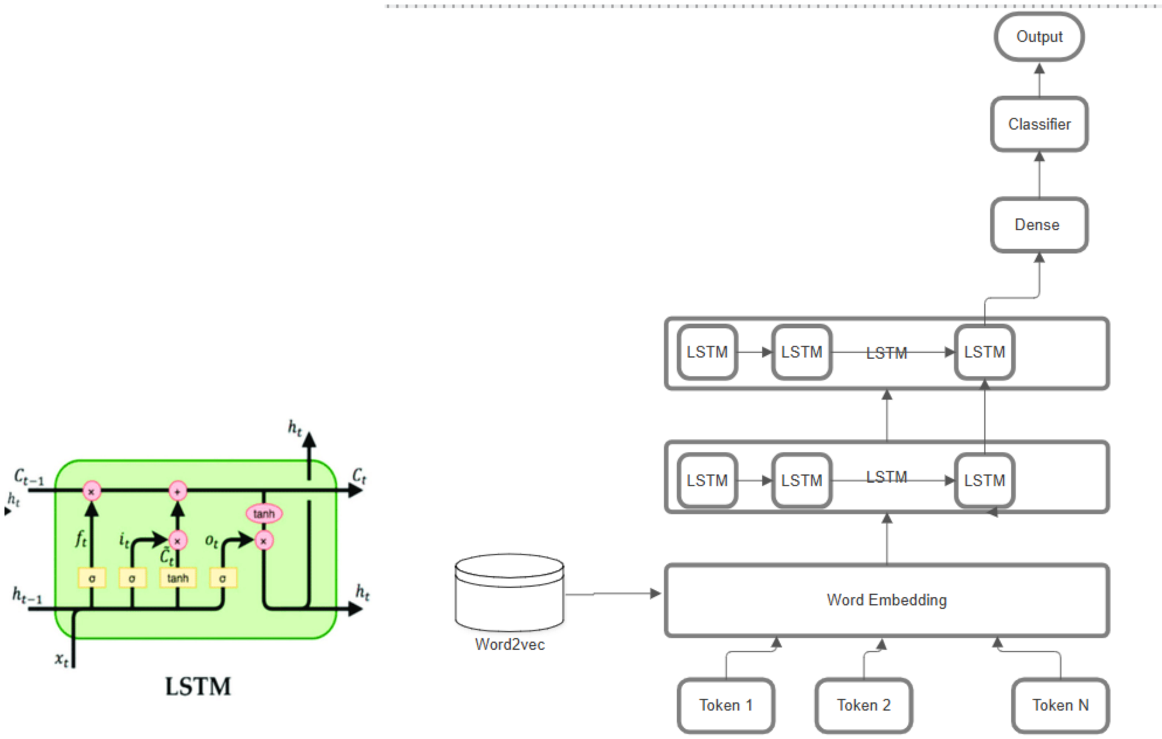


Figure 4.2 Architecture Diagram of LSTM model

An LSTM cell is composed of variety of components, including the cell state, input gate, output gate, and forget gate [30]. These elements collaborate to manage the flow of information. Cell state acts as 'memory' of the LSTM, retaining relevant information throughout sequence processing. This enables LSTMs to handle long-term dependencies effectively. The purpose of forget gate is to determine which information is discarded from the cell state, while the input gate decides what new information is stored. The output gate controls which data from the cell state contributes to computing the output at each timestep [2]. LSTMs are particularly adept at learning long-range dependencies and are less sensitive to gap lengths compared to other types of recurrent neural networks.

In the LSTM model, two LSTM layers are employed. The LSTM (Long Short-Term Memory) units are designed to handle sequential data and capture long-term dependencies. The first LSTM layer is configured with a variable number of units, as specified by the hyperparameter hp.Int('units', min_value=32, max_value=256, step=32). This layer uses dropout and recurrent dropout rates set by hp.Float('dropout_rate', min_value=0.1, max_value=0.5, step=0.1) and hp.Float('recurrent_dropout_rate', min_value=0.1, max_value=0.5, step=0.1), respectively, to prevent overfitting. The return_sequences=True parameter ensures that the output from the first LSTM layer is a sequence, allowing the next LSTM layer to process it. The second LSTM layer is also parameterized by the same units and dropout rates but does not return sequences, providing a final representation of the input sequence for classification.

4.1.3 GRU

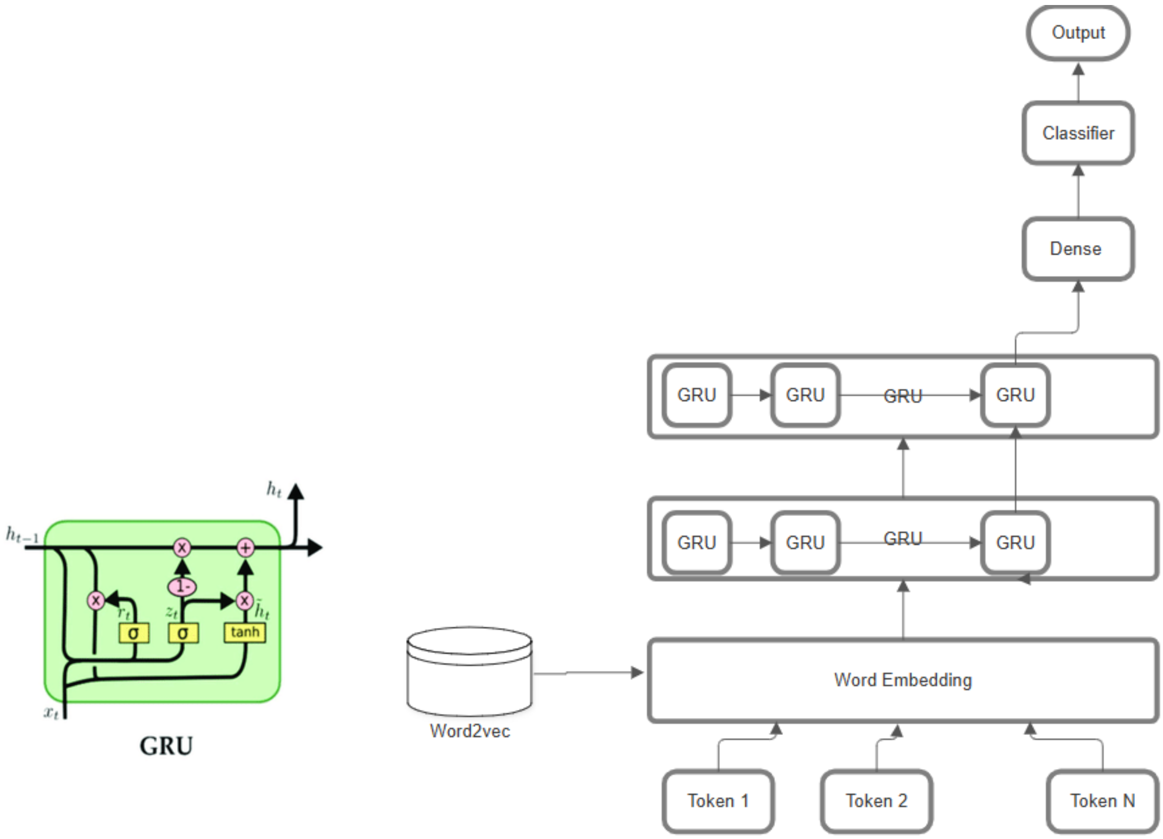


Figure 4.3 Architecture Diagram of GRU model

GRUs (Gated Recurrent Units) have a simpler structure compared to LSTMs, featuring only two gates: reset and update [2]. Update gate functions similarly to LSTM's input and forget gates, while the reset gate determines the amount of past information to forget. This allows GRUs to discard irrelevant details from previous timesteps.

The GRU model uses two stacked GRU (Gated Recurrent Unit) layers. GRUs are a simplified variant of LSTMs, designed to handle sequential data with fewer parameters, which can lead to faster training times. The first GRU layer is configured with a number of units specified by the hyperparameter hp.Int('units', min_value=32, max_value=256, step=32) and includes dropout and recurrent dropout rates as defined by hp.Float('dropout_rate', min_value=0.1, max_value=0.5, step=0.1) and hp.Float('recurrent_dropout_rate', min_value=0.1, max_value=0.5, step=0.1). This layer returns the entire sequence of outputs (return_sequences=True). The output is then processed by the second GRU layer, which also uses similar units and dropout settings but does not return sequences, providing a final condensed representation for classification.

4.1.4 Bi-LSTM

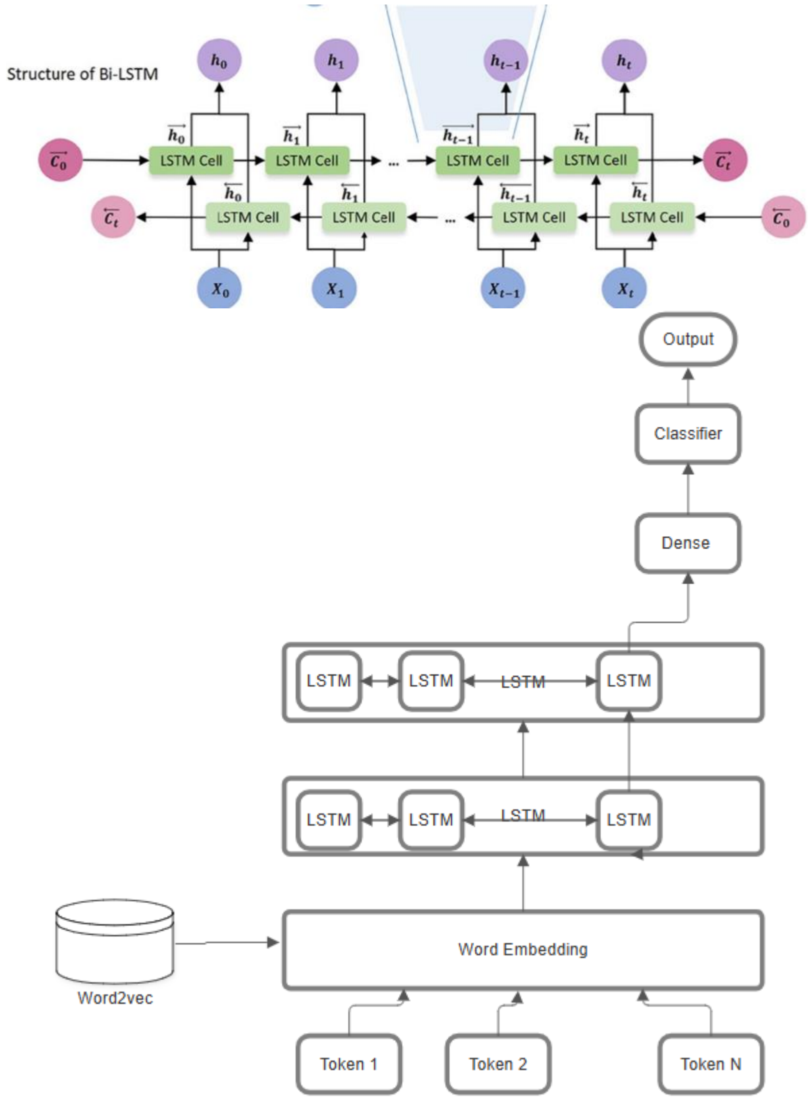


Figure 4.4 Architecture Diagram of Bi-LSTM model[21]

Bi-LSTM network structure advanced traditional LSTMs by handling data in both back and forth directions. These networks consist of two LSTM layers, allowing them to provide additional context to the model. To achieve this, the initial recurrent layer is duplicated. The input sequence is fed as-is to the first layer, while a reversed copy of the input is used in the second layer.

The Bi-LSTM model incorporates two Bidirectional LSTM layers. Bidirectional LSTMs process the input sequence in both forward and backward directions, which helps in capturing contextual information from both ends of the sequence. The first Bidirectional LSTM layer returns the full sequence of outputs (return_sequences=True), enabling the subsequent Bidirectional LSTM layer to further process this sequence. The second Bidirectional LSTM layer outputs a condensed representation of the sequence. This bidirectional processing can enhance the model's understanding of complex dependencies and patterns in the data.

4.1.5 Bi-GRU

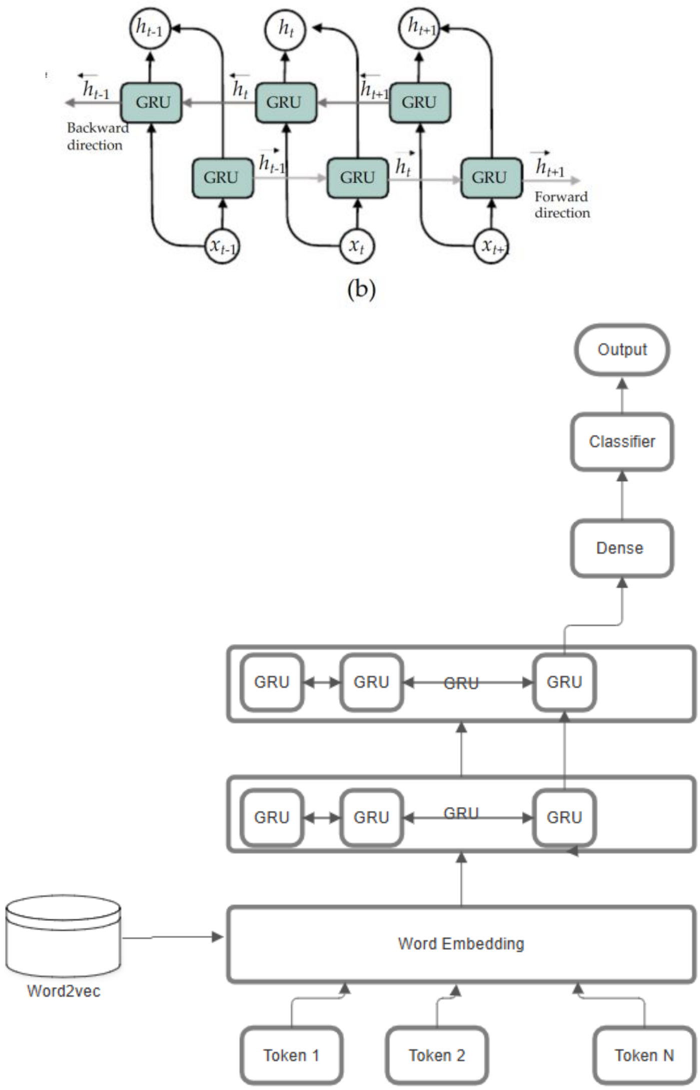


Figure 4.5 Architecture Diagram of Bi-GRU model

In a Bi-GRU, two GRU layers run in parallel. One part processes the input sequence in the onward direction, while the other processes it in the reverse direction [28]. Each GRU within the Bi-GRU has its own set of gates (reset and update gates) that manage the flow of information [28][29]. The outputs of the two GRUs are typically combined at each timestep to form the final result."

In the Bi-GRU model, two Bidirectional GRU layers are used. The Bidirectional GRUs analyze the sequence in both forward and backward directions, improving the model's ability to understand the context from both ends of the sequence. The first Bidirectional GRU layer returns the full sequence (return_sequences=True), and the second Bidirectional GRU layer processes this sequence to produce a final output. This bidirectional approach can be particularly useful in scenarios where understanding the sequence in both directions enhances the model's performance.

4.1.6 RNN-LSTM

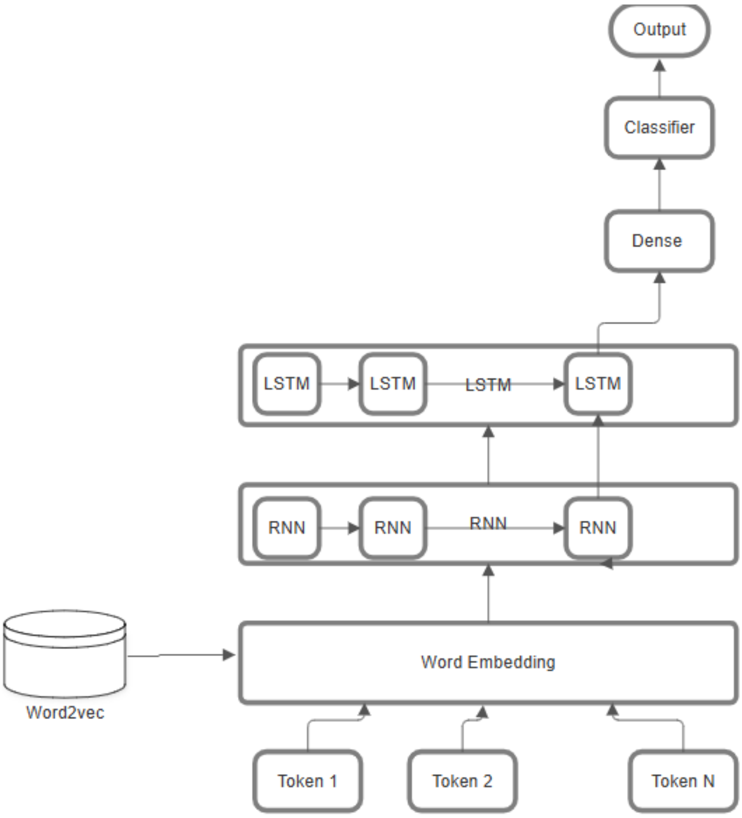


Figure 4.6 Architecture Diagram of RNN-LSTM model

The RNN-LSTM model integrates a Recurrent Neural Network (RNN) with a Long Short-Term Memory (LSTM) network to handle sequential data. Initially, the RNN layer processes the input sequence, capturing basic temporal dependencies through a series of hidden states. This RNN layer is configured with a variable number of units, specified by hp.Int('units', min_value=32, max_value=256, step=32), and incorporates dropout and recurrent dropout mechanisms to prevent overfitting, with rates defined by hp.Float('dropout_rate', min_value=0.1, max_value=0.5, step=0.1) and hp.Float('recurrent_dropout_rate', min_value=0.1, max_value=0.5, step=0.1), respectively. The return_sequences=True setting ensures that the RNN outputs the full sequence of hidden states. This output is then passed to the LSTM layer, which further processes the sequence using its advanced gating mechanisms to manage long-term dependencies. The LSTM layer also has a configurable number of units and uses dropout and recurrent dropout to mitigate overfitting. In the first LSTM layer, return_sequences=True maintains the sequence output for further processing, while the second LSTM layer provides a final summary of the sequence.

4.1.7 RNN-GRU

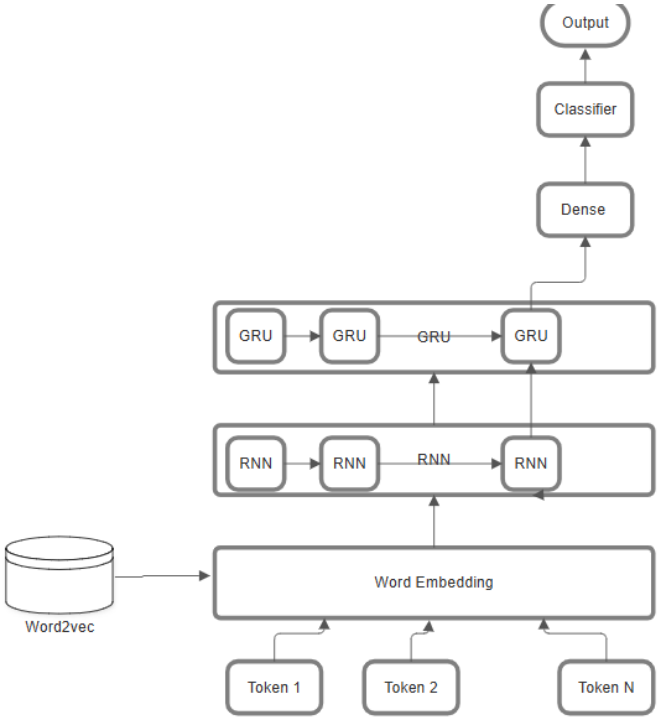


Figure 4.7 Architecture Diagram of RNN-GRU model

In contrast, the RNN-GRU model combines an RNN with a Gated Recurrent Unit (GRU) network. Similar to the RNN-LSTM model, the RNN layer in the RNN-GRU model processes the input sequence to capture basic temporal features, with units and dropout rates set similarly. This RNN layer outputs the full sequence of hidden states due to return_sequences=True. The GRU layer that follows simplifies the processing by integrating the cell state and hidden state into a single state, using gating mechanisms that offer computational efficiency while capturing complex temporal patterns. The GRU layer also allows for the configuration of units and dropout rates, with the first GRU layer returning the full sequence of outputs and the second GRU layer summarizing the sequence. Both models leverage the strengths of RNNs and advanced recurrent architectures to enhance their ability to classify sequential data effectively.

CHAPTER 5: Experiment/Simulation

5.1 Hardware Setup

The hardware utilized in this project consists of computer devices specifically designated for processing machine learning models.

Description	Specifications
Model	Huawei MateBook D 15
Processor	11^{th} Gen Intel® Core™ i5-1135G7 @ 2.40GHz 2.42GHz
Operating System	Windows 11 Home Single Language – 64 bit
Graphic	Intel ® Iris ® Xe Graphics
Memory	8.00 GB RAM
Storage	120 GB

Table 5.1 Specifications of laptop

5.2 Software Setup

Development	Software Tools
Programming Language	Python
Software	Jupyter Notebook, Anaconda

Table 5.2 Specifications of software

1) Jupyter Notebook

Anaconda so called as Anaconda Navigator is a program that allows user to launch application such as Jupyter Notebook easily without using line commands. Jupyter Notebook is a web application that allow user to create and share documents, scientific computing, machine learning and education. In this project, it will be used to write code in Python language.

2) Jieba

Jieba is a popular Chinese text segmentation library. It efficiently splits Chinese text into individual words or phrases, which is critical for natural language processing (NLP) tasks, such as text classification, word embedding, and machine translation. Jieba uses a combination of dictionary-based word segmentation and machine learning algorithms to find the best segmentation points in Chinese sentences, which lack spaces between words. It's used for tokenizing Chinese text into words, which are then fed into models like Word2Vec or neural networks for processing.

3) TensorFlow

TensorFlow is an open-source deep learning library developed by Google. It is widely used for building machine learning models, particularly neural networks. Within TensorFlow, Keras is a high-level API that simplifies building and training neural networks. It provides layers like LSTM, GRU, SimpleRNN, Dense, and others to design architectures for sequence processing. TensorFlow allows the creation of computational graphs, making it possible to train deep neural networks using backpropagation and gradient descent. It also handles automatic differentiation and optimization.

4) SMOTE

SMOTE is a method to address the problem of class imbalance in datasets. It generates synthetic samples of the minority class to balance the dataset, preventing the model from being biased toward the majority class. SMOTE works by selecting examples from the minority class and creating new synthetic instances by interpolating between the selected samples and their nearest neighbors.

5) Emojiswitch

Emojiswitch provides utility functions to convert emojis into their textual descriptions (demojize) and vice versa (emojize). This is useful in NLP tasks involving text containing emojis.

6) Deep Translator (GoogleTranslator)

Deep Translator provides a simple API for translating text between languages using services like Google Translate, Microsoft Translator, and others. Using the GoogleTranslator function, the library sends a request to the translation service, translating text from one language (e.g., English) to another (e.g., Chinese).

6) Genism

Gensim is a robust library for unsupervised topic modeling and natural language processing. It is commonly used for creating word embeddings, such as Word2Vec, which map words to continuous vector space representations. Gensim uses algorithms like Word2Vec, which are trained on large corpora of text to capture semantic relationships between words. Words with similar meanings are mapped to vectors that are close together in the vector space.

5.3 Experiment Setting

This project aims to evaluate and compare the accuracy of different neural networks on detecting sexism cyberbullying in SWSR and combined dataset. Thus, it is important to maintain the consistency of models for fairness. Following are the parameters to retain consistency:

Parameters	**Description**	**Value**
input_dim	Vocabulary size for the embedding layer	max_words (variable depending on the dataset)
output_dim	Embedding dimension	embedding_dim (variable)
input_length	Length of input sequences	maxlen (variable)
weights	Pre-trained embedding matrix	embedding_matrix (pre-trained)
trainable	Whether the embedding layer is trainable	False
units	Number of units in RNN/LSTM/GRU layers	32 to 256 (tuned using Keras Tuner)
dropout_rate	Dropout rate for regularization	0.1 to 0.5 (tuned using Keras Tuner)
recurrent_dropout_rate	Dropout rate for recurrent connections (RNN/LSTM/GRU)	0.1 to 0.5 (tuned using Keras Tuner)
return_sequences	Whether to return sequences (used in RNN, LSTM, GRU)	True
optimizer	Optimization algorithm	adam or rmsprop (tuned

		using Keras Tuner)
loss	Loss function for binary classification	binary_crossentropy
metrics	Performance metrics	accuracy, Precision, Recall,
batch_size	Batch size for model training	Variable (depends on experiment setup)
epochs	Number of training epochs	10
validation_data	Data used for model validation during training	(X_valid, y_valid)
max_trials	Number of hyperparameter tuning trials (Keras Tuner)	2
executions_per_trial	Number of executions per trial for Keras Tuner	1
directory	Directory to store the Keras Tuner results	'kt_dir_<model_type>'
project_name	Name of the Keras Tuner project	'text_classification'
model_type	Type of neural network model (RNN, LSTM, GRU, etc.)	RNN, LSTM, Bi-LSTM, GRU, Bi-GRU, etc.

Table 5.3 Experiment Parameters

5.4 Hyper tuning

This table summarized the best hyperparameters found for each model. Units refer to the number of neurons in the RNN/LSTM/GRU layers. Dropout rate and recurrent dropout rate are used for regularization, helping to prevent overfitting. Optimizer refers to the optimization algorithm used for model training.

Model	Units	Dropout Rate	Recurrent Dropout Rate	Optimizer
RNN	64	0.3	0.1	Rmsprop
LSTM	224	0.2	0.4	Adam

Bi-LSTM	96	-	-	Adam
GRU	64	0.3	0.5	Adam
Bi-GRU	64	-	-	Adam
RNN-LSTM	128	0.3	0.1	Adam
RNN-GRU	96	0.1	0.2	Adam

Table 5.4 Experiment Hyperparameters

5.5 Implementation Challenges

Implementing advanced language models for detecting offensive content and analyzing sentiment presents several challenges. One major issue is handling the vast variability and subtlety of human language, which can include implicit biases, sarcasm, and nuanced expressions. For instance, detecting sexism or offensive language in Chinese requires not only accurate dataset annotations but also the ability to interpret context effectively. To address these challenges, researchers have developed specialized datasets like SWSR and COLD, which provide targeted training data for specific types of offensive content.

Another challenge is the integration of additional features such as emojis to enhance sentiment analysis. Emojis add valuable emotional context but can also introduce complexity in their interpretation. Solutions include employing models like CEmo-LSTM that effectively incorporate emoji embeddings, improving classification accuracy. Additionally, ongoing refinement of models through manual annotation and addressing limitations such as the misclassification of anti-bias content are critical for improving performance. By leveraging advanced techniques and continuously updating resources, these challenges can be mitigated, leading to more accurate and robust language analysis systems.

CHAPTER 6: System Evaluation and Discussion

This chapter will discuss the performance of the chosen models and present testing results using a custom dataset to demonstrate their functionality. Various evaluation methods will be employed to gain deeper insights into the models' performance. Additionally, the chapter will address the project's limitations and propose future approaches to enhance the research, aiming to contribute to a more comprehensive understanding in this field.

6.1 System Testing

In the experiment, the SWSR dataset alone is referred to as the imbalanced dataset, while the combination of the SWSR and COLD datasets is referred to as the balanced dataset. Another distinction among the three result tables below is that one balanced dataset underwent English and Emoji translation, while the other did not, allowing for comparison.

Model	**Loss**	**Accuracy**	**Precision**	**Recall**	**F1-score**	**ROC AUC**	**Confusion Matrix (TN FP, FN TP)**
RNN	0.8723	0.5719	0.3988	0.4241	0.4110	0.5273	[[379, 202], [182, 134]]
LSTM	0.8507	0.5440	0.3982	0.5759	0.4709	0.5664	[[306, 275], [134, 182]]
Bi-LSTM	0.9515	0.5563	0.4060	0.5601	0.4707	0.5678	[[322, 259], [139, 177]]

GRU	0.8510	0.5619	0.4072	0.5348	0.4624	0.5753	[[335, 246], [147, 169]]
Bi-GRU	0.8672	0.5574	0.4043	0.5411	0.4628	0.5656	[[329, 252], [145, 171]]
RNN-LSTM	0.8487	0.5619	0.4077	0.5380	0.4638	0.5790	[[334, 247], [146, 170]]
RNN-GRU	0.8488	0.5563	0.3975	0.5032	0.4441	0.5772	[[340, 241], [157, 159]]

Table 6.1 Results for Imbalanced dataset and without English And Emoji Translation

Model	**Loss**	**Accuracy**	**Precision**	**Recall**	**F1-score**	**ROC AUC**	**Confusion Matrix (TN FP, FN TP)**
RNN	0.5225	0.7660	0.7447	0.6739	0.7075	0.8229	[[906, 182], [257, 531]]
LSTM	0.5306	0.7559	0.6937	0.7500	0.7207	0.8138	[[827, 261], [197, 591]]

Bi-LSTM	0.5702	0.7420	0.6818	0.7234	0.7020	0.8030	[[822, 266], [218, 570]]
GRU	0.5486	0.7500	0.7027	0.7018	0.7022	0.8080	[[854, 234], [235, 553]]
Bi-GRU	0.5214	0.7596	0.7073	0.7297	0.7183	0.8216	[[850, 238], [213, 575]]
RNN-LSTM	0.5187	0.7681	0.7231	0.7259	0.7245	0.8237	[[869, 219], [216, 572]]
RNN-GRU	0.5261	0.7649	0.7354	0.6878	0.7108	0.8217	[[893, 195], [246, 542]]

Table 6.2 Results for Balanced dataset and without English And Emoji Translation

Model	**Loss**	**Accuracy**	**Precision**	**Recall**	**F1-score**	**ROC AUC**	**Confusion Matrix (TN FP, FN TP)**
RNN	0.4814	0.7889	0.7782	0.7019	0.7381	0.8515	[[922, 159], [237, 558]]

LSTM	0.4616	0.7873	0.7374	0.7736	0.7551	0.8603	[[862, 219], [180, 615]]
Bi-LSTM	0.4774	0.7937	0.7361	0.8000	0.7667	0.8573	[[853, 228], [159, 636]]
GRU	0.4760	0.7857	0.7575	0.7270	0.7420	0.8532	[[896, 185], [217, 578]]
Bi-GRU	0.4112	0.8198	0.7824	0.7962	0.7893	0.8903	[[905, 176], [162, 633]]
RNN-LSTM	0.4285	0.8161	0.7863	0.7774	0.7818	0.8810	[[913, 168], [177, 618]]
RNN-GRU	0.4102	0.8188	0.8021	0.7597	0.7804	0.8923	[[932, 149], [191, 604]]

Table 6.3 Results for Balanced dataset and with English And Emoji Translation

6.1.1 Summary and Discussion of the Model Performance

The performance metrics of the deep learning models used for detecting Chinese sexism reveal nuanced insights into each model's strengths and weaknesses. Here is a detailed analysis based on the key performance metrics: accuracy, precision, recall, F1-score, and ROC AUC.

Accuracy measures the overall correctness of the model in classifying instances. Among the models, Bi-GRU achieved the highest accuracy at 81.98%, closely followed by RNN_GRU with 81.88%. These results indicate that Bi-GRU and RNN_GRU models are highly reliable in terms of general classification performance. On the other hand, RNN achieved the lowest accuracy at 78.89%, suggesting that its simpler architecture may not capture the complexities of the task as effectively as more advanced models.

Precision quantifies the accuracy of positive predictions. The RNN_GRU model excelled with a precision of 80.21%, meaning it was very effective at correctly identifying positive instances of Chinese sexism. Similarly, Bi-GRU also performed well with a precision of 78.24%. The lower precision in the LSTM model, at 73.74%, indicates that it had more false positives compared to the best-performing models. High precision is crucial in applications where the cost of false positives is significant.

Recall measures the model's ability to identify all relevant positive instances. The Bi-LSTM model achieved the highest recall of 80.00%, suggesting it was most effective at identifying instances of sexism. RNN_GRU also showed strong recall at 75.97%, reflecting its ability to capture a substantial number of true positives. The RNN model had the lowest recall at 70.19%, which implies it missed a higher proportion of actual sexism instances compared to other models. High recall is essential for ensuring that most instances of sexism are identified, even if it means accepting some false positives.

F1-Score combines precision and recall into a single metric, providing a balanced measure of performance. The Bi-GRU model achieved the highest F1-score of 78.93%, reflecting a good balance between precision and recall. The RNN_LSTM model also performed well with an F1-score of 78.18%. Models like LSTM and GRU had lower F1-scores, indicating a trade-off between precision and recall that affects their overall performance balance.

ROC AUC measures the model's ability to distinguish between classes across various thresholds. The Bi-GRU model achieved the highest ROC AUC of 0.8903, indicating excellent performance in differentiating between positive and negative instances. Similarly, RNN_GRU scored 0.8923, showcasing its robust discriminatory power. The RNN model, with the lowest ROC AUC of 0.8515, displayed less effective class separation, underscoring its limitations in distinguishing between classes compared to more complex models.

Overall, models with more advanced architectures, such as Bi-GRU and RNN_GRU, demonstrated superior performance across most metrics. Their ability to achieve higher accuracy, precision, recall, F1-score, and ROC AUC underscores their effectiveness in detecting Chinese sexism. Conversely, simpler models like RNN lagged behind, highlighting the benefits of using more sophisticated deep learning techniques for this task.

6.2 Objectives Evaluation

The first objective which is to investigate techniques to translate Emoji and English into Chinese text was successfully achieved. Various techniques were explored and implemented to handle emoji and English translation into Chinese. Specifically, we utilized the deep_translator library, which includes GoogleTranslator, to translate English words into Chinese. For emojis, the emojiswitch library was used to convert emojis into their descriptive text equivalents in Chinese. The translations were then integrated into the text processing pipeline, ensuring that both emoji and English words were properly converted into Chinese.

The second objective is to develop various deep learning models to detect Chinese sexism. This objective was also achieved. Multiple deep learning models, including RNN, LSTM, Bi-LSTM, GRU, Bi-GRU, RNN_LSTM, and RNN_GRU, were developed and trained to detect Chinese sexism. Each model was tuned with different hyperparameters, including the number of units, dropout rates, and optimizers, in order to enhance performance. The models were designed specifically to classify Chinese text data for sexism detection, providing a diverse set of architectures to analyze.

The third objective is to evaluate the performance of various models in detecting Chinese sexism. This objective was thoroughly fulfilled by evaluating the performance of each deep learning model using metrics such as accuracy, precision, recall, F1-score, and ROC AUC. The results showed that Bi-GRU and RNN_GRU were the top-performing models, achieving high accuracy, precision, and F1-scores. The RNN model, however, performed worse than the other models, illustrating the importance of more sophisticated architectures for this task. Through careful analysis of confusion matrices and performance scores, the strengths and weaknesses of each model were identified and discussed.

All the objectives were met. The techniques for translating emoji and English into Chinese were successfully implemented, multiple deep learning models were developed, and their performance in detecting Chinese sexism was thoroughly evaluated. Through these processes, valuable insights were gained into the effectiveness of different deep learning architectures for this task.

6.3 Future Remark

Looking ahead, the evaluation and discussion of the system's performance provide a foundation for future improvements and developments. The successful application of deep learning models like Bi-GRU and RNN_GRU has demonstrated their potential in detecting Chinese sexism with high accuracy and reliability. However, several opportunities for further refinement and advancement remain.

While the models have shown strong performance, there is always room for enhancement. Future testing could involve exploring additional deep learning architectures or hybrid models to further improve detection capabilities. Implementing more sophisticated techniques, such as attention mechanisms or transformer-based models, might offer even better performance by capturing more nuanced patterns in the data.

The promising results from models like Bi-GRU and RNN_GRU highlight the importance of leveraging advanced architectures for text classification tasks. Future work should focus on optimizing these models and experimenting with other state-of-the-art approaches. Additionally, it would be beneficial to incorporate domain-specific knowledge and contextual understanding to refine the models' accuracy further.

The project's objectives were achieved through effective translation techniques and the development of robust deep learning models. Going forward, refining the translation process, especially for emojis and nuanced text, could enhance the overall system's performance. Expanding the dataset to include a more diverse range of examples might also improve model generalizability and robustness.

In summary, while the current system evaluation has confirmed the effectiveness of the implemented models, future efforts should focus on continuous improvement and adaptation. By exploring new techniques, optimizing existing models, and addressing any limitations, the system can be further refined to meet evolving needs and provide even more accurate and reliable results in detecting Chinese sexism.

CHAPTER 7: Conclusion

In conclusion, this project aimed to explore and compare various deep learning models for cyberbullying detection, particularly focusing on Chinese language datasets. By utilizing the SWSR and COLD datasets, we were able to assess the performance of models such as RNN, LSTM, GRU, Bi-LSTM, and Bi-GRU across imbalanced and balanced data conditions. The inclusion of emoji and English translations further enriched the analysis, providing insights into how such features can impact model accuracy and performance.

The results demonstrated that combining datasets to address imbalances, as well as incorporating auxiliary features like emojis and translations, can significantly improve detection performance, particularly in terms of precision, recall, and F1-score. While some models, such as the RNN-LSTM and Bi-GRU, consistently outperformed others, the analysis also highlighted the importance of dataset composition and feature selection in enhancing model robustness. This project not only contributes to the field of cyberbullying detection but also emphasizes the need for continuous dataset refinement and model optimization to tackle evolving linguistic challenges.

REFERENCES

[1] A. Jiang, X. Yang, Y. Liu, and A. Zubiaga, “SWSR: A Chinese dataset and lexicon for online sexism detection,” *Online Social Networks and Media*, vol. 27, p. 100182, Jan. 2022, doi: 10.1016/j.osnem.2021.100182.

[2] M. Kaur and A. Mohta, “A Review of Deep Learning with Recurrent Neural Network,” *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Nov. 2019, doi: 10.1109/icssit46314.2019.8987837.

[3] S. Chen, J. Wang, and K. He, “Chinese cyberbullying detection using XLNET and Deep Bi-LSTM hybrid model,” *Information*, vol. 15, no. 2, p. 93, Feb. 2024, doi: 10.3390/info15020093.

[4] T. Fu and H. Liu, “Research on Chinese Text Classification Based on Improved RNN,” *IEEE 3rd International Conference on Electronic Technology, Communication and Information (ICETCI), Changchun, China,* May 2023, doi: 10.1109/icetci57876.2023.10176780.

[5] M. P. Akhter, J. Zheng, I. R. Naqvi, M. Abdelmajeed, and T. Zia, “Abusive language detection from social media comments using conventional machine learning and deep learning approaches,” *Multimedia Systems*, vol. 28, no. 6, pp. 1925–1940, Apr. 2021, doi: 10.1007/s00530-021-00784-8.

[6] J. Yadav, D. Kumar, and D. Chauhan, “Cyberbullying Detection using Pre-Trained BERT Model,” *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, Jul. 2020, doi: 10.1109/icesc48915.2020.9155700.

[7] N. Haydar and B. N. Dhannoon, “A comparative study of cyberbullying detection in social media for the last five years,” *˜ Al-œ Nahrain Journal of Science*, vol. 26, no. 2, pp. 47–55, Jun. 2023, doi: 10.22401/anjs.26.2.08.

[8] Md. T. Hasan, Md. A. E. Hossain, Md. S. H. Mukta, A. Akter, M. Ahmed, and S. Islam, “A review on Deep-Learning-Based Cyberbullying Detection,” *Future Internet*, vol. 15, no. 5, p. 179, May 2023, doi: 10.3390/fi15050179.

[9] M. Raj, S. Singh, K. Solanki, and R. Selvanambi, “An application to detect cyberbullying using machine learning and deep learning techniques,” *SN Computer Science/SN Computer Science*, vol. 3, no. 5, Jul. 2022, doi: 10.1007/s42979-022-01308-5.

[10] J. Deng *et al.*, “COLD: A Benchmark for Chinese Offensive Language Detection,” *The CoAI Group, DCST, Institute for Artificial Intelligence, State Key Lab of Intelligent Technology and Systems, 1Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing*, Jan. 2022, doi: 10.18653/v1/2022.emnlp-main.796.

[11] A. Muneer, A. Alwadain, M. G. Ragab, and A. Alqushaibi, “Cyberbullying detection on social media using stacking ensemble learning and enhanced BERT,”

Information, vol. 14, no. 8, p. 467, Aug. 2023, doi: 10.3390/info14080467.

[12] Y. Cui, W. Che, T. Liu, B. Qin, S. Wang, and G. Hu, “Revisiting Pre-Trained Models for Chinese Natural Language Processing,” *Findings of the Association for Computational Linguistics: EMNLP*, Jan. 2020, doi: 10.18653/v1/2020.findings-emnlp.58.

[13] Y.-H. Wu, S. Huang, W.-Y. Chung, C.-C. Yu, and J.-L. Wu, “Factor detection task of cyberbullying using the deep learning model,” *2022 IEEE International Conference on Big Data (Big Data)*, Dec. 2022, doi: 10.1109/bigdata55660.2022.10020779.

[14] B. Ogunleye and B. Dharmaraj, “The use of a large language model for cyberbullying detection,” *Analytics (Basel)*, vol. 2, no. 3, pp. 694–707, Sep. 2023, doi: 10.3390/analytics2030038.

[15] A. Dewani, M. A. Memon, and S. Bhatti, “Cyberbullying detection: advanced preprocessing techniques & deep learning architecture for Roman Urdu data,” *Journal of Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00550-7.

[16] M. Alotaibi, B. Alotaibi, and A. Razaque, “A multichannel deep learning framework for cyberbullying detection on social media,” *Electronics*, vol. 10, no. 21, p. 2664, Oct. 2021, doi: 10.3390/electronics10212664.

[17] A. Muneer and S. M. Fati, “A comparative analysis of machine learning techniques for cyberbullying detection on Twitter,” *Future Internet*, vol. 12, no. 11, p. 187, Oct. 2020, doi: 10.3390/fi12110187.

[18] F. Elsafoury, S. Katsigiannis, Z. Pervez, and N. Ramzan, “When the timeline meets the pipeline: A survey on Automated Cyberbullying Detection,” *IEEE Access*, vol. 9, pp. 103541–103563, Jan. 2021, doi: 10.1109/access.2021.3098979.

[19] K. Maity, S. Bhattacharya, S. Saha, and M. Seera, “A deep learning framework for the detection of Malay hate speech,” *IEEE Access*, vol. 11, pp. 79542–79552, Jan. 2023, doi: 10.1109/access.2023.3298808.

[20] H. Rosa *et al.*, “Automatic cyberbullying detection: A systematic review,” *Computers in Human Behavior*, vol. 93, pp. 333–345, Apr. 2019, doi: 10.1016/j.chb.2018.12.021.

[21] Y. Gao, “A systematic review and outlook on machine-learning-based methods for spam-filtering,” *Applied and Computational Engineering*, vol. 4, no. 1, pp. 702–709, Jun. 2023, doi: 10.54254/2755-2721/4/2023400.

[22] M. Song, X. Fu, S. Wang, Z. Du, and Y. Zhang, “Predicting Organization Performance Changes: a Sequential Data-Based framework,” *Frontiers in Psychology*, vol. 13, May 2022, doi: 10.3389/fpsyg.2022.899466.

[23] S. Mekruksavanich, N. Hnoohom, and A. Jitpattanakul, “A hybrid deep residual network for efficient transitional activity recognition based on wearable sensors,”

Applied Sciences, vol. 12, no. 10, p. 4988, May 2022, doi: 10.3390/app12104988.

[24] Y.-H. Wu, S. Huang, W.-Y. Chung, C.-C. Yu, and J.-L. Wu, "Factor detection task of cyberbullying using the deep learning model," *2022 IEEE International Conference on Big Data (Big Data)*, Dec. 2022, doi: 10.1109/bigdata55660.2022.10020779.

[25] B. Ogunleye and B. Dharmaraj, "The use of a large language model for cyberbullying detection," *Analytics (Basel)*, vol. 2, no. 3, pp. 694–707, Sep. 2023, doi: 10.3390/analytics2030038.

[26] A. Dewani, M. A. Memon, and S. Bhatti, "Cyberbullying detection: advanced preprocessing techniques & deep learning architecture for Roman Urdu data," *Journal of Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00550-7.

[27] M. Alotaibi, B. Alotaibi, and A. Razaque, "A multichannel deep learning framework for cyberbullying detection on social media," *Electronics*, vol. 10, no. 21, p. 2664, Oct. 2021, doi: 10.3390/electronics10212664.

[28] A. Muneer and S. M. Fati, "A comparative analysis of machine learning techniques for cyberbullying detection on Twitter," *Future Internet*, vol. 12, no. 11, p. 187, Oct. 2020, doi: 10.3390/fi12110187.

[29] F. Elsafoury, S. Katsigiannis, Z. Pervez, and N. Ramzan, "When the timeline meets the pipeline: A survey on Automated Cyberbullying Detection," *IEEE Access*, vol. 9, pp. 103541–103563, Jan. 2021, doi: 10.1109/access.2021.3098979.

[30] K. Maity, S. Bhattacharya, S. Saha, and M. Seera, "A deep learning framework for the detection of Malay hate speech," *IEEE Access*, vol. 11, pp. 79542–79552, Jan. 2023, doi: 10.1109/access.2023.3298808.

[31] H. Rosa *et al.*, "Automatic cyberbullying detection: A systematic review," *Computers in Human Behavior*, vol. 93, pp. 333–345, Apr. 2019, doi: 10.1016/j.chb.2018.12.021.

[32] Y. Gao, "A systematic review and outlook on machine-learning-based methods for spam-filtering," *Applied and Computational Engineering*, vol. 4, no. 1, pp. 702–709, Jun. 2023, doi: 10.54254/2755-2721/4/2023400.

[33] M. Song, X. Fu, S. Wang, Z. Du, and Y. Zhang, "Predicting Organization Performance Changes: a Sequential Data-Based framework," *Frontiers in Psychology*, vol. 13, May 2022, doi: 10.3389/fpsyg.2022.899466.

[34] S. Mekruksavanich, N. Hnoohom, and A. Jitpattanakul, "A hybrid deep residual network for efficient transitional activity recognition based on wearable sensors," *Applied Sciences*, vol. 12, no. 10, p. 4988, May 2022, doi: 10.3390/app12104988.

[35] J. Ma and L. Li, “Data augmentation for Chinese text classification using Back-Translation,” *Journal of Physics. Conference Series*, vol. 1651, no. 1, p. 012039, Nov. 2020, doi: 10.1088/1742-6596/1651/1/012039.

[36] C. Liu *et al.*, “Improving sentiment analysis accuracy with emoji embedding,” *Journal of Safety Science and Resilience*, vol. 2, no. 4, pp. 246–252, Dec. 2021, doi: 10.1016/j.jnlssr.2021.10.003.

[37] J. Li, D. Zhang, and A. Wulamu, “Chinese text classification based on ERNIE-RNN,” *2021 2nd International Conference on Electronics, Communications and Information Technology (CECIT)*, Dec. 2021, doi: 10.1109/cecit53797.2021.00072.

[26]J. Feng, “Cyberbullying in China finds victims in all corners,” *The China Project*, Mar. 29, 2023. [Online]. Available: https://thechinaproject.com/2023/03/29/cyberbullying-in-china-finds-victims-in-all-corners/

[27] N. A. Samee, U. Khan, S. Khan, M. Jamjoom, M. Sharif, and D. H. Kim, “Safeguarding Online Spaces: a powerful fusion of federated learning, word embeddings, and emotional features for cyberbullying detection,” *IEEE Access*, vol. 11, pp. 124524–124541, Jan. 2023, doi: 10.1109/access.2023.3329347.

[28] G. Kazbekova, Z. Ismagulova, Z. Kemelbekova, S. Tileubay, B. Baimurzayev, and A. Bazarbayeva, “Offensive Language Detection on Online Social Networks using Hybrid Deep Learning Architecture,” *International Journal of Advanced Computer Science and Applications/International Journal of Advanced Computer Science & Applications*, vol. 14, no. 11, Jan. 2023, doi: 10.14569/ijacsa.2023.0141180.

[29] A. M. E. Koshiry, E. H. I. Eliwa, T. A. El-Hafeez, and M. Khairy, “Detecting cyberbullying using deep learning techniques: a pre-trained glove and focal loss technique,” *PeerJ. Computer Science*, vol. 10, p. e1961, Mar. 2024, doi: 10.7717/peerj-cs.1961.

[30] L. Xiaoyan, R. C. Raga, and X. Shi, “GLOVE-CNN-BILSTM model for sentiment analysis on text reviews,” *Journal of Sensors*, vol. 2022, pp. 1–12, Oct. 2022, doi: 10.1155/2022/7212366.

[31] D. Sultan, M. Mendes, A. Kassenkhan, and O. Akylbekov, “Hybrid CNN-LSTM Network for Cyberbullying Detection on Social Networks using Textual Contents,” *International Journal of Advanced Computer Science and Applications/International Journal of Advanced Computer Science & Applications*, vol. 14, no. 9, Jan. 2023, doi: 10.14569/ijacsa.2023.0140978.

[32] X. Huang, O. Elshafiey, K. Farzia, Л. Удпа, M. Han, and Y. Deng, “Acoustic Emission Source Localization using Deep Transfer Learning and Finite Element Modeling–based Knowledge Transfer,” *Materials Evaluation*, vol. 81, no. 7, pp. 71–84, Jul. 2023, doi: 10.32548/2023.me-04348.

Ts. Dr. Vikneswary Jayapal

[33] H. Wang, Y. Wang, X. Song, B. Zhou, X. Zhao, and F. Xie, “Quantifying controversy from stance, sentiment, offensiveness and sarcasm: a fine-grained controversy intensity measurement framework on a Chinese dataset,” *World Wide Web,* vol. 26, no. 5, pp. 3607–3632, Aug. 2023, doi: 10.1007/s11280-023-01191-x.

[34] M. Xin, J. Shen, and P.-W. Hao, “Cyberbullying detection and classification with improved IG and BiLSTM,” *2022 International Conference on Electronics and Devices, Computational Science (ICEDCS)*, Sep. 2022, doi: 10.1109/icedcs57360.2022.00065.

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 2,3	**Study week no.: 3**
Student Name & ID: HAZEL LIM BENIN (22ACB00107)	
Supervisor: DR VIKNESWARY A/P JAYAPAL	
Project Title: HATE SPEECH DETECTION IN CHINESE LANGUAGE USING DEEP LEARNING	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Read new relevant journal articles and research papers for literature reviews
- Timeline done
- Project direction confirmed

2. WORK TO BE DONE

- Balancing data
- Embedding layers (emoji)
- Architecture diagrams
- Fine tune hyperparameters

3. PROBLEMS ENCOUNTERED

- Improve model performance

4. SELF EVALUATION OF THE PROGRESS

- Progress is slow, need to catch up both report and model.

vicky
Supervisor's signature

Hazel
Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 2,3	**Study week no.: 4**
Student Name & ID: HAZEL LIM BENIN (22ACB00107)	
Supervisor: DR VIKNESWARY A/P JAYAPAL	
Project Title: HATE SPEECH DETECTION IN CHINESE LANGUAGE USING DEEP LEARNING	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

- Read new relevant journal articles and research papers for literature reviews
- Timeline done
- Project direction confirmed

2. WORK TO BE DONE

- Balancing data
- Embedding layers (emoji)
- Architecture diagrams
- Fine tune hyperparameters

3. PROBLEMS ENCOUNTERED

- no

4. SELF EVALUATION OF THE PROGRESS

- Progress is slow, need to catch up both report and model.

vicky

Supervisor's signature

Hazel

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: 2,3	**Study week no.: 8**
Student Name & ID: HAZEL LIM BENIN (22ACB00107)	
Supervisor: DR VIKNESWARY A/P JAYAPAL	
Project Title: HATE SPEECH DETECTION IN CHINESE LANGUAGE USING DEEP LEARNING	

1. WORK DONE
[Please write the details of the work done in the last fortnight.]

- Balancing data
- Refine objectives
- Change project title, project type

2. WORK TO BE DONE

- Confirm methods for English and Emoji conversion
- Architecture diagrams
- Fine tune hyperparameters

3. PROBLEMS ENCOUNTERED

- Find suitable model for embeddings

4. SELF EVALUATION OF THE PROGRESS

- Need to finish the model and continue writing report.

vicky
Supervisor's signature

Hazel
Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

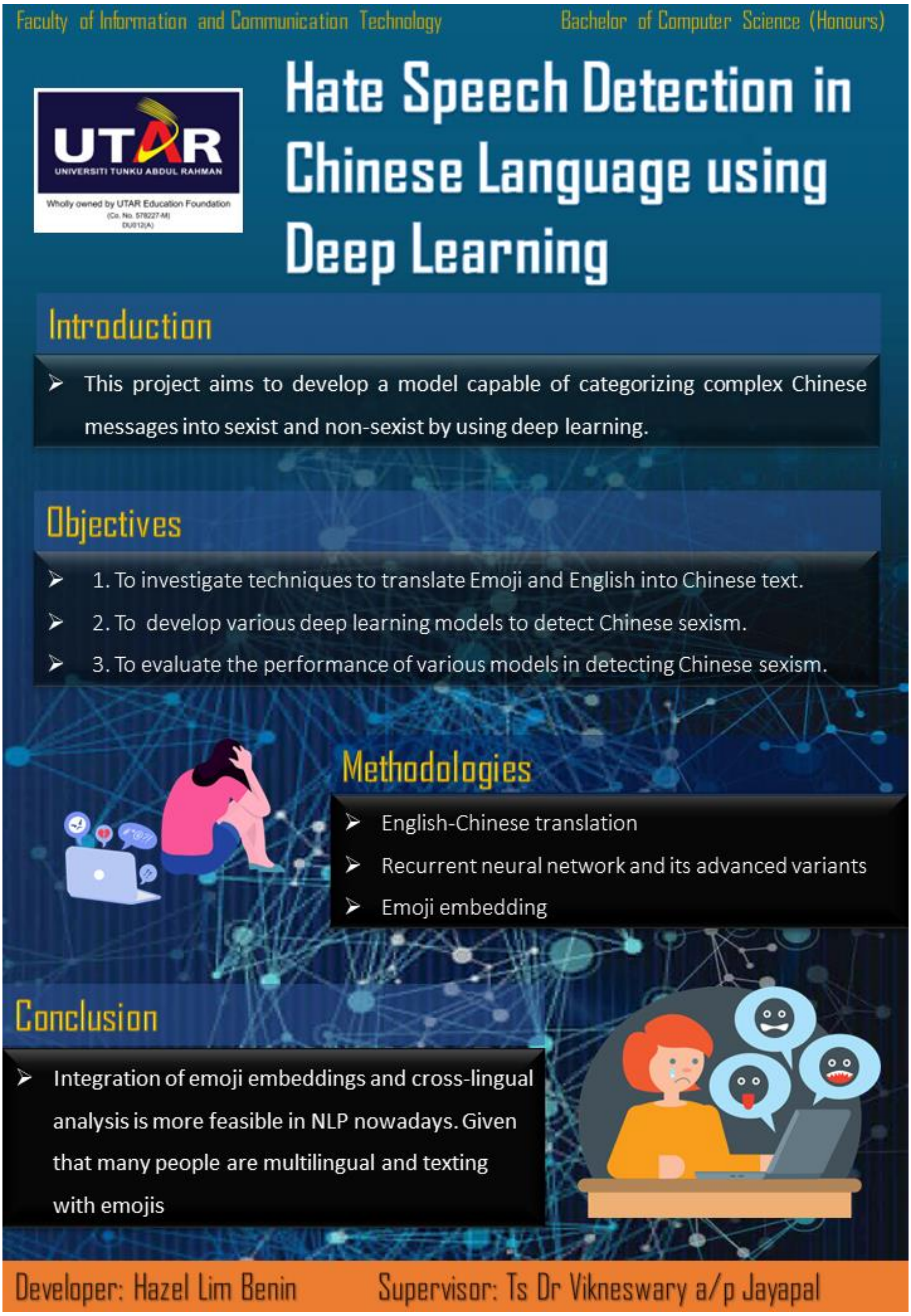
Trimester, Year: 2,3	**Study week no.: 10**
Student Name & ID: HAZEL LIM BENIN (22ACB00107)	
Supervisor: DR VIKNESWARY A/P JAYAPAL	
Project Title: HATE SPEECH DETECTION IN CHINESE LANGUAGE USING DEEP LEARNING	

1. WORK DONE [Please write the details of the work done in the last fortnight.] - Balancing data - Refine objectives - Change project title, project type
2. WORK TO BE DONE - Confirm methods for English and Emoji conversion - Architecture diagrams - Fine tune hyperparameters - Report writing
3. PROBLEMS ENCOUNTERED - Find suitable model for embeddings
4. SELF EVALUATION OF THE PROGRESS - Need to finish the model and continue writing report.

vicky
Supervisor's signature

Hazel
Student's signature

POSTER


Faculty of Information and Communication Technology
Bachelor of Computer Science (Honours)
Hate Speech Detection in Chinese Language using Deep Learning
Introduction
This project aims to develop a model capable of categorizing complex Chinese messages into sexist and non-sexist by using deep learning.
Objectives
1. To investigate techniques to translate Emoji and English into Chinese text.
2. To develop various deep learning models to detect Chinese sexism.
3. To evaluate the performance of various models in detecting Chinese sexism.
Methodologies
English-Chinese translation
Recurrent neural network and its advanced variants
Emoji embedding
Conclusion
Integration of emoji embeddings and cross-lingual analysis is more feasible in NLP nowadays. Given that many people are multilingual and texting with emojis
Developer: Hazel Lim Benin
Supervisor: Ts Dr Vikneswary a/p Jayapal

PLAGIARISM CHECK RESULT

22ACB00107_FYP2.docx

ORIGINALITY REPORT

14%	10%	7%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	fastercapital.com Internet Source	2%
2	pureadmin.unileoben.ac.at Internet Source	1%
3	Vibhav Sachan, Shahid Malik, Ruchita Gautam, Kumar Parvin. "Advances in AI for Biomedical Instrumentation, Electronics and Computing", CRC Press, 2024 Publication	1%
4	Hemant Kumar Soni, Sanjiv Sharma, G. R. Sinha. "Text and Social Media Analytics for Fake News and Hate Speech Detection", CRC Press, 2024 Publication	1%
5	www.mdpi.com Internet Source	1%
6	www.researchgate.net Internet Source	<1%
7	Mehdi Ghayoumi. "Generative Adversarial Networks in Practice", CRC Press, 2023	<1%

PLAGIARISM CHECK RESULT

	Publication	
8	ijritcc.org Internet Source	<1 %
9	Submitted to University College London Student Paper	<1 %
10	jdmdc.com Internet Source	<1 %
11	Submitted to srmengineeringcollege Student Paper	<1 %
12	Submitted to WorldQuant University Student Paper	<1 %
13	medium.com Internet Source	<1 %
14	Submitted to University of Technology, Sydney Student Paper	<1 %
15	Submitted to AlHussein Technical University Student Paper	<1 %
16	www.fastercapital.com Internet Source	<1 %
17	Submitted to Harrisburg University of Science and Technology Student Paper	<1 %
18	www.ijraset.com Internet Source	<1 %

19	Ramchandra S. Mangrulkar, Antonis Michalas, Narendra M. Shekokar, Meera Narvekar, Pallavi V. Chavan. "Design of Intelligent Applications Using Machine Learning and Deep Learning Techniques", CRC Press, 2021 Publication	<1 %
20	Submitted to University of Wales Institute, Cardiff Student Paper	<1 %
21	developer.aliyun.com Internet Source	<1 %
22	www.appinio.com Internet Source	<1 %
23	Submitted to Swiss School of Business and Management - SSBM Student Paper	<1 %
24	Submitted to UCL Student Paper	<1 %
25	assets-eu.researchsquare.com Internet Source	<1 %
26	jurnal.iaii.or.id Internet Source	<1 %
27	kylo.tv Internet Source	<1 %

28	Cheng Zhang, Nilam Nur Amir Sjarif, Roslina Ibrahim. "Deep learning models for price forecasting of financial time series: A review of recent advancements: 2020–2022", WIREs Data Mining and Knowledge Discovery, 2023 Publication	<1 %
29	Submitted to University of Hong Kong Student Paper	<1 %
30	Submitted to University of Limerick Student Paper	<1 %
31	etd.aau.edu.et Internet Source	<1 %
32	Submitted to University of Leeds Student Paper	<1 %
33	Submitted to Addis Ababa Science and Technology University Student Paper	<1 %
34	Submitted to University of Sunderland Student Paper	<1 %
35	ictactjournals.in Internet Source	<1 %
36	qiita.com Internet Source	<1 %
37	M. Kalpana Chowdary, J. Anitha, D. Jude Hemanth. "Emotion Recognition from EEG	<1 %

	Signals Using Recurrent Neural Networks", Electronics, 2022 Publication	
38	Ruey-Chyi Wu. "Development of an Intelligent Virtualization Platform Key Metrics Monitoring System: Collaborative Implementation with Self-Training and Bagging Algorithm", Research Square Platform LLC, 2024 Publication	<1 %
39	Submitted to University of Southampton Student Paper	<1 %
40	bmcmedimaging.biomedcentral.com Internet Source	<1 %
41	openreview.net Internet Source	<1 %
42	Submitted to Liverpool John Moores University Student Paper	<1 %
43	graphite-note.com Internet Source	<1 %
44	keylabs.ai Internet Source	<1 %
45	trace.tennessee.edu Internet Source	<1 %
	uis.brage.unit.no	

46	Internet Source	<1 %
47	dokumen.pub Internet Source	<1 %
48	mdpi-res.com Internet Source	<1 %
49	Submitted to Multimedia University Student Paper	<1 %
50	Submitted to University of Sheffield Student Paper	<1 %
51	Xiaoding Guo, Hongli Zhang, Lin Ye, Shang Li. "RnnTd: An Approach Based on LSTM and Tensor Decomposition for Classification of Crimes in Legal Cases", 2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC), 2019 Publication	<1 %
52	cdn.techscience.cn Internet Source	<1 %
53	Meet Patel, Keyaba Gohil, Aditya Gohil, Fenil Ramoliya et al. "Explainable AI for gastrointestinal disease diagnosis in telesurgery Healthcare 4.0", Computers and Electrical Engineering, 2024 Publication	<1 %

54	Mohamed Lahby, Al-Sakib Khan Pathan, Yassine Maleh. "Combatting Cyberbullying in Digital Media with Artificial Intelligence", CRC Press, 2023 Publication	<1 %
55	link.springer.com Internet Source	<1 %
56	www.carmatec.com Internet Source	<1 %
57	www.tet.tuhh.de Internet Source	<1 %
58	Taskin Kavzoglu, Brandt Tso, Paul M. Mather. "Classification Methods for Remotely Sensed Data", CRC Press, 2024 Publication	<1 %
59	Submitted to University of South Africa (UNISA) Student Paper	<1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		

PLAGIARISM CHECK RESULT

Form Title: Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)

Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	HAZEL LIM BENIN
ID Number(s)	2200107
Programme / Course	Bachelor of Computer Science (Honours)
Title of Final Year Project	Hate Speech Detecting in Chinese Language using Deep Learning

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceed the limits approved by UTAR)**
Overall similarity index: 14 % **Similarity by source** Internet Sources: 10 % Publications: 7 % Student Papers: 7 %	
Number of individual sources listed of more than 3% similarity: 0	

Parameters of originality required, and limits approved by UTAR are as Follows:

(i) Overall similarity index is 20% and below, and
(ii) Matching of individual sources listed must be less than 3% each, and
(iii) Matching texts in continuous block must not exceed 8 words

Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.

Note: Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor vicky

Name: Ts Dr Vikneswary a/p Jayapal

Date: 13/9/2024

Signature of Co-Supervisor

Name: ____________________

Date: ____________________

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	22ACB00107
Student Name	HAZEL LIM BENIN
Supervisor Name	Ts Dr Vikneswary a/p Jayapal

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

____Hazel_______
(Signature of Student)
Date:13/9.24