PROMPT-BASED DATA AUGMENTATION FOR GAME REVIEW SENTIMENT ANALYSIS

ZHANG ZHERUI

UNIVERSITI KEBANGSAAN MALAYSIA

PROMPT-BASED DATA AUGMENTATION FOR GAME REVIEW SENTIMENT ANALYSIS

ZHANG ZHERUI

PROJECT SUBMITTED IN PARTIAL FULFILMENT FOR THE DEGREE OF MASTER OF DATA SCIENCE

FACULTY OF INFORMATION SCIENCE AND TECHNOLOGY
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

2024

PENAMBAHAN DATA BERASASKAN CEPAT UNTUK ANALISIS SENTIMEN SEMAKAN PERMAINAN

ZHANG ZHERUI

PROJEK YANG DIKEMUKAKAN UNTUK MEMENUHI
SEBAHAGIAN DARIPADA SYARAT MEMPEROLEHI
IJAZAH SARJANA SAINS DATA

FAKULTI TEKNOLOGI DAN SAINS MAKLUMAT
UNIVERSITI KEBANGSAAN MALAYSIA
BANGI

2024

DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries which have been duly acknowledged.

ZHANG ZHERUI
P130036

ACKNOWLEDGEMENT

Firstly, I would like to thank my supervisor, Professor Nazlia Omar, for her help, encouragement and guidance during my study at the UKM. She guided me to discover the research objectives through discussions and greatly improved the content quality of this paper through comments and suggestions.

In addition, I would like to thank all the staff and staff of the UKM. I would like to thank the teachers and classmates of the Data Science major of the School of Computer Science at the UKM for their help, friendship and creating a good working environment during my study at the UKM.

Finally, I would like to thank my family and friends for their support and encouragement.

ABSTRAK

Dalam bidang ulasan permainan, data mempunyai masalah semantik dan masalah ketidakseimbangan yang serius. Kajian ini berharap dapat memperbaiki kedua-dua masalah dalam analisis sentimen dalam bidang ulasan permainan melalui kaedah peningkatan data berasaskan segera untuk mencapai keputusan analisis sentimen yang lebih baik. Kajian ini memilih tiga kaedah analisis sentimen pembelajaran mesin, CNN, LSTM dan BERT, untuk memproses tugasan analisis sentimen dalam bidang semakan permainan sebagai tugas hiliran dan membandingkan kesan analisis sentimen kaedah peningkatan data berasaskan segera yang berbeza di bawah ketiga-tiga kaedah tersebut. Keputusan menunjukkan bahawa setiap peningkatan data berasaskan segera meningkatkan kesan analisis sentimen tugas hiliran. Antaranya, kaedah penambahbaikan yang mula-mula menjana ayat dengan kekutuban sentimen bertentangan sebagai tambahan kepada set data dan kemudian menulis semula semua data telah mencapai hasil terbaik, dengan berkesan mengurangkan masalah semantik dan masalah ketidakseimbangan dalam data dalam bidang ulasan permainan. Kajian ini menetapkan penambahan data berasaskan petunjuk sebagai kaedah yang boleh dipercayai untuk meningkatkan analisis sentimen ulasan permainan. Ia juga mengenal pasti jalan untuk penyelidikan masa depan untuk meningkatkan kebolehgunaan model, skalabiliti dan penggunaan praktikal dalam domain yang berbeza selain daripada permainan. Dengan menangani cabaran sedia ada dan mencadangkan hala tuju masa hadapan, kajian ini memberikan pandangan berharga ke dalam bidang aplikasi pemprosesan bahasa semula jadi dan pembelajaran mesin dalam analisis sentimen.

ABSTRACT

In the field of game reviews, data has semantic problems and serious imbalance problems. This study hopes to improve these two problems in sentiment analysis in the field of game reviews through a prompt-based data enhancement method to achieve better sentiment analysis results. This study selected three machine learning sentiment analysis methods, CNN, LSTM and BERT, to process sentiment analysis tasks in the field of game reviews as downstream tasks and compared the sentiment analysis effects of different prompt-based data enhancement methods under the three methods. The results show that each prompt-based data enhancement improves the sentiment analysis effect of downstream tasks. Among them, the enhancement method that first generates sentences with opposite sentiment polarity as a supplement to the data set and then rewrites all the data has achieved the best results, effectively alleviating the semantic problems and imbalance problems in the data in the field of game reviews. This study establishes hint-based data augmentation as a reliable method to improve sentiment analysis of game reviews. It also identifies avenues for future research to enhance the model's applicability, scalability, and practical deployment in different domains beyond gaming. By addressing existing challenges and proposing future directions, this study provides valuable insights into the application fields of natural language processing and machine learning in sentiment analysis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

Figure No. **Page**

LIST OF ABBREVIATIONS

UKM	Universiti Kebangsaan Malaysia
NLP	Natural Language Processing
CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
BERT	Bidirectional Encoder Representations from Transformers
GPT	Generative Pre-trained Transformer
QA	Question Answering
GAN	Generative Adversarial Network
Seq2Seq	Sequence-to-Sequence
MLM	Masked Language Model
GRU	Gated Recurrent Unit
SVM	Support Vector Machine
T5	Text-To-Text Transfer Transformer
BCE	Binary Cross Entropy
Adam	Adaptive Moment Estimation
ReLU	Rectified Linear Unit
API	Application Programming Interface
AutoDL	Automated Deep Learning

CHAPTER I

INTRODUCTION

1.1 RESEARCH BACKGROUND

As of 2024, the number of internet users worldwide has reached billions, with an increasing number of internet users from various social backgrounds and industries accustomed to expressing their comments, opinions, and views on social media platforms and e-commerce platforms. This presents new opportunities and challenges for text sentiment analysis (Ghous et al. 2023).

We can extract more valuable information from complex and massive natural language data, a process also known as opinion mining (Manke & Shivale 2015). This process entails extracting, processing, condensing, and dissecting text using subjective emotional expressions, then proceeding with logical deduction and evaluation. Its primary uses encompass categorizing texts, tracking public opinions online, and deriving opinion data. Employing text sentiment analysis for tasks aids in analyzing the emotions and viewpoints of text authors when they were publishing.

Tasks classifying text emotions hold considerable practical importance and societal relevance across different facets of human lives. As an illustration, within the business industry, firms can enhance their offerings in response to consumer input to tailor products to the client's requirements and deliver more individualized services, thus boosting loyalty among customers. Within the realm of culture, how to categorize emotions in movie, TV, and book reviews cuts down on the duration individuals dedicate to genres like movies, TV shows, and books, thus enabling the rapid discovery of useful information. Within social management, state agencies can gauge public responses to a policy by observing online comments, enabling them to suitably modify policies to adjust to societal shifts and more effectively benefit the broader populace. Furthermore, observing online users' remarks on a trending event and examining sentiment classification outcomes can steer web-based public views logically and establish a neat online setting.

In sentiment analysis, data augmentation techniques are key methods used to increase the diversity and richness of data, thereby improving the performance and generalization ability of models (Shorten & Khoshgoftaar 2019). Common data augmentation techniques include synonym replacement, sentence reversal, random insertion/deletion, text generation, sentence-level perturbation, and data synthesis. These techniques help improve model robustness, reduce overfitting, increase generalization ability, and improve model performance in sentiment analysis tasks. Through data augmentation, models can better handle various types and styles of text data, capture patterns and rules in the data more effectively, and thus better adapt to the challenges in practical applications.

Prompt-based data augmentation is particularly suitable for natural language processing tasks. It guides the model to generate the desired output by designing prompts instead of generating text based on the context of input data. In this approach, a prompt is first designed to guide the model in generating the expected output, and then the model is trained through fine-tuning or zero-shot learning. Once trained, the model generates text based on the designed prompts, using the information in the prompts to guide generation and produce the expected output. The advantage of prompt-based learning is that it helps the model better understand tasks and generate more accurate outputs, and it performs well in many NLP tasks such as question answering, text generation, and summarization (Duangvisate 2022).

Given the enormous success of GPT models, this study believes that using GPT-3.5-turbo for prompt-based data augmentation will yield excellent results.

The development of text data augmentation methods is driven by many contributors. Early rule-based methods benefited from pioneers in the field of natural language processing, such as Noam Chomsky and Alan Turing (sagar 2019), as well as later researchers like Karen Spärck Jones and Christopher Manning. With the rise of neural generation models, creators of Seq2Seq models include Ilya Sutskever (Naseem et al. 2021), Oriol Vinyals and Quoc V. Le, while the concept of generative adversarial networks (GAN) was proposed by Ian Goodfellow and others (Naseem et al. 2021). Contributors to self-supervised learning methods include Yoshua Bengio, Geoffrey Hinton, Yann LeCun and others (Davie 2024). In the field of pre-trained language models, Jacob Devlin proposed the BERT model, and Alec Radford proposed the GPT model (Naseem et al. 2021). The concept of adversarial training was proposed by Ian Goodfellow and others, while domain adaptation methods were explored by Daume III and others. In the field of multimodal data augmentation, the contributions of Andrej Karpathy and interdisciplinary team collaborations in image processing are noteworthy (Zheng 2023). These contributors collectively drive the continuous progress and development of text data augmentation methods. Through data augmentation, diversity of data can be increased, annotation costs can be reduced, and model performance and robustness can be improved. Specific methods include synonym replacement, sentence reorganization, masked language models, post-processing techniques, word insertion and deletion, and language model-based methods. These methods, combined, provide rich tools and techniques for text data augmentation, offering effective pathways for model training and performance improvement.

1.2 PROBLEM STATEMENT

In the domain of game reviews, sentiment analysis tasks face difficulties such as semantic understanding and data imbalance. The gaming market encompasses various types and genres of games, ranging from action games to role-playing games, strategy games, and more. Each type of game has its unique characteristics and audience. Different types of games may contain specific words or abbreviations that make it difficult for non-seasoned players to understand the meaning of these terms or expressions, thereby affecting further judgment of sentiment polarity.

The data imbalance problem is rooted in the uniqueness of the gaming review domain. Typically, games produced by well-known companies or highly recommended by gaming communities have higher popularity and player numbers. Usually, these games have a certain level of quality and a relatively stable player base. Such games often receive numerous positive reviews and recommendations. Consequently, more players join these games and provide feedback. These newly joined players usually have a high rate of positive reviews. As a result, games in the gaming domain tend to have significantly more positive reviews (with over 50% of reviews being positive) or even overwhelmingly positive reviews (with over 80% of reviews being positive) compared to games with predominantly negative reviews (with over 50% of reviews being negative). This imbalance in data poses a serious challenge for sentiment analysis in the gaming domain, where positive reviews outnumber negative ones. This situation can lead to many problems, such as the model tending to predict the majority class. Even though the overall accuracy of the model may appear high, its performance on the minority class will be very poor. For instance, if 90% of the data is positive, the model can achieve 90% accuracy simply by predicting all data as positive, but such a model is unreliable in practical applications. With fewer samples in the minority class, the model cannot learn sufficient features during training, resulting in poor performance in predicting the minority class. This leads to a higher misclassification rate for minority class samples. The model may perform inconsistently when handling new data, especially for the minority class. Imbalanced data affects the model's stability and generalization ability.

Through new methods like prompt-based GPT rewriting used as data augmentation, this study can alleviate this problem. Prompt-based data augmentation can mitigate data imbalance by using prompts to generate new data with pretrained language models, thereby supplementing or replacing the original dataset.

1.3 RESEARCH OBJECTIVE

The goals of this study can be encapsulated thus:

1. To identify effective prompts to augment the training set with high-quality generated data for sentiment analysis tasks.

2. To evaluate the quality of the augmented data by evaluating the performance of different deep neural network models, including convolutional neural networks

(CNNs), long short-term memory neural networks (LSTMs), and BERT pre-trained models, based on their performance in sentiment analysis tasks.

1.4 RESEARCH SCOPE

1. This study will primarily concentrate on modifying the prompts to enhance data based on these prompts. This study aims to assess how enhancing prompt-based data with varied prompts influences subsequent classification activities, employing three neural network classifiers: convolutional neural networks (CNNs), long-term memory neural networks (LSTMs), and BERT pre-trained models. The primary objective of this study is to improve the precision of subsequent classification activities through the modification of prompt terms for immediate data augmentatio.

2. The study will employ the dataset of gaming reviews on Steam found on Kaggle.com. The dataset contains data like the game's name on Steam, review timestamps, text review details, and user endorsements.

1.5 RESEARCH SIFINICANT

The results of this study will assist researchers in comprehending customer requirements and reactions, mirroring the diverse effects it has on the gaming environment. Initially, examining emotions assists in detecting prospective market patterns and user interests, offering game creators chances to refine their tactics and enhance the quality of products. Developers can adjust to market shifts with faster responses by swiftly assessing players' emotional responses, aligning the games with user anticipations. Additionally, this endeavor aids in fostering stronger bonds within the community. Game developers, by attentively hearing out player perspectives and reacting to their input, can create a favorable brand perception, thereby boosting customer loyalty and involvement. Creating a transparent communication channel can aid gaming companies in building improved trust with gamers, enhancing community evolution and expansion. Furthermore, this study offers crucial perspectives for both creators and marketers of game content. Through grasping the emotional responses of players to various aspects of games and marketing tactics, developers and advertisers can refine approaches to amplify the appeal of the content and the impact on the market.

This refined market research and user feedback analysis contribute to improving the competitiveness of games and promoting the sustainable development of the gaming industry.

The research in this study addresses the semantic specificity, semantic ambiguity, and data imbalance issues in sentiment analysis in the gaming domain. The study directly contributes to the empirical testing of prompt selection for prompt-based data augmentation and its impact on sentiment analysis tasks. Researchers can replicate and carefully select prompts that fit their experimental designs.

1.6 THESIS ORGANIZATION

The study chapters are organized as follows:

1. **Chapter I**: Introduction to the research background and related studies. Identification of limitations in sentiment analysis tasks in the gaming review domain within the current research context, along with proposed improvements. Emphasis on the contribution of this study towards addressing the identified limitations. Presentation of research objectives, scope, and the overall research approach.
2. **Chapter II**: This chapter will describe different data augmentation techniques and the neural network technologies used. The main focus of this study is prompt-based data augmentation. This chapter will also compare prompt-based data augmentation with some other data augmentation techniques and list the uses of prompt-based data augmentation. Additionally, it will briefly introduce the fundamental models and methods relevant to this study, including the basic structures of Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and their variant, Long Short-Term Memory (LSTM) networks, as well as the loss functions. The chapter will also cover the BERT pre-training model from the perspectives of the self-attention mechanism and Transformer. Finally, it will present related research on the application of prompt-based data augmentation in sentiment analysis.
3. **Chapter III**：This chapter mainly introduces the experimental design and the specific methods used in the process. First, it describes the data preprocessing

process for game review data, detailing how prompt-based data augmentation is implemented. Then, it introduces two neural network methods, CNN and LSTM, as well as the BERT pretrained model in the context of sentiment analysis of game review data, including hyperparameter tuning. Afterward, it briefly describes the evaluation metrics used to assess the performance of the algorithms.

4. **Chapter IV** : This chapter provides a detailed description of the experimental details and results. In terms of experimental details, it first introduces the dataset used in the experiments, followed by the experimental setup, including the configuration of the experimental platform, API settings, prompt settings, and optimizer parameters. Regarding the experimental results, an ablation study is conducted on the three methods proposed in this study to explore the optimal hyperparameter configuration. Then, a quantitative comparison of the three proposed methods is presented. Finally, the parameter count and running speed of sentiment analysis after different data augmentations are compared, and their limitations are summarized.
5. **Chapter V** : The experimental results are summarized to show the achievement of the expected goals. The limitations and deficiencies of the experimental design are summarized, and finally some possible future work is proposed.

CHAPTER II

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter will describe different data augmentation techniques and the neural network technologies used. The main focus of this study is prompt-based data augmentation (Wang, Wang & Lian 2019). This chapter will also compare prompt-based data augmentation with some other data augmentation techniques and list the uses of prompt-based data augmentation.

Additionally, it will briefly introduce the fundamental models and methods relevant to this study, including the basic structures of Convolutional Neural Networks (CNNs) (Li et al. 2021), Recurrent Neural Networks (RNNs) (Kaibing Zhang et al. 2020), and their variant, Long Short-Term Memory (LSTM) networks, as well as the loss functions (Kun Bu, Yuanchao Liu & Xiaolong Ju 2023) The chapter will also cover the BERT pre-training model from the perspectives of the self-attention mechanism and Transformer.

Finally, it will present related research on the application of prompt-based data augmentation in sentiment analysis. The structure of this chapter is as follows: 2.1 Chapter Introduction 2.2 Data Augmentation Techniques 2.3 Relevant Neural Network Technologies 2.4 Research on Prompt-Based Data Augmentation 2.5 Chapter Summary.

2.2 DATA AUGMENTATION TECHNIQUES IN NLP

2.2.1 Data Augmentation Techniques in NLP

Data augmentation plays a crucial role in NLP by generating or transforming training data to improve the performance and generalization ability of models (Zhiyuan Liu, Yan Kai Lin & Maosong Sun 2020). Each method has its unique advantages and disadvantages and is suitable for different application scenarios. Below are descriptions of some common data augmentation techniques.

Synonym Replacement is a simple and effective technique that increases data diversity by replacing words in a sentence with their synonyms. This allows the model to better understand and process synonyms, thereby recognizing the same semantic meaning in different expressions. However, this method relies on the quality and coverage of the synonym dictionary, and synonyms may not always be fully equivalent in some contexts, leading to slight semantic changes. For example, the sentence "The quick brown fox jumps over the lazy dog." can be augmented to "The speedy brown fox leaps over the lazy dog (Yan Yan & Hong-yan Xing 2021)."

Back Translation generates new sentences with consistent semantics but different expressions by translating a sentence into another language and then back to the original language. This method can significantly increase data diversity and improve model robustness, but it is computationally expensive, and the quality of the translation model directly affects the back translation results. For example, the original sentence "The weather is nice today." can be translated into French and back to generate "The weather is pleasant today." (Zhou et al. 2021)

Noise Injection is another common technique that enhances data by randomly deleting (Chemchem & Drias 2015), swapping, or inserting words. This method improves the model's noise resistance, helping it perform better when faced with noisy data in real applications. However, too much noise can generate meaningless data, so the type and degree of noise must be carefully controlled. For example, "The quick brown fox jumps over the lazy dog." can be augmented to "The quick brown jumps fox over the lazy dog."

Text Generation uses generative models (like GPT) to generate large amounts of new data, covering more linguistic phenomena and expressions, thus enhancing the model's generalization ability. However, the quality of generated data can be unstable, possibly resulting in low-quality or irrelevant data, which requires post-processing and filtering. For example, the sentence "The cat sat on the mat." can be augmented to "The feline rested on the rug."

2.2.2 Prompt-Based Data Augmentation

Prompt-based Data Augmentation uses pre-trained language models to generate diverse and high-quality data by designing specific prompts (Seo et al. 2024). Its applications include sentence completion, generating question-answer pairs, rephrasing, sentence expansion, and dialogue generation. For example, it can generate context-relevant complete sentences from partial sentences or generate related questions and answers from given sentences to increase question-answer pair data for training QA systems. Rephrasing generates different ways of expressing sentences, sentence expansion generates more detailed sentences, and dialogue generation increases dialogue data, enhancing the training effect of dialogue systems.

Prompt-based Data Augmentation can generate context-relevant and semantically consistent sentences, enriching the diversity of the dataset, providing various expressions, improving the model's ability to understand and process different expressions, and generating high-quality question-answer pairs, expanded sentences, and dialogue content, thereby enhancing the dataset for specific tasks. This method relies on the powerful capabilities of pre-trained language models to generate natural and meaningful text variants, making it an essential tool for modern NLP data augmentation. However, prompt design is complex, requiring expertise and experimental tuning, and calling large-scale pre-trained models to generate data requires substantial computational resources (Duangvisate 2022).

Prompt-based Data Augmentation generates new data samples using generative models (like GPT-3, BERT). For example, GPT-3 can generate new sentences with the same sentiment label based on a given sentiment label and example sentence; BERT's masked language model (MLM) feature can generate new sentences or replace parts of sentences through masking and prediction. Template generation creates new sentences through preset templates, while prompt learning guides generative models to generate data using prompts.

The main difference between Prompt-based Data Augmentation and other data augmentation techniques is its ability to generate high-quality and context-relevant data using pre-trained language models. By designing specific prompts, this method can generate diverse and semantically coherent data, suitable for complex tasks like dialogue generation and question-answer pair generation. In contrast, other data augmentation techniques like synonym replacement, back translation, noise injection, and text generation increase data diversity by replacing synonyms, bidirectional translation, introducing noise, and generating new text, respectively, but typically lack the ability to understand context and generate high-quality relevant data. Therefore, Prompt-based Data Augmentation is more suitable for tasks requiring high-quality and diverse data, while other techniques are suitable for simpler data augmentation needs.

Using GPT for data augmentation is closely related to Prompt-based Data Augmentation. GPT (Generative Pre-trained Transformer) is a powerful pre-trained language model commonly used for Prompt-based Data Augmentation. Prompt-based Data Augmentation relies on designing specific prompts to guide pre-trained language models (like GPT) to generate the required augmented data. GPT provides robust generative capabilities for Prompt-based Data Augmentation, producing high-quality and context-relevant text that significantly enhances dataset diversity and model generalization. In this study, GPT is used as the pre-trained model for Prompt-based Data Augmentation.

Table2.1 Examples of Augmentation Method

Original Sentence	**Augmentation Method**	**Augmented Sentences**

to be continued…

…continuation

She quickly finished her work.	Synonym Replacement	She rapidly finished her work.
		She promptly finished her work.
		She swiftly finished her work.
The beautiful scenery amazed the tourists.	Synonym Replacement	The stunning scenery amazed the tourists.
		The breathtaking scenery amazed the tourists.
		The picturesque scenery amazed the tourists.
He decided to buy the expensive car.	Synonym Replacement	He decided to purchase the costly car.
		He decided to acquire the pricy car.
		He decided to buy the pricey automobile.
She loves to paint during the weekends.	Back Translation	She likes painting during the weekends.
		She enjoys painting on weekends.
		She loves painting on the weekends.
The children played in the park all afternoon.	Back Translation	The kids played in the park all afternoon.
		The children were playing in the park all afternoon.

to be continued…

…continuation

		The children played at the park all afternoon.
The quick brown fox jumps over the lazy dog.	Noise Injection	The quick brown fox jumps over lazy the dog.
		The brown quick fox jumps over the lazy dog.
		The quick brown fox jumps over the dog lazy.
She enjoys reading books in her free time.	Noise Injection	She enjoys reading books in her time free.
		She enjoys reading in books her free time.
		She enjoys books reading in her free time.
The weather today is very nice.	Noise Injection	The weather today very is nice.
		The weather is today very nice.
		Today the weather is very nice.
The cat sat on the mat.	Text Generation	The feline rested on the rug.
		The kitty perched on the carpet.
		The cat lounged on the mat.
She loves to play the piano.	Text Generation	She adores playing the piano.
		She is passionate about playing the piano.
		She enjoys performing on the piano.
They went for a walk in the park.	Text Generation	They took a stroll in the park.
		They enjoyed a walk in the park.

to be continued…

…continuation

The quick brown fox jumps over the lazy dog because	Sentence Completion	it saw a squirrel on the other side.
		it wanted to reach the other side of the fence.
		it was being chased by a hunter.
She opened the door and	Sentence Completion	saw a surprise party waiting for her.
		found her lost dog sitting on the porch.
		was greeted by a warm breeze.
Alice went to the market to buy some fresh vegetables.	Question-Answer Pairs	Why did Alice go to the market? - To buy some fresh vegetables.
		What did Alice buy at the market? - Fresh vegetables.
		Where did Alice go to buy fresh vegetables? - The market.
John finished his work early.	Question-Answer Pairs	Why did John finish his work early? - Because he worked efficiently.
		What did John do early? - He finished his work.
		Who finished their work early? - John.
Can you rephrase the following sentence? 'The weather is quite pleasant today.'	Paraphrasing	The weather today is very nice.
		Today, the weather is very agreeable.
		It's quite a lovely day today.
Rephrase the sentence: 'She quickly ran to the store.'	Paraphrasing	She hurried to the store.
		She swiftly went to the store.

to be continued…

…continuation

Expand the sentence: 'She was happy.'	Sentence Expansion	She was happy because she had just received good news about her job application.
		She was happy, having spent a wonderful day with her friends.
		She was happy, enjoying the beautiful weather and a relaxing afternoon.
Expand the sentence: 'He decided to leave.'	Sentence Expansion	He decided to leave after the meeting ended.
		He decided to leave, feeling it was the best choice.
		He decided to leave, knowing he had done all he could.
A: Hi! How are you today? B:	Dialogue Generation	B: I'm doing well, thank you! How about you?
		B: I'm great, thanks for asking! What about you?
		B: I'm fine, just a bit tired. How are you?
A: What are your plans for the weekend? B:	Dialogue Generation	B: I'm planning to go hiking. What about you?
		B: I'll be visiting some friends. How about you?
		B: I'm going to relax and catch up on some reading. You?

These examples demonstrate how different data augmentation methods can generate diverse training data, thereby improving model performance and generalization ability. Each method has its specific application scenarios and advantages.

Different data augmentation methods significantly increase data diversity, enhance model generalization (Zhu-li Ren, Jin-long Zhang & Rui-fu Yuan 2024) and improve robustness. However, they also have their respective drawbacks, such as potentially introducing semantic changes, high computational costs, and unstable generation quality. Therefore, in practical applications, it is essential to select appropriate data augmentation methods based on specific tasks and data characteristics, and to optimize and adjust accordingly to achieve the best results

2.3 BASIC STRUCTURE OF CNN

CNN is a classical deep learning model, which is used to process data with hierarchical structure, such as images, audio and video, etc (Suresha, Kuppa & Raghukumar 2020). CNN usually consists of convolutional layer, pooling layer, fully connected layer, etc (Derry, Krzywinski & Altman 2023). The pooling layer is used to down sample the input features to increase the receptive field and reduce the channel dimension of the features to reduce the number of parameters for the subsequent network layer processing (Faroog Zaman et al. 2020). The pooling layer is used to down sample the input features, increase the receptive field and reduce the channel dimension of the features (Yuqing Wang 2023) which reduces the number of parameters for the subsequent network layer processing; the fully connected layer performs a linear transformation of the input features, which has the ability of global feature sensing, in the text sentiment analysis task, the output dimension of the fully connected layer is consistent with the number of classification categories, and the output of each neuron is the probability that the text belongs to a certain category; the convolutional layer is the core of the CNN that is used to extract features from the input raw data (Zhang Fang et al. 2020). In the convolution process, the center position of the convolution kernel is taken as the benchmark, and each convolution only operates on the local pixels about the input data, and by adjusting the position of the convolution kernel, the convolution operation is gradually applied to all the positions of the input data, so as to obtain the complete feature map of the input data (Hongchan Li & Yu Ma et al. 2021). The convolution operation can effectively extract the local features of the input data, and by increasing the number of convolution kernels, the convolution layer can extract multi-channel features at the same time and combine them into a higher-level feature map (Shaika Chowdhury, Chenwei Zhang & Philip S. Yu 2021).

2.4 RNN AND THEIR VARIANTS

2.4.1 Basic Structure of RNN

RNN is a type of neural network model that is very common in sequence data modeling and processing. It can store contextual information in sequential data for processing while considering previous inputs (Dong Fang Ma et al. 2018). The basic structure of RNN is shown in Figure 2.1. It is a cyclic unit that generates the hidden state of the current time step by combining the current input with the hidden state of the previous time step. The concealed status is identifiable as whether the network's memory or state is continually refreshed and disseminated over time. This iterative framework permits RNN to preserve specific past data during the processing of sequential data and to convey information over time intervals.

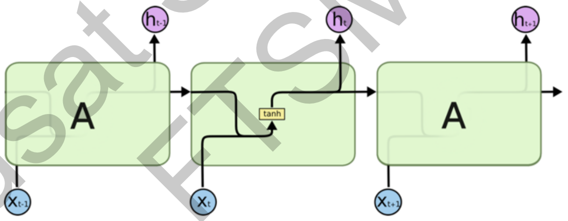


Figure 2.1 Basic structure of RNN

Within RNN systems, the activation function determines the modification of concealed states. At every interval, the RNN acquires the concealed condition of the present input and the prior one, determines the hidden condition of the ongoing moment, and forwards it to the subsequent phase. The recursive nature of this model enables RNN to produce outputs using both past and present inputs to simulate and forecast sequence data (Okologume & Esue 2022).

The RNN technique is extensively employed in natural language processing (NLP) to handle processes like linguistic modeling, machine translation, and emotional analysis(Xin Guo et al. 2023). This is also applicable for simulating additional sequential data, like predicting time series, processing audio, and so forth. Traditional

RNNs face challenges due to fading or excessive surges in gradient when managing extended dependencies, restricting their effectiveness in real-world scenarios (Yang Zhang et al. 2023).

In tackling this challenge, various RNN models have surfaced, including LSTM and Gated Recursive Unit (GRU) (Ahmed et al. 2023). Such modifications enhance RNN's efficacy in handling extended sequences by incorporating gating mechanisms to precisely manage information transfer and deletion (Alzahrani 2023).

2.4.2 Long Short-Term Memory Network

LSTM is a special type of RNN that learns long-term dependent information0. An LSTM memory unit is also called a cell state, which serves to provide the memory capacity to remember previous information (Fan Zhang 2024). The LSTM unit consists of three parts: an input gate, an oblivion gate, and an output gate. It adds or removes information to or from the memory cell through a gate structure that allows different pieces of information to selectively pass through (Hongjie Deng 2022). The schematic diagram of an LSTM cell is shown in Figure 2.2:

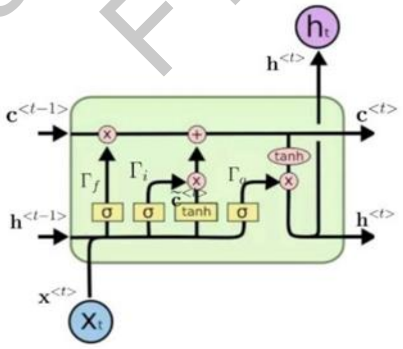


Figure 2.2 Basic structure of the LSTM unit

Among other things, the oblivion gate (Li Lyu et al. 2021) denoted by Γ_f , determines what information is discarded from the cell state. The oblivion gate outputs a value from 0 to 1 to the cell state based on $\boldsymbol{h}^{<\text{t-1}>}, \boldsymbol{x}^{<\text{t}-1>}\ \boldsymbol{c}^{<\text{t-1}>}$. 1 means completely retained and 0 means completely discarded. Its formula is:

$$\Gamma_{\text{f}} = \boldsymbol{\sigma}(\boldsymbol{W}_{\text{f}}\left[\boldsymbol{h}^{<\text{t-1}>},\quad \boldsymbol{x}^{<\text{t}>}\right] + \boldsymbol{b}_{\text{f}})$$

The input gate (memory gate)(Mateusz Kochanek et al,2024) 1denoted by Γ_{i} , is divided into two layers, which determine what kind of information is stored in the cell state. One of these layers, the Sigmoid layer, determines what values will be updated and is calculated as:

$$\Gamma_{\text{i}} = \boldsymbol{\sigma}(\boldsymbol{W}_{\text{i}}\left[\boldsymbol{h}^{<\text{t-1}>},\quad \boldsymbol{x}^{<\text{t}>}\right] + \boldsymbol{b}_{\text{i}})$$

The Tanh layer creates a new candidate cell unit $\hat{\boldsymbol{c}}^{<t>}$, calculated as:

$$\hat{\boldsymbol{c}}^{<t>} = \tanh(\boldsymbol{W}_{\text{c}}\left[\boldsymbol{h}^{<\text{t-1}>},\quad \boldsymbol{x}^{<\text{t}>}\right] + \boldsymbol{b}_{\text{c}}])$$

Multiply the forgetting gate with the old state, forgetting the information determined to be discarded; and multiply the memory gate with the candidate cellular units, determining the information to be updated. The two parts are added and summed to update the old cell state $\boldsymbol{c}^{<\text{t-1}>}$ to $\boldsymbol{c}^{<\text{t}>}$ (Kaibing Zhang et al 2020). The formula is:

$$\boldsymbol{c}^{<\text{t}>} = \Gamma_{\text{f}} \times \boldsymbol{c}^{<\text{t-1}>} + \Gamma_{\text{i}} \times \hat{\boldsymbol{c}}^{<t>}$$

The output gate(sepp Hochreiter&Schmidhuber, 1997)i.e. determining what value to output, is denoted by Γ_{o} . The Sigmoid activation function is first used to determine which part of the cell state to output31. The formula is calculated as:

$$\Gamma_{\text{o}} = \boldsymbol{\sigma}(\boldsymbol{W}_{\text{o}}\left[\boldsymbol{h}^{<\text{t-1}>},\quad \boldsymbol{x}^{<\text{t}>}\right] + \boldsymbol{b}_{\text{o}})$$

Finally, the cell state is processed using tanh as an activation function and then multiplied with the output gate to get the output:

$$\boldsymbol{h}^{<t>} = \Gamma_o \times \tanh(\boldsymbol{c}^{<t>})$$

2.4.3 Bidirectional Long Short-Term Memory Network

Within unidirectional recurrent neural networks, the framework solely incorporates data "above," disregarding the "below" (Vaishali Baviskar et al. 2024). Yet, for making precise forecasts in practical scenarios, it's essential to amalgamate the entire data sequence. Consequently, the Bi-Bi-LSTM framework presents a 'bi-directional' approach to computing feature computation (Shaika Chowdhury, Chenwei Zhang, Philip S. Yu, 2017) diverging from the conventional LSTM structure. The framework analyzing word contextual data involves concurrently entering textual data from both consecutive forward and backward directions, aimed at more effectively conveying the sentence's characteristics. This design, which supports two directions, enhances the model its ability to grasp the sentence's semantic and syntactic content accurately. The integration of forward and backward computation outcomes within the bidirectional LSTM model enables a more comprehensive and integrated depiction of sentences, thus enhancing its modeling strength for tasks involving natural language handling.

2.5 BERT NETWORK ARCHITECTURE

2.5.1 Self-attention Mechanism

Attention processes are drawn from the unique features of the human visual system. Upon observing the current scene, humans tend to not be completely vigilant of the scene's entire elements, instead choosing to concentrate specifically on their areas of interest. The primary function of the attention mechanism involves understanding the significance of every element via an array of weighted parameters, and then combining these elements according to their importance tiers (Davis 2023). The introduction of an attention mechanism in natural language processing tasks began with the allocation of varying weights to elements in the input sequence, enabling the model to concentrate on more critical elements, thereby enhancing its representational capabilities. This method allows the attention mechanism to better represent crucial elements in the input sequence, while giving credence to parts of lesser significance (Jin Zhang, zekang Bian & Shitong Wang 2022). Implementing this method has notably advanced natural language processing, enabling the model to more effectively manage intricate semantic connections and linguistic frameworks, thereby enhancing its performance and efficacy.

BERT employs the self-attention technique to model interconnections among sequential data elements, a method originally prevalent in natural language analysis (Yang et al. 2021). The fundamental concept behind the self-concern mechanism involves calculating the correlation weights for every sequence element and integrating these weights among all elements. Unlike the conventional focus mechanism, the self-attention mechanism examines not just the interconnections among elements in a sequence but also determines the degree of correlation among each element and others via learning.

In the self-attention mechanism, the correlation between each element and other elements is calculated by introducing three learnable matrices: the Query Matrix (Q), the Key Matrix (K) and the Value Matrix (V). Its basic structure is shown in Figure 2.3 (Uday Kamath & John Liu 2019). Specifically, the self-attention mechanism obtains the similarity scores between elements by performing dot-product operation on the query matrix and the key matrix, and then normalized by the Softmax function to obtain the attention weights of each element to other elements. Finally, the attention weights are multiplied with the value matrix and weighted and summed to obtain a weighted representation of the current element (Han et al. 2023).

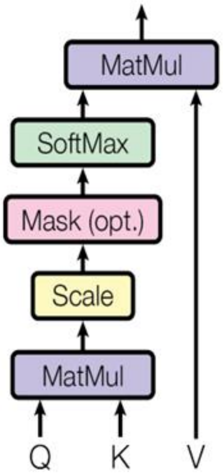


Figure 2.3 Self-attention mechanism calculation flow

2.5.2 Transformer

Transformer is a model that utilizes the attention mechanism to increase the speed of model training, it is completely based on the self-attention mechanism and has a more complex model structure (Lohrenz et al. 2023), compared with the previous popular recurrent neural networks, there is a significant improvement in accuracy and performance (Ye, Xie & Chen 2019). The core composition of Transformer is a module consisting of multiple encoders and decoders. For each text data input to Transformer, it is first encoded by the encoder module, and then the encoded data is passed to the decoder module for decoding, and finally the translated text result is obtained. The structure of the whole Transformer module is shown in Figure 2.4.

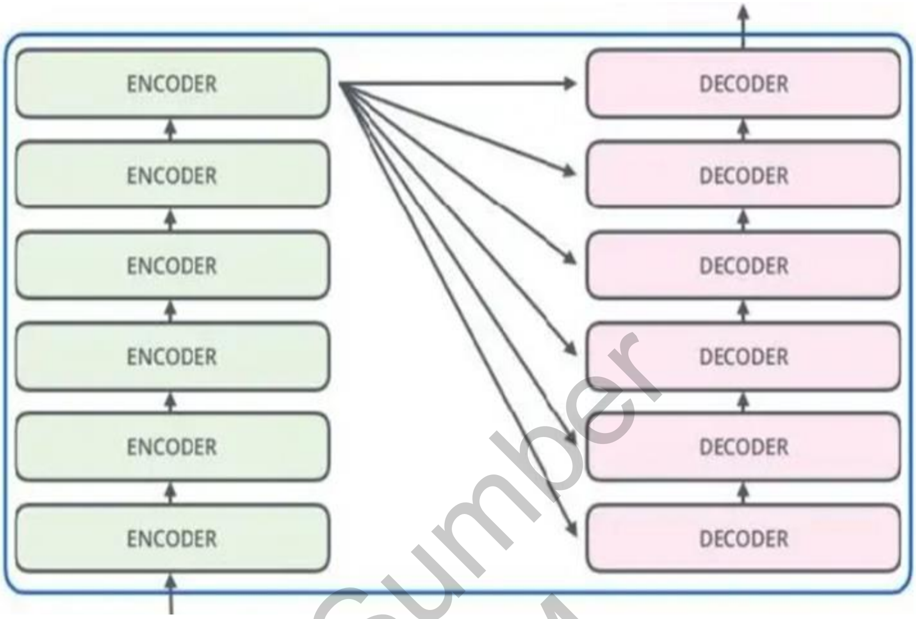


Figure 2.4 Schematic diagram of the complete Transformer module

As can be seen from the figure, in the encoding part, the input of each encoder is the output of the previous encoder, while the output of each decoder contains both the output of the previous decoder and the outputs of all encoders. This hierarchical structure is designed so that the Transformer can fully utilize the contextual information to better express the relationship between sequences.

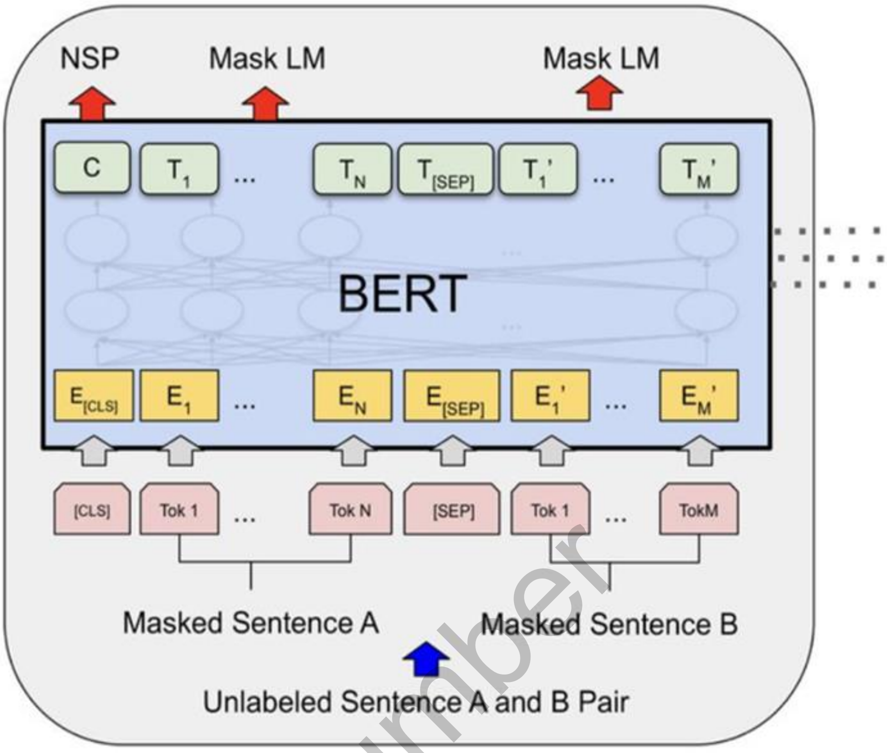
Every encoder is comprised of a mechanism for self-focused attention and a neural network that feeds in. The mechanism of self-attention is utilized to calculate the relationships among elements in the sequence and create appropriate attention weights, enhancing the understanding of the dependencies among these elements. Conversely, the feed-forward neural network serves to modify the output tensor's dimensionality for its input into the subsequent encoder module.

The decoder's architecture is akin to that of the encoder, except it requires calculating both the self-attention score and the attention score in relation to the decoder's output. This is crucial to fully engage the data from the encoder and its predecessor during the decoding stage.

Addressing the issue of gradient disappearance, the encoder and decoder both implement the configuration of residual networks. Essentially, each feedforward neural network's input encompasses not just self-attention outputs but also the most basic input, enhancing the maintenance of original information and simplifying the model's training and optimization process. The introduction of the self-attention system and residual connectivity enhances the Transformer model's capability in identifying distant dependencies within sequences, thereby boosting its representational proficiency and training efficacy (Islam et al. 2024). It has made many important breakthroughs in areas such as natural language processing and computer vision and has become one of the most advanced models currently available.

2.5.3 BERT

BERT is a pre-trained language representation model that innovates based on traditional pre-trained language models (Borges Scoggin & Torres Marques-Neto 2023). Traditional pre-trained language models usually use a unidirectional language model or a shallow splicing of two unidirectional language models. In contrast, BERT overcomes the limitations of unidirectional language models by adopting a new Masked Language Model (MLM) (Wei, Wang & Kuo 2023), which aims to enable BERT to merge left and right contextual information, thus pre-training a deep bi-directional Transformer and generating deep bi-directional linguistic representations that can synthesize left and right contextual information. The pre-trained BERT model can be fine-tuned by an additional output layer without making drastic modifications to the model, and thus is applicable to a wide range of tasks, including question-answer tasks and linguistic reasoning tasks, etc. The model structure of BERT is shown in Figure 2.5, which mainly consists of multiple Transformers stacked on top of each other.



3 Figure 2.5 BERT model structure

The input to BERT is a vector representation corresponding to each word (token). The vector can explicitly represent a single text sentence or a pair of text sentences. In order to obtain the vector representation of the input token, the text first needs to be sliced and diced using the WordPiece method based on vocabularys to obtain a dictionary of words. Subsequently, the start symbol CLS is inserted at the beginning of the input sequence, which corresponds to the output of the last Transformer layer and is used to aggregate the whole sequence and feature information for the sequence sentiment analysis task. This symbol is ignored for non-classification tasks. In addition, BERT represents different sentences by inserting the segmentation symbol SEP after each sentence.

For different tasks, the output of BERT is slightly different. The C in the upper left of Figure 2.5 represents the output of the last Transformer of the classification token ([CLS]), indicating the output of the last Transformer of the other tokens. Therefore, for character-level based tasks (e.g. sequence annotation, question and answer tasks, etc.) (Liu Y, Yin Y & Zhang S 2024) it can be fed into the subsequent network for prediction; for sentence-level based tasks (e.g., natural language inference, sentiment categorization tasks, etc. (Future Generation Computer Systems 2024) C can be fed into the subsequent network for prediction. This task is an sentence-level two-classification task, so C is eventually fed into the fully connected network for classification prediction.

2.6 RELATED WORK

Next, this study introduces the evolution of machine learning sentiment classification methods in chronological order, as well as related research on data enhancement, especially prompt-based data enhancement, to improve sentiment analysis.

Hopfield et al. introduced neural network models, laying the foundation for deep learning research (Zamzami, Himdi & Sabbeh 2023). This pivotal work would later become integral to advancements in natural language processing and sentiment analysis. Bruce et al. investigated the subjectivity of text recognition using artificial labels, contributing to early efforts in distinguishing subjective content in text (Bruce et al. 1999). Their work provided a framework for understanding how machine learning could be applied to detect sentiment in various forms of textual data. Pang et al. pioneered the application of machine learning algorithms to text sentiment classification, analyzing text features. They demonstrated that Support Vector Machine (SVM) models achieved superior classification performance, setting a benchmark for future sentiment analysis research (Kaibing Zhang et al,2020). Turney et al. utilized unsupervised learning algorithms for sentiment classification of review texts, marking a shift from rule-based methods to machine learning approaches (Ullah, Khan & Nawi 2022). This research highlighted the potential of unsupervised methods in automatically determining sentiment without extensive labelled data. Kushal and colleagues utilized semantic classification methods for textual reviews, leading to remarkable outcomes in classification and the enhancement of sentiment analysis (Ullah, Khan & Nawi 2022). Their work underscored the importance of semantic understanding in improving the accuracy of sentiment classification systems. Beineke refined text sentiment classification techniques through the use of manually annotated data, thereby boosting the precision and dependability of sentiment analysis (Beineke 2004). This advancement emphasized the value of high-quality annotated datasets in training effective sentiment classifiers. Sanjiv introduced a method for deriving emotions from textual data on networks such as Yahoo and Amazon (Sanjiv 2007), demonstrating the real-world utility of sentiment analysis in data from actual scenarios (Huang Qin 2019). This practical approach showcased the applicability of sentiment analysis to large-scale, real-world data. Moens crafted a machine learning technique for classifying sentiments in multiple languages, broadening the applicability of sentiment analysis to varied languages and cultural contexts (li Meng & Qing 2019). This research was crucial in demonstrating the potential for multilingual sentiment analysis. Polpinij unveiled a digital method for sentiment classification in consumer reviews, investigating how diverse linguistic factors, such as verbs and nouns, influence sentiment categorization activities (Polpinij 2008). This study provided insights into the linguistic elements that play a critical role in sentiment detection. Hinton introduced an innovative unsupervised deep belief network, demonstrating the capability of deep learning to identify intricate data patterns and characteristics (Hinton 2012). This work markedly impacted sentiment analysis methods by showing how deep architectures could uncover complex structures in text data. Kim pioneered the utilization of convolutional neural networks (CNNs) in sentiment analysis, employing sliding window methods for extracting word vectors (Kim 2014). This approach accurately distinguished text sentiment and highlighted the power of CNNs in capturing local textual features. Kalchbrenner employed convolutional neural networks, equipped with dynamic pooling layers, for emotional analysis (Kalchbrenner 2014). This improved the model's capacity to identify and decode specific semantic features within text, enhancing sentiment classification performance. Irsoy introduced an innovative recurrent neural network (RNN) designed for learning time series data, enhancing the mapping of text word vectors by identifying time-related dependencies (Irsoy 2014). This development was significant for capturing sequential patterns in text for better sentiment analysis. Xu incorporated a caching feature into the LSTM framework, tackling the deficiency in storage capacity (Xu 2016). This enhancement improved the model's capacity to preserve emotional data from textual content over longer sequences. Wang integrated the focus mechanism into the LSTM model, showcasing a notable enhancement in the precision and effectiveness of text sentiment identification (Wang 2016). This was achieved through the use of diverse educational methods, demonstrating the power of attention mechanisms. Google, in

2017, debuted the Transformer approach using the attention mechanism, which addressed time complexities in text analysis. This model lessened the need for computational demands and enhanced the efficiency of parallel processing, revolutionizing sentiment analysis. Fadaee developed methods of template-based data augmentation to improve sentiment categorization. This resulted in varied and efficient training datasets (Fadaee 2017), boosting the model's efficiency and effectiveness in sentiment classification tasks. Vaswani employed Transformer models to enhance sentiment data creation, thereby improving the training datasets (Vaswani 2017). This work showed superior results in sentiment classification assignments and set a new standard for model architectures. Tian introduced a bidirectional GRU model for classifying text sentiment, enhancing its accuracy and making it more understandable (Tian 2018). This model leveraged the strengths of bidirectional processing to capture context more effectively. Wang developed the DRNN model by merging CNN and RNN capabilities, aiming to minimize overfitting and improve text sentiment classification (Wang 2018). This hybrid approach combined the strengths of both architectures for better performance. Edunov employed reverse translation methods to produce varied sentiment analysis training datasets, which notably enhanced model efficiency in environments with limited resources by creating diverse training examples (Edunov 2018). Jiang crafted the LSTM-CNN framework utilizing the attention mechanism to identify interdependencies among features, boosting the precision of text emotion categorization by focusing on important parts of the text (Jiang 2019). Cao et al. introduced a sentiment analysis technique utilizing BGRU, which proved to be more accurate and rapid in training than CNN and BiLSTM models, highlighting the efficiency and accuracy of bidirectional GRUs in sentiment analysis tasks (Cao et al. 2019). Yang & Cui utilized Generative Adversarial Networks (GANs) to produce superior sentiment data, thereby amplifying the efficiency of sentiment analysis models during adversarial training. This innovative use of GANs demonstrated the potential for generating high-quality synthetic data (Yang & Cui 2019). Raffel and colleagues employed the T5 model to convert sentiment analysis into tasks involving text creation, which enhanced training data and boosted the model's efficiency, showcasing the versatility of the T5 model in NLP tasks (Raffel et al. 2019). Radford utilized GPT-2 for creating sentiment information, showing marked enhancements in sentiment analysis methods. The generation of sentences with distinct sentiments highlighted the capabilities of generative models in augmenting training datasets (Radford 2019).

Xu introduced the CNN-Text-Word2vec framework for managing Weibo information, surpassing conventional models in analyzing sentiment. This novel approach effectively combined CNNs with Word2vec embeddings to capture semantic nuances in social media text, leading to superior performance in sentiment classification tasks (Xu 2020).

Wu & Wang employed BERT's Masked Language Model (MLM) to augment data, creating fresh sentiment data samples to boost the model's capability. Their innovative use of MLM for data augmentation generated diverse training examples, significantly enhancing the robustness and accuracy of sentiment analysis models (Wu & Wang 2020).

Brown and colleagues employed GPT-3 for creating varied sentences aligned with sentiment labels, greatly enhancing the precision and applicability of sentiment analysis models. This work showcased the power of GPT-3 in generating high-quality, sentiment-specific text, providing substantial improvements in model performance across various datasets (Brown et al. 2020).

Fan utilized the FastText algorithm along with a bidirectional GRU recurrent neural network for analyzing sentiment in Chinese short texts, attaining impressive results in fitting and generalization. This combined approach leveraged the strengths of both FastText and GRU networks, achieving high accuracy and robustness in sentiment analysis of Chinese language data (Fan 2021).

Schick & Schütze effectively directed pre-trained language models in producing sentiment analysis data using prompt-based learning, enhancing the precision of models with a limited number of samples. Their research demonstrated how strategic prompt engineering could guide language models to generate accurate sentiment labels, even with scarce training data (Schick & Schütze 2021).

Gao examined the effects of few-shot sentiment analysis with carefully crafted prompts, showing the effectiveness of pre-trained language models with minimal data. This study highlighted the potential of few-shot learning techniques to achieve high-performance sentiment classification with significantly reduced data requirements (Gao 2021).

Hansen concentrated on advancing self-supervised learning in reinforcement learning by utilizing data enhancements such as arbitrary cropping and color variability to improve the efficiency and applicability of samples in gaming simulations. This approach improved the generalization and robustness of reinforcement learning models by incorporating diverse visual data augmentations (Hansen 2021).

Gu suggested an immediate data enhancement technique involving BERT, dynamically producing training samples for sentiment assessment to enhance model resilience and function. By generating on-the-fly augmented data, this method significantly improved the adaptability and performance of sentiment analysis models (Gu 2022).

Li employed a blend of GPT-1, GPT-3, and human-in-the-loop methods to produce superior augmented data, markedly improving the effectiveness of sentiment analysis models. This hybrid approach combined the strengths of various generative models and human expertise, resulting in high-quality, diverse training datasets that boosted model accuracy (Li 2022).

Zhang developed an innovative data enhancement model through contrastive learning and prompts, enhancing sentiment analysis by producing varied and contextually pertinent training sets. This method leveraged the principles of contrastive learning to generate rich, relevant data augmentations, improving the model's ability to distinguish subtle sentiment differences (Zhang 2022).

Chen et al. introduced a brief, prompt-oriented learning method for analyzing sentiments, demonstrating how thoughtfully crafted prompts can greatly enhance model precision even with restricted data (Chen et al. 2023).

Lee used a blend of conventional augmentation techniques and GPT-3-oriented prompt strategies to enhance performance in sentiment sorting operations. This combined approach leveraged traditional data augmentation with advanced prompt-based generation, resulting in significant improvements in classification accuracy (Lee 2023).

Zhao introduced a method in automated prompt engineering, utilizing reinforcement learning to refine prompt choices for data enhancement, thereby improving the precision of sentiment analyses. This innovative technique employed reinforcement learning to optimize prompt selection, leading to more effective data augmentation and enhanced sentiment model performance (Zhao 2023).

The development of text sentiment classification has evolved from early rule-based methods and manual labelling to advanced machine learning and deep learning techniques. Initial efforts involved studying the semantic orientation of adjectives and subjectivity recognition. The introduction of machine learning algorithms marked significant progress, with SVM models showing high performance in sentiment analysis tasks. Unsupervised learning and semantic classification techniques followed, improving classification accuracy and applicability. The advent of deep learning brought convolutional and recurrent neural networks into the field, enhancing feature extraction and classification capabilities. Transformer models revolutionized sentiment analysis by efficiently handling temporal issues and improving computational efficiency. Recent advancements include data augmentation techniques such as back-translation, GANs, and pre-trained language models like GPT-2, GPT-3, and BERT, significantly enhancing model performance. Prompt-based learning and few-shot techniques have further improved accuracy with minimal data, demonstrating the continuous evolution and impact of deep learning on sentiment analysis. Recent studies, particularly those focusing on prompt-based data augmentation, have shown promising results in generating diverse and high-quality training samples, leading to improved robustness and performance of sentiment analysis models.

These related works show that prompt-based data augmentation shows good results in sentiment analysis tasks in natural language processing, especially in scenarios with limited data. However, due to different language models and different prompts, the effect of prompt-based data augmentation fluctuates. Based on these findings, this study decided to use GPT-3.5-turbo, the public pre-trained language model, to handle the sentiment analysis challenges in the field of game reviews.

CHAPTER III

METHODOLOGY

3.1 INTRODUCTION

This chapter introduces the data preprocessing process for game review data and provides a detailed explanation of prompt-based data augmentation. Additionally, the chapter presents the research design and some hyperparameter adjustments for two neural network methods, CNN and LSTM, as well as the BERT pre-trained model in handling sentiment analysis of game review data. Evaluation metrics for assessing algorithm performance are also described. Finally, a flowchart of the experimental design is presented.

The study comprises six stages. The first stage involves preparing the game review data dataset. The second stage is data preprocessing, focusing on data cleaning and randomly selecting a subset of data to be used for training. The proportion of data used for training is 80%, and the proportion of data set used for testing is 20%. The third stage involves prompt-based data augmentation of the training data subset, preparing it for downstream sentiment analysis tasks. The fourth stage is the experimental setup in the AutoDL cloud environment. The fifth stage involves testing various hyperparameters of three models on different training sets. The sixth stage compares the classification performance on different training sets after prompt-based data augmentation.

As previously mentioned, the experimental objectives are divided into two main parts. First, to identify high-quality augmented data for prompt-based data augmentation. Then, to evaluate the effectiveness of the prompts and prompt-based data augmentation by using CNN, LSTM neural network methods, and the BERT pre-trained model for downstream sentiment analysis tasks. By comparing the results of prompt-based data augmentation on the same dataset using multiple deep neural networks, this study can gain insights into which neural network achieves higher accuracy with prompt-based data augmentation and which neural network is more resilient to prompt-based data augmentation.

The second part involves the sentiment analysis task, introducing the deep learning text sentiment analysis methods used in this study. These methods combine different types of deep neural networks with various prompt-based data augmentations to achieve the highest accuracy. The text in the dataset is converted into tensor forms that can be input into the networks. Then, the CNN-based text classification method and the LSTM-based text classification method are introduced and improved upon.

3.2 DATA AUGMENTATION

Employing GPT for Prompt-based Data Augmentation capitalizes on the use of pre-established language models like GPT-2 or GPT-3 to produce varied data samples, thereby enriching training data for activities such as sentiment analysis. This approach enhances the efficiency of the model by creating targeted instructions that direct the previously trained language model to produce data samples with specific attributes. The stages of this technique are outlined below:

1. Choose a suitable GPT model tailored to the task's necessities, like GPT-2 or GPT-3. Selecting a model may hinge on the task's intricacy and the requisite quantity of data. Subsequently, develop cues that will inform GPT in compiling targeted sentiment data. For example, creating sentences reflecting positive emotions, a prompt such as "Generate a sentence with positive sentiment" might be employed. Ensuring clarity in the prompts is crucial to guarantee that the generated data fulfills the anticipated attributes.

2. Add the crafted prompts into the GPT model and produce a variety of answers. Enhancing the variety of data can involve obtaining diverse samples through either modifying the prompts or repeatedly generating them. As an illustration, the phrase "Generate a sentence with positive sentiment" could provoke reactions such as, "Today is a wonderful day, I feel very happy." This study employed prompts such

as "Substitute adjectives in a sentence with words of inverse meaning" and "Revamp this sentence."

3. Categorize and confirm the produced data for its accuracy and pertinence. The task might be executed manually or through alternate models to automatically evaluate if the produced data aligns with the anticipated sentiment classification. Exclude clearly erroneous or inaccurate samples and keep only the finest data for training purposes. Incorporating superior samples into the initial training dataset creates an expanded training set, guaranteeing that the quality of the augmented data and volume of the created data considerably enhance the training dataset's diversity and breadth.

4. Reeducate the sentiment analysis model through the enhanced training dataset to assess its efficacy on both validation and testing datasets. Assess the model's efficacy pre and post data augmentation to confirm their success. The approach can considerably enhance the variety and magnitude of the training data, thus boosting the sentiment analysis model's efficiency and its ability to generalize.

This study utilizes GPT-3.5-turbo. The aim of the research is to improve the semantic issues in review data through synonym rewriting using GPT. By antonym rewriting, the study aims to add the generated data back to the original dataset, thereby addressing the data imbalance problem within the dataset.

3.3 DATA PRE-PROCESSING

Data preprocessing is a very critical step when using deep learning methods for text categorization tasks. As text data has various forms and expressions, including case, punctuation, special characters, etc. And some of the data contain noise and invalid information, such as garbage characters, stop words, etc. Data preprocessing can be carried out to normalize the text, transform the text into a uniform format, remove unnecessary characters or punctuation marks_and eliminate case differences. This reduces noise and unnecessary complexity, thus reducing interference with the model and improving its accuracy and generalization. This enables the model to better understand the content of the text. The flow of data preprocessing is shown in Figure

3.1.

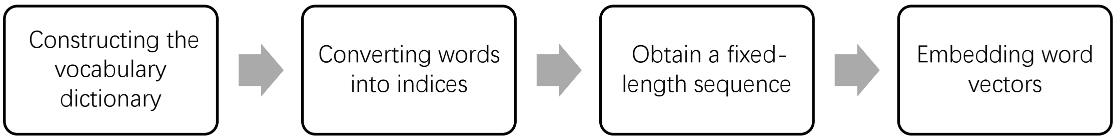

Constructing the vocabulary dictionary
Converting words into indices
Obtain a fixed-length sequence
Embedding word vectors

Figure 3.1 Flow chart of data preprocessing

1.Constructing the vocabulary dictionary

Before performing the sentiment analysis task, it is first necessary to construct a dictionary of vocabulary so that the words in the text, are transformed into their indexes in this dictionary. In order to construct this dictionary, it is first necessary to read all the training set text and segment each line of text by word and remove useless information such as deactivated words, case characters, etc., so as to reduce noise and unnecessary complexity. Then, the frequency of occurrence of all characters is counted and sorted from highest to lowest frequency. In order to control the size of the dictionary, the first 10,000 characters with the highest frequency of occurrence are retained. These characters are then sorted in descending order of frequency and assigned a unique index number as the dictionary value for subsequent encoding processes.

2.Converting words into indices

All text sentences in the training collection may be interpreted as a series of words. In the initial stages of data processing, utilizing an established vocabulary dictionary, every word can be converted into a respective index. Thus, every phrase within a text can be depicted using a collection of numerical indices, which are then fed into the deep learning system. Converting words into indexes alone suffices to convert text data into a numerical format for algorithmic processing by the model. Aligning with numerical representation maintains the sequence of words in text sentences and minimizes data complexity, thus enabling the model to more effectively process and comprehend the text's semantic and syntactical characteristics.

3.Obtain a fixed-length sequence

Given the varying lengths of text sentences, their integration into the deep learning model necessitates initial conversion into sequences of identical lengths. The mechanism identifies a typical length (for instance, 64), known as the pad_size. Should the sentence's length surpass the pad_size, the segment containing a word longer than the pad_size is shortened. Should the length of the statement fall below pad_size, it's expanded to the designated length using a specified number (like 0). Subsequent to this phase, each input text becomes identical in length, thereby simplifying the manipulation by the input network.

4.Embedding word vectors

Embedding is a common and important step in text classification. Embedding is the process of mapping words or characters to a low dimensional continuous vector space. Word embedding converts words or characters from discrete symbolic forms to continuous vector representations to capture the semantic information of words or characters. Similar words or characters are closer in the embedding space, which enables the model to better understand and infer the semantic relationships of the text. In addition, word embedding reduces the feature dimension; The vocabulary in text data is usually large, and using unique hot encoding or sparse representation can lead to high-dimensional sparse input features, which increases the complexity and computational cost of the model. Embedding can reduce the dimensionality of data by mapping words or characters to a low dimensional continuous vector space, thereby reducing the number of parameters and computational complexity of the model. Word embedding can encode the contextual information of words or characters into vectors. This means that the embedding vector of a word or character not only represents its own meaning, but also contains information about the surrounding words or characters. This helps the model better understand and capture the grammar and context in the text. Due to the large size of the dataset for this task, this task uses randomly initialized embedding matrices and continuously learns the word embedding matrix parameters during the training process, thus making the obtained word embedding matrices more suitable for this task.

3.4 CNN-BASED CLASSIFICATION METHOD FOR GAME REVIEWS

Text CNN (Text Convolutional Neural Network) is a convolutional neural network model for text classification and text representation learning. Its network structure is shown in Figure 3.2. It extracts local features in text by applying a 2-dimensional convolutional operation and uses a fully connected network to classify these features for prediction. The structure of Text CNN is similar to that of traditional convolutional neural networks, but with some adjustments in the input to adapt to text data (Alzubaidi et al. 2021). Typically, the input to a Text CNN is a fixed-length text sequence where each word is represented by a vector. Therefore, before feeding the text into the network, it needs to be word-embedded to transform it into a vector representation. Moreover, these vectors are pre-trainable and can be learned during the training process of the network.

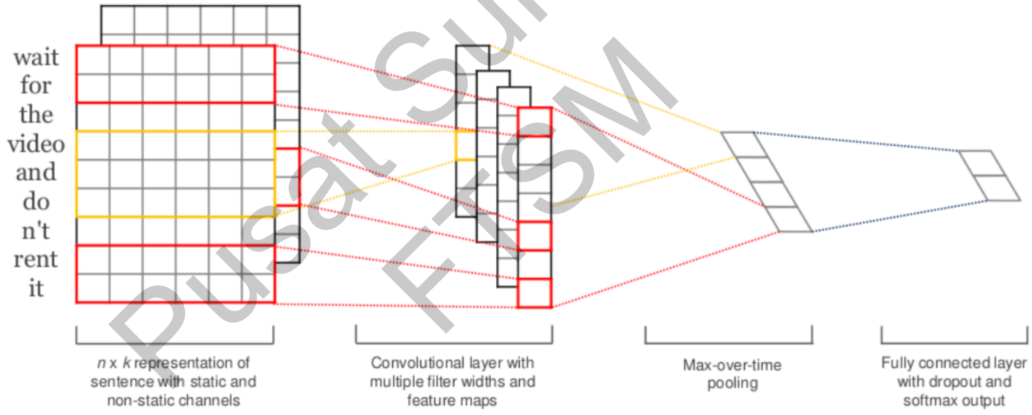

wait
for
the
video
and
do
n't
rent
it
n x k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure 3.2 Text CNN network structure

The core idea of Text CNN is to capture the local information in the text through convolutional operations. It uses multiple convolutional kernels of different sizes to perform convolutional operations on the input text and nonlinear mapping through activation functions. These convolutional kernels slide over different window sizes so that local features within different scales can be captured, and the result of the convolutional operation is a set of feature mappings, each corresponding to a convolutional kernel. In order to reduce the dimensionality of the feature mappings, they are usually subjected to a maximum pooling operation, which selects the most salient features in each feature mapping, and ultimately splices the features at different scales to obtain the features of the input text (2017). In this study, three 2D convolutions of sizes 3×1, 4×1, and 5×1 are used to extract multi-scale local features, and the number of output channels of each convolution is 100. In addition, a fully connected layer and a softmax layer are connected after the convolution and pooling layers for classifying and predicting the input features. During training, the model learns a feature representation suitable for the sentiment analysis task by means of a backpropagation algorithm.

Since Text CNN uses 2-dimensional convolution to learn text sequence features, and 2-dimensional convolution can capture local features in the text, it is suitable for short sentiment analysis tasks with the advantages of simple structure and high computational efficiency. However, Text CNN also has some limitations, for example, it ignores a large range of contextual information in the text and cannot capture long-distance dependencies. Therefore, for some complex natural language processing tasks, such as longer text categorization, named entity recognition and syntactic analysis tasks, Text CNN may not be as effective as using RNN or models based on attention mechanisms. Overall, Text CNN is a simple but effective model for many text categorization tasks and has achieved good performance in practice.

3.5 LSTM-BASED CLASSIFICATION METHOD FOR GAME REVIEWS

LSTM solves the gradient vanishing and gradient explosion problems in traditional RNNs by introducing memory cells and gating mechanisms. The memory unit allows the LSTM to retain important contextual information when processing long sequences, while the gating mechanisms (input gate, forgetting gate, and output gate) determine how to input, forget, and output information (Landi et al. 2021). In LSTM based text categorization methods, text data is first segmented into word level sequences. Each word or character is vectorized and embedded into a low-dimensional vector representation that serves as the input to the LSTM. The LSTM network is able to process the input sequences step-by-step and update its internal state at each time step. This enables LSTM to model sequences of textual data, capturing the contextual relationships between words or characters. Compared to traditional bag-of-words or ngram models, LSTM-based methods can better take into account the order of words and the structure of text.

Once the entire input sequence has been processed, there is an option to use the LSTM hidden state from the last time step as a representation of the entire text, or to summarize it using the hidden states from all time steps. This representation will be passed to the part connecting the fully connected layer to the softmax layer for making classification predictions. The improvement of LSTM in this study consists of the following parts:

1.Use of multilayer LSTM structures

In this study, this study uses a multilayer LSTM structure, i.e., stacking multiple LSTM layers and using the output of one LSTM layer as the input of the next layer to form a deep network structure, which enhances the model's representational capability, learns more complex features and abstract representations, and improves the performance of text categorization.

2. Using a bidirectional LSTM structure

In traditional unidirectional LSTMs, each time step can only utilize past information to make predictions. Bi-LSTM, or bi-directional LSTM, feeds input sequences into two separate LSTM layers: one layer processes the input sequences in the normal order (left to right), and the other processes the input sequences in the reverse order (right to left). In this way, each LSTM layer has its own hidden state and memory unit that captures the forward and reverse contextual information, respectively. At each time step, the forward and reverse hidden states are spliced together to form the final bidirectional hidden state. This bidirectional hidden state can be used as a representation of the entire sequence for subsequent sentiment analysis tasks. (Abid, Li & Alam 2020) Finally, the final classification prediction is achieved using the fully connected and softmax layers. The output of the bidirectional LSTM can provide rich information to help the model better understand the semantic and contextual relationships in the text, thus improving the classification performance.

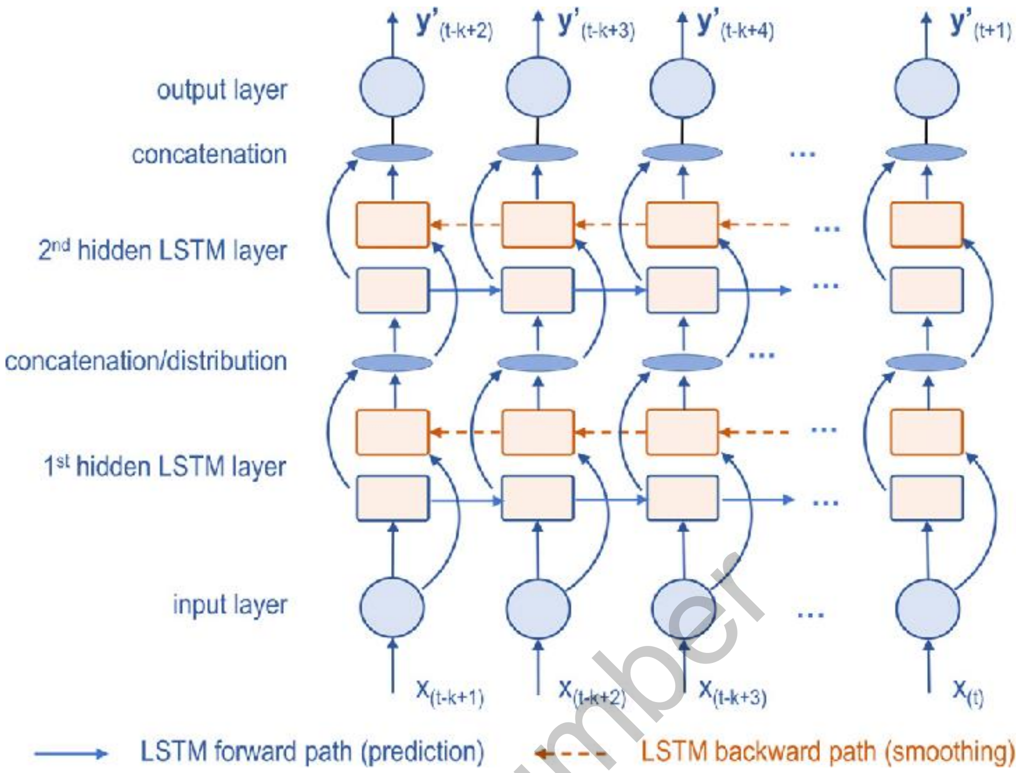


Figure 3.3 2-layer Bi-LSTM network structure

3.Use of attention mechanisms

The attention mechanism allows the model to dynamically assign different weights to different parts of the input sequence based on their content, thus focusing more purposefully on important information (Li et al. 2020). It improves the model's ability to focus on key parts of the input sequence and enhances the model's expressiveness and performance. In this study, this study use the following method to combine LSTM with the attention mechanism, so as to learn the semantic relationships between the output states of LSTM at different time steps and weight and sum these states, thus improving the degree of the model's attention to the key parts in the input sequence, and thus enhancing the model's classification performance. The process is shown below.

First, the input sequence is modeled using an LSTM network as an encoder. The input sequence goes through each time step of the LSTM to obtain a series of hidden states. Subsequently, the computation of attention weights is performed, specifically, for each hidden state, the attention weight associated with it is computed. In this study, this study uses additive attention in the attention mechanism. Specifically, for each hidden state h_t , the attention weight α_t can be computed as shown in Equation (3.1):

$$\alpha_t = \text{softmax}(W_a \cdot \tanh(W_h \cdot h_t + W_s \cdot s)) \tag{3.1}$$

where W_a , W_h and W_s are learnable weight matrices and s is a vector representing the whole sequence.

Then a weighted sum operation is performed, which weights and sums the hidden states using the attention weights to obtain the attention vector c , which is a weighted representation of the different parts of the input sequence. The formula for the attention vector c is shown in equation (3.2):

$$c = \sum \alpha_t \cdot h_t \tag{3.2}$$

Finally, the attention vector c is spliced with the hidden state h_t of the current time step to form the context vector u_t : as shown in Equation (3.3):

$$u_t = [h_t, c] \tag{3.3}$$

The context vector contains all the feature information of the input text, in order to realize the classification prediction, only need to input the context vector u_t into the subsequent fully connected layer and softmax layer, and then this study can get the prediction results for text classification.

3.6 BERT-BASED CLASSIFICATION METHOD FOR GAME REVIEWS

Two steps, pre-training and fine-tuning, are required for text sentiment analysis using BERT. In the pre-training phase, BERT is pre-trained using large-scale unlabeled text data to obtain word and sentence representations by learning contextual information.The BERT model is able to better capture the semantic associations of words by using a bi-directional Transformer encoder, which is able to take into account both the left and the right side of the contextual information. This enables BERT to learn richer lexical and syntactic features. In the fine-tuning phase, the BERT model applies its pre-trained representation to specific text categorization tasks by performing supervised fine-tuning on specific tasks. For text categorization, it is often necessary to add an additional fully-connected layer as a classifier on top of the BERT model, which converts the output of BERT into the desired classification labels. During the fine-tuning process, the parameters of the classification layer are trained along with those of BERT in order to apply the BERT representation capabilities to specific sentiment analysis tasks.

In this study, this study use a pre-trained BERT model, based on which the pre-trained BERT is fine-tuned with the game review dataset, and the game review texts are classified by a fully connected layer.The fine-tuning process of BERT includes data preprocessing and model construction. These two parts of the task are briefly described below, respectively.

1. Data preprocessing

For each input text, it needs to be processed according to BERT's input representation. First, the text needs to be divided into sentence A (i.e., the game review) and sentence B (the blank text), and the "[SEP]" markers are added between the sentences to represent the sentence boundaries. Then, "[CLS]" markers are added at the beginning of the sequence to represent the representation of the entire input sequence. Each word in the input sequence is converted into a corresponding word vector.BERT uses WordPiece or other similar word-splitting methods to divide words into subwords, and then maps these subwords into corresponding word vectors.

2. Modeling

Construct a classification model based on BERT. In this study, this study use a fully connected layer as an additional classification layer to transform the sequence features output from BERT into classification labels. In addition, this study uses a random node deactivation (Dropout) strategy to prevent overfitting and improve the generalization ability of the model.

3.7 LOSS FUNCTION

In this study, game review classification is considered as a categorization task with the number of categorization categories as 2, i.e., recommended or not recommended, and the gap between the model output probability distribution and the true label is measured using the Cross Entropy (BCE) loss function, which is defined as shown in Eqn. (3.4) for the binary categorization task. Where, $\hat{y}$ and y denote the prediction and label respectively.

$$L(y,\hat{y}) = -y\log(\hat{y}) + (1-y)\log(1-\hat{y}) \tag{3.4}$$

By minimizing the cross-entropy loss function, the model can be made to improve the classification performance on the training data, making the model's predictions closer to the real labels. During the training process, optimization algorithms such as gradient descent are usually used to update the parameters of the model to minimize the cross-entropy loss function.

CNN is a classical deep learning model, which is used to process data with hierarchical structure, such as images, audio and video, etc. CNN usually consists of convolutional layer, pooling layer, fully connected layer, etc. The pooling layer is used to down sample the input features to increase the receptive field and reduce the channel dimension of the features to reduce the number of parameters for the subsequent network layer processing. The pooling layer is used to down sample the input features, increase the receptive field and reduce the channel dimension of the features, which reduces the number of parameters for the subsequent network layer processing; the fully connected layer performs a linear transformation of the input features, which has the ability of global feature sensing, in the sentiment analysis task, the output dimension of the fully connected layer is consistent with the number of classification categories, and the output of each neuron is the probability that the text belongs to a certain category;

the convolutional layer is the core of the CNN that is used to extract features from the input raw data. In the convolution process, the center position of the convolution kernel is taken as the benchmark, and each convolution only operates on the local pixels about the input data, and by adjusting the position of the convolution kernel, the convolution operation is gradually applied to all the positions of the input data, so as to obtain the complete feature map of the input data. The convolution operation can effectively extract the local features of the input data, and by increasing the number of convolution kernels, the convolution layer can extract multi-channel features at the same time and combine them into a higher-level feature map.

3.8 EVALUATION METRICS

In sentiment sentiment analysis tasks, evaluating the performance of a model is crucial. These metrics help quantify various aspects of the model's performance, such as accuracy, precision, recall, and more. Below are some commonly used evaluation metrics and their explanations:

To quantitatively assess the performance of the methods in this study, the following evaluation metrics are used.

1.Accuracy:

Accuracy rate is one of the most common classification evaluation metrics in classification problems, which indicates the ratio of the number of samples correctly predicted by the model to the total number of samples. The formula for calculating the accuracy rate is shown in equation (4.1):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

Where TP (True Positive) denotes the number of true examples, TN (True Negative) denotes the number of true negative examples, FP (False Positive) denotes the number of false positive examples, and FN (False Negative) denotes the number of false negative examples.

Accuracy measures the overall accuracy of the model predictions but can be misleading in the presence of category imbalance. When categories are unevenly distributed, accuracy may be higher than actual model performance.

2. F1-score

The F1-score, a metric that merges precision and recall efficiency, is frequently employed in the assessment of sentiment analysis task performance. The computation of the F1 score involves the adjusted mean of Precision and Recall. Precision is defined as the proportion between samples the model predicts as positive versus those actually identified as such, essentially the ratio of samples accurately forecasted by the model to the total samples estimated as positive. Equation (4.2) displays how to determine accuracy. TP (true positive) indicates the count of genuine cases, while FP (false positive) denotes the tally of falsely positive instances of sexually transmitted infections. Accuracy evaluates the precision of a model' s forecasts as a definitive instance, namely, the degree of accuracy the model' s predictions possess. An elevated rate of accuracy suggests that the model possesses a reduced tendency to incorrectly predict samples in their positive form.

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

The recall rate represents the proportion of samples that are actually positive cases that are predicted by the model, i.e., the number of samples that are correctly predicted by the model to be positive cases as a proportion of the total number of samples that are actually positive cases. The recall rate is calculated as shown in Equation (4.3). Where TP (True Positive) denotes the number of true cases and FN (False Negative) denotes the number of false negative cases. Recall measures the model's ability to recognize positive example samples, i.e., how much of the model is able to correctly find samples that are actually positive examples. A higher recall indicates that the model is better able to capture positive example samples.

$$Recall \ = \ \frac{TP}{TP \ + \ FN} \tag{4.3}$$

Recall rate signifies the ratio of samples accurately identified as positive cases by the model to those that are in fact positive, essentially the total samples predicted as positive cases. The calculation of the recall rate is outlined in Equation (4.3). Where TP (True Positive) represents the count of true instances, and FN (False Negative) refers to the tally of false negatives. The recall metric assesses the model's proficiency in identifying positive sample samples, meaning the extent to which it accurately recognizes those that are indeed positive examples. Enhanced recall is a sign that the model has a superior capacity to identify accurate sample samples.

$$F1-score \ = \ 2\times\frac{Precision\times Recall}{Precision \ + \ Recall} \tag{4.4}$$

CHAPTER IV

RESULTS AND DISCUSSION

4.1 INTRODUCTION

This chapter provides a detailed description of the experimental details and results. In terms of experimental details, it first introduces the dataset used in the experiments, followed by the experimental setup, including the configuration of the experimental platform, API settings, prompt settings, and optimizer parameters. Regarding the experimental results, an ablation study is conducted on the three methods proposed in this study to explore the optimal hyperparameter configuration. Then, a quantitative comparison of the three proposed methods is presented. Finally, the parameter count and running speed of sentiment analysis after different data augmentations are compared, and their limitations are summarized. The contents of this chapter include: 4.1 Dataset, 4.2 Experimental Setup, 4.3 Prompt-based Data Augmentation, 4.4 Ablation Study, and 4.5 Chapter Summary.

4.2 DATASET

The performance of deep learning methods is typically closely related to the quality and quantity of the training data, thus requiring the collection of large-scale, high-quality, and representative data for model training and testing. The dataset used in this study is sourced from Steam, where many players write reviews on game pages and can choose whether to recommend the game to others. This study analyzes user reviews to determine whether users are willing to recommend the game. As shown in Figure 4.1, the dataset contains 6,144,899 valid reviews with an imbalanced distribution of categories (recommend and not recommend). The number of recommendations far exceeds the number of non-recommendations, with 1,156,686 non-recommendations and 4,988,213 recommendations. It is important to note that the classification of review data is based on whether the user recommends the game, not the actual sentiment of the review. Due to the large dataset size, 20,000 reviews are extracted for the experiments. Additionally, a smaller model is more conducive to validating the effect of data augmentation and evaluating model performance. The extracted dataset includes 20,000 valid reviews, with 2,879 non-recommendations and 17,121 recommendations.

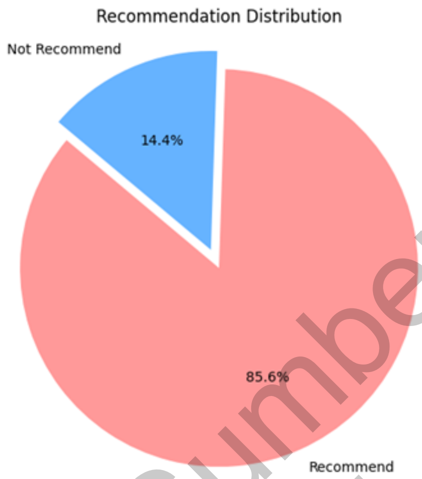


Figure 4.1 Agree ratio chart

The dataset includes five attributes: app_id (Game ID), Game Name, Review text, Review Sentiment (whether the review recommends the game or not), and Review vote (whether the review was recommended by another user or not). The primary focus is on the Review text and Review Sentiment. this study uses Review Sentiment as this study sentiment classification standard. this study considers a review to be positive if the user recommends the game, even if the review includes some criticism and suggestions for improvement.