

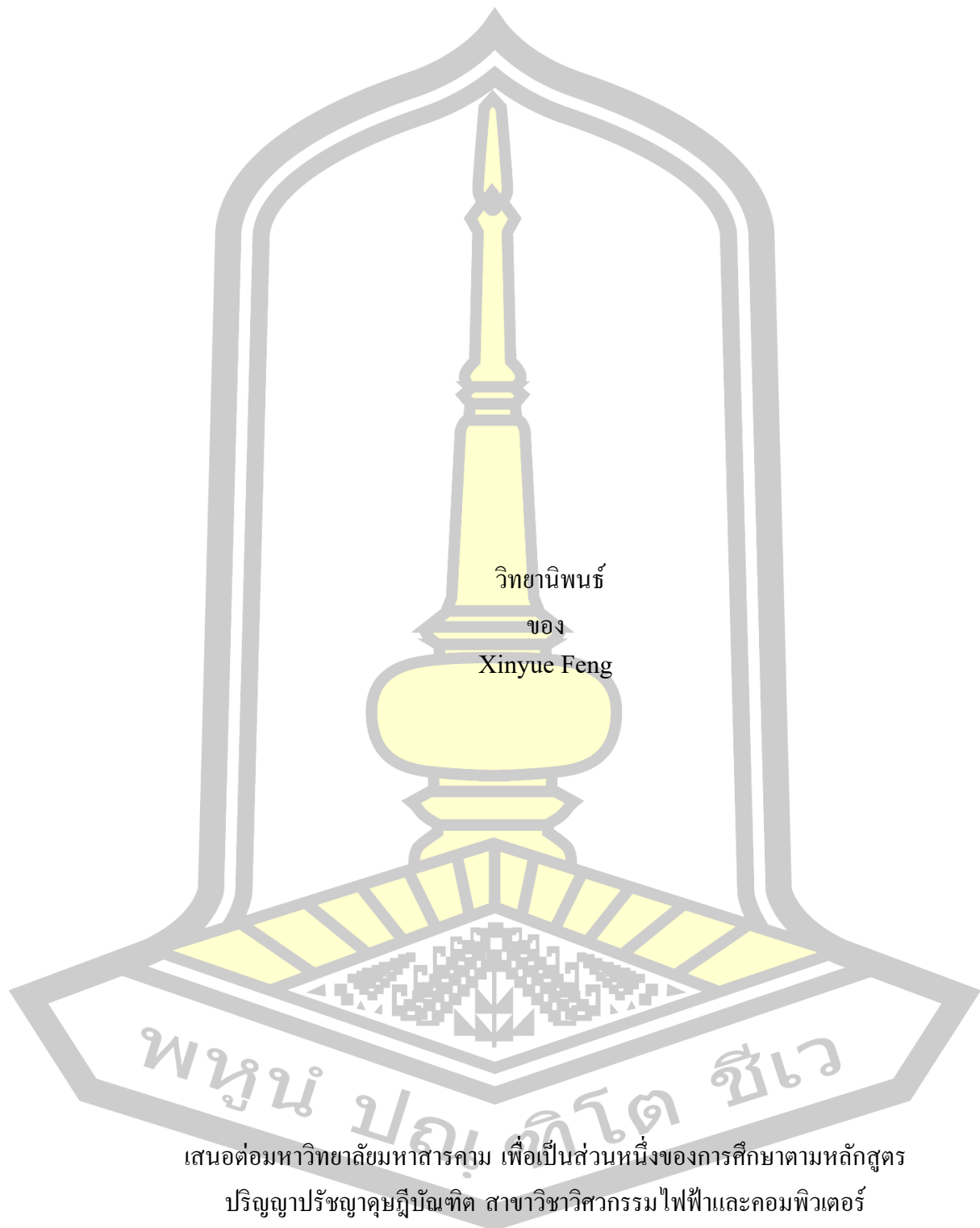
Research on Sentiment Analysis of Microblog Text based on Web Crawler

Xinyue Feng

A Thesis Submitted in Partial Fulfillment of Requirements for
degree of Doctor of Philosophy in Electrical and Computer Engineering
June 2025

Copyright of Mahasarakham University

Research on Sentiment Analysis of Microblog Text based on Web Crawler



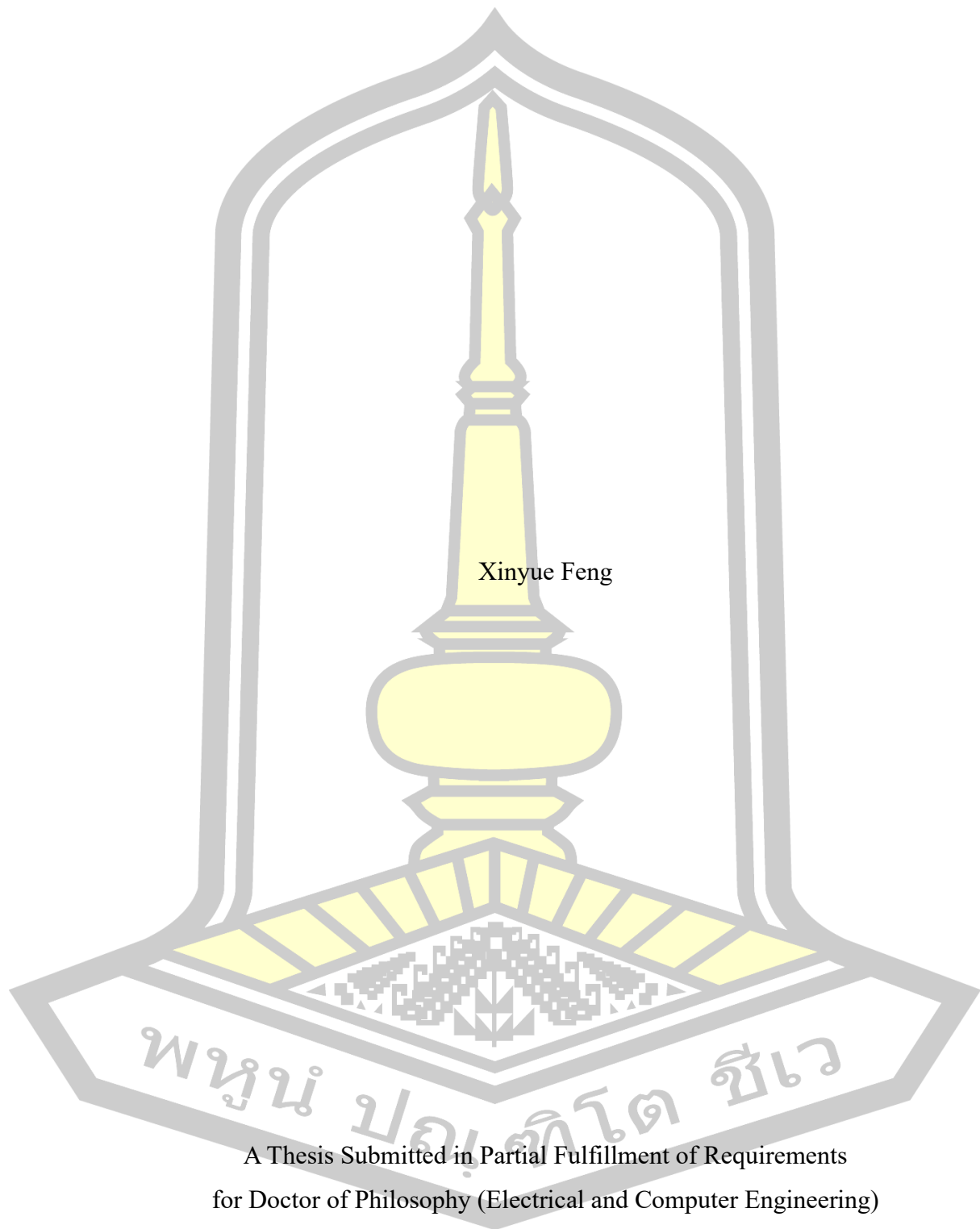
วิทยานิพนธ์
ของ
Xinyue Feng

เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร
ปริญญาปรัชญาดุษฎีบัณฑิต สาขาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

มิถุนายน 2568

ลิขสิทธิ์เป็นของมหาวิทยาลัยมหาสารคาม

Research on Sentiment Analysis of Microblog Text based on Web Crawler



Xinyue Feng

A Thesis Submitted in Partial Fulfillment of Requirements
for Doctor of Philosophy (Electrical and Computer Engineering)

June 2025

Copyright of Mahasarakham University



The examining committee has unanimously approved this Thesis, submitted by Ms. Xinyue Feng , as a partial fulfillment of the requirements for the Doctor of Philosophy Electrical and Computer Engineering at Mahasarakham University

Examining Committee

Chairman

(Asst. Prof. Adirek Jantakun , Ph.D.)

Advisor

(Assoc. Prof. Niwat Angkawisittpan
, Ph.D.)

Committee

(Prof. Worawat Sa-Ngiamvibool ,
Ph.D.)

Committee

(Assoc. Prof. Chonlatee Photong ,
Ph.D.)

Committee

(Taweesak Thongsan , Ph.D.)

Mahasarakham University has granted approval to accept this Thesis as a partial fulfillment of the requirements for the Doctor of Philosophy Electrical and Computer Engineering

(Assoc. Prof. Juckamas Laohavanich ,
Ph.D.)

Dean of The Faculty of Engineering

(Asst. Prof. Pondej Chaowarat , Ph.D.)
Dean of Graduate School

TITLE	Research on Sentiment Analysis of Microblog Text based on Web Crawler		
AUTHOR	Xinyue Feng		
ADVISORS	Associate Professor Niwat Angkawisittpan , Ph.D.		
DEGREE	Doctor of Philosophy	MAJOR	Electrical and Computer Engineering
UNIVERSITY	Maharakham University	YEAR	2025

ABSTRACT

Sentiment analysis of Microblog text has become an important research direction in the field of natural language processing (NLP), thanks to the rapid development of social media platforms such as Microblog. However, the unique characteristics of Microblog data—such as short text length, informal language, the inclusion of emoticons, and dynamic changes in sentiment over time—pose significant challenges to traditional sentiment analysis methods. This paper proposes a comprehensive Microblog sentiment analysis framework that combines improved web scraping techniques, advanced data preprocessing methods, and deep learning-based sentiment classification models to enhance the accuracy and robustness of the analysis.

Microblog data collection: We explore the challenges of Microblog web scraping, where anti-scraping mechanisms limit large-scale data collection. We propose an optimised scraping strategy to efficiently collect text data and overcome access restrictions.

Microblog data preprocessing: Due to noise in raw Microblog data, we employ regular expressions, Jieba word segmentation, and stopwords removal techniques to extract clean, sentiment-relevant text for analysis.

Deep learning-based sentiment analysis: We introduced a CNN-BiLSTM model with an attention mechanism, which can capture local features through CNN, process long-term dependencies through BiLSTM, and assign weights to key sentiment clues through the attention mechanism. Experiments on real Microblog datasets show that our model outperforms baseline methods in classification accuracy.

This study not only advances innovations in sentiment analysis methods but also provides practical insights for applications such as public opinion monitoring and social media analysis. Future research could explore multimodal sentiment analysis (integrating text and emojis) and domain adaptability across different Microblog scenarios.

Keyword : Microblog, Sentiment Analysis, Web Scraping, Data Preprocessing, CNN, BiLSTM, Attention Mechanism

ACKNOWLEDGEMENTS

Time flies, and before I know it, my doctoral studies at Mahasarakham University in Thailand are coming to an end. Looking back on this unforgettable journey, my heart is filled with gratitude.

First of all, I would like to express my sincere gratitude to my supervisor, Professor Nivart. Thank you for guiding me into the field of academic research with your extensive knowledge and rigorous academic attitude. You have always patiently answered every question I have asked, and I have benefited greatly from your guidance. In addition, I would like to thank all the professors on the thesis defence committee for enabling me to successfully complete this research.

I would like to thank Mahasarakham University for providing an excellent learning environment and research platform. Here, I not only gained knowledge but also made many good friends and mentors. I will always cherish the days spent discussing academic issues with Thai classmates, and I am grateful to my Chinese peers for their mutual support in a foreign land.

I sincerely thank my parents and family for their selfless love and support, which allowed me to pursue my studies with peace of mind. Special thanks to my beloved partner; your understanding and companionship have been the greatest motivation for me to persevere.

Finally, I thank all the teachers and friends who have helped me. This academic journey in Thailand will undoubtedly become the most precious treasure of my life.

Xinyue Feng

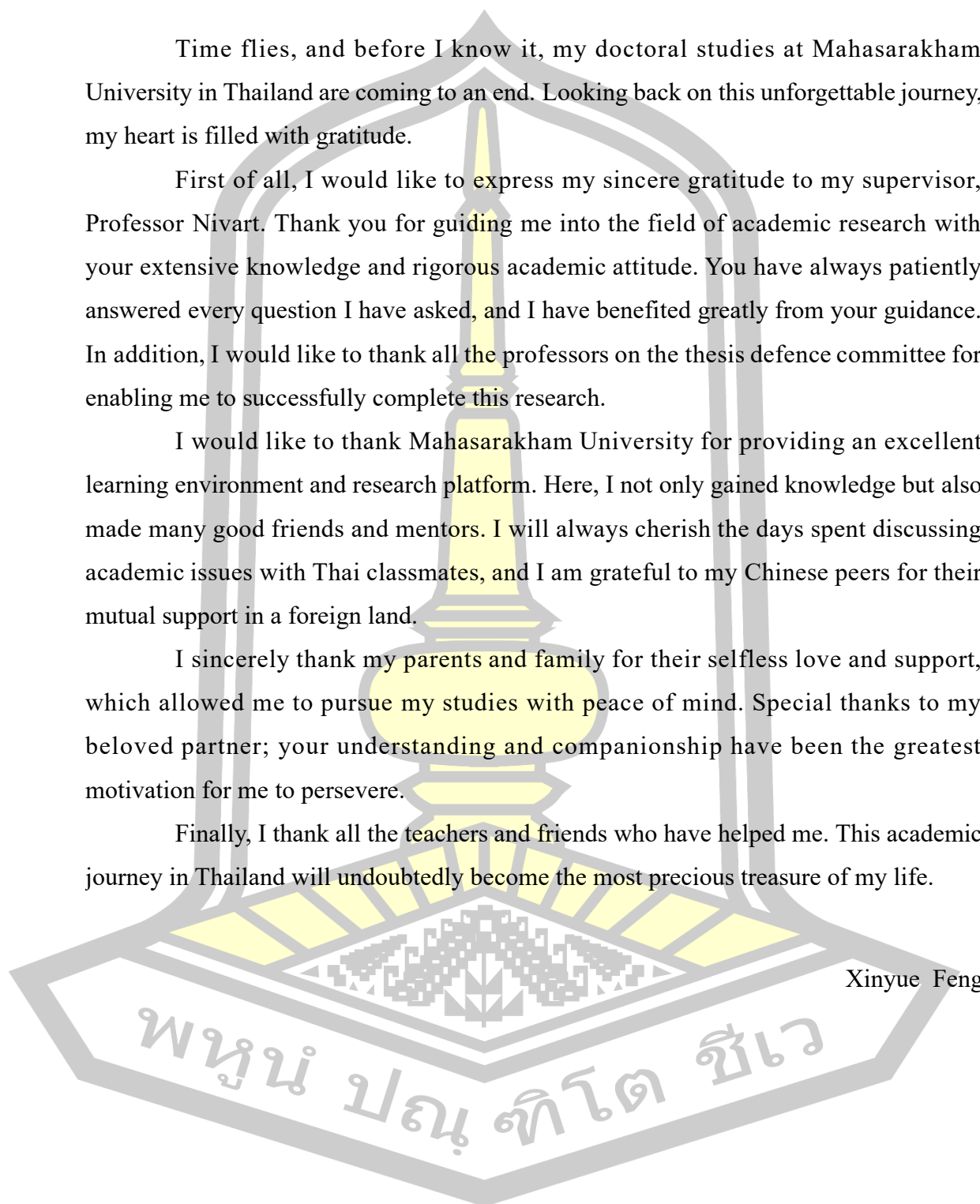


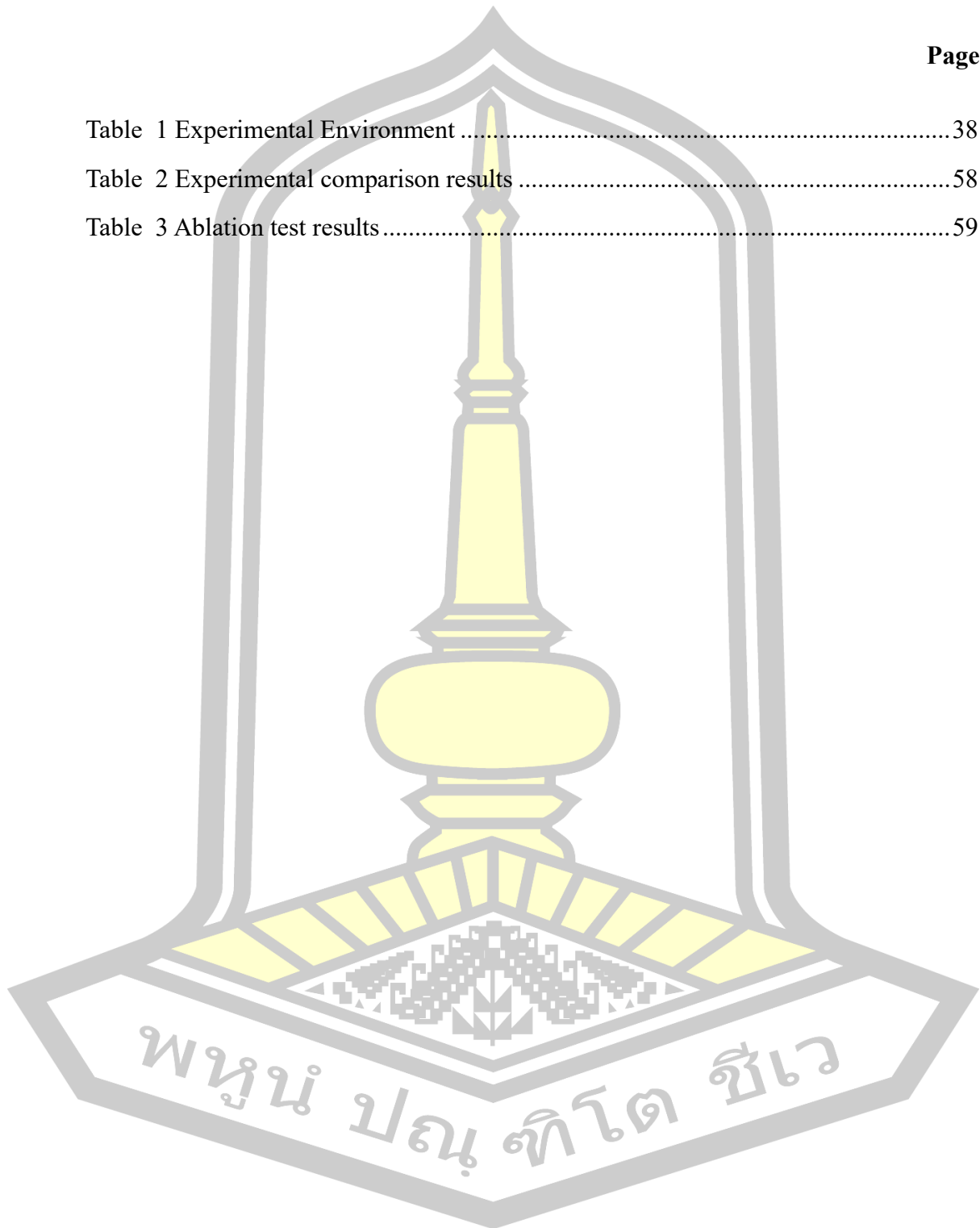
TABLE OF CONTENTS

	Page
ABSTRACT.....	D
ACKNOWLEDGEMENTS.....	E
TABLE OF CONTENTS.....	F
LIST OF TABLES.....	H
LIST OF FIGURES.....	I
Chapter 1 Introduction.....	1
1.1 Research Background.....	1
1.2 Research Objectives.....	5
1.3 Benefits of the Research.....	6
Chapter 2 Literature Reviews.....	8
2.1 Current Status of Data Acquisition Research.....	8
2.2 State of the Art of Sentiment Analysis Research.....	14
2.2.1 Sentiment Classification based on Sentiment Dictionary.....	15
2.2.2 Sentiment Classification based on Machine Learning.....	19
2.2.3 Sentiment Classification based on Deep Learning.....	21
Chapter 3 Methodology.....	25
3.1 Data Acquisition Technology.....	25
3.1.1 API-based Data Acquisition.....	25
3.1.2 Web Crawler Based Data Acquisition.....	27
3.2 Data Processing Technology.....	31
3.2.1 Data Cleaning.....	31
3.2.2 Text Segmentation.....	32
3.2.3 Remove Stop Words.....	33
3.3 Text emotion classification.....	34
3.3.1 The types of different emotions.....	34

3.3.2 Discrete representation of words	35
3.3.3 Distributed table of words	36
Chapter 4 Improved Microblog Data Acquisition and Processing Methods	38
4.1 Microblog Simulation Login	39
4.2 Microblog Data Crawling	41
4.3 Microblog Data Parsing	42
4.4 Microblog Data Storage	46
4.5 Microblog Data Processing	47
4.6 Microblog Data Visualization	49
Chapter 5 A CNN-BiLSTM algorithm for Microblog emotion classification with attention mechanism	52
5.1 The overview of CNN-BiLSTM algorithm	52
5.1.1 CNN Network	52
5.1.2 BiLSTM neural network	52
5.1.3 The joint mechanism of CNN-BiLSTM algorithm	53
5.2 Attention mechanism	54
5.3 Data sources and processing	56
5.3.1 Construction of Microblog dataset	56
5.3.2 Dataset preprocessing	56
5.3.3 Training process	56
5.4 Experimental results and analysis	57
5.4.1 The benchmark model	57
5.4.2 The optimality of the architecture	59
5.5 Conclusions	60
REFERENCES	62
REFERENCES	72
BIOGRAPHY	74

LIST OF TABLES

	Page
Table 1 Experimental Environment	38
Table 2 Experimental comparison results	58
Table 3 Ablation test results	59



LIST OF FIGURES

	Page
Figure 1 Chart of changes in the size of Internet users and Internet penetration rate...	1
Figure 2 The scale and proportion of Chinese mobile Internet users	1
Figure 3 A comparison of the usage purposes of typical social apps.....	2
Figure 4 Number of monthly active users of microblog corporation from 1st quarter of 2015 to 1st quarter of 2023.....	3
Figure 5 The microblog vertical ecosystem.....	4
Figure 6 Emotion classification process based on emotion dictionary.....	16
Figure 7 Flow chart of data acquisition based on API.....	25
Figure 8 The operation mechanism of OAuth 2	26
Figure 9 The basic workflow of a crawler	28
Figure 10 Different data parsing methods.....	29
Figure 11 Basic flow of using the PyMongo library.....	30
Figure 12 Word2Vec Network Architecture.....	37
Figure 13 Simulated logon flowchart	40
Figure 14 The webpage source code for the username.....	43
Figure 15 The webpage source code for the repost	44
Figure 16 The crawled data.....	47
Figure 17 The flowchart of the preprocessing module	48
Figure 18 The data after the segmentation process.....	49
Figure 19 The data visualization.....	50
Figure 20 CNN network architecture.....	52
Figure 21 Bi LSTM network architecture.....	53
Figure 22 CNN BiLSTM joint mechanism.....	54
Figure 23 The attention mechanism architecture.....	55

Chapter 1 Introduction

1.1 Research Background

In recent years, Internet technology and related industry chains are experiencing rapid development, especially the rapid development of mobile Internet technology. More and more people are starting to use the Internet and its derivative products, resulting in an ever-expanding user base. According to the 51st Statistical Report on Internet Development in China released by China Internet Network Information Center (CNNIC) in March 2023, by December 2022, the number of Internet users in China will reach 1.067 billion [1], an increase of 35.49 million over December 2021, and the Internet penetration rate will reach 75.6%. Among them, the number of social media users accounts for 95.13% of all Internet users, as shown in Figure 1. In the whole scale of Internet users, after a long period of uninterrupted development, the scale of mobile Internet users has reached 1.065 billion, an increase of 36.36 million compared with December 2021, and the proportion of Internet users using mobile phones is 99.8%, as shown in Figure 2.

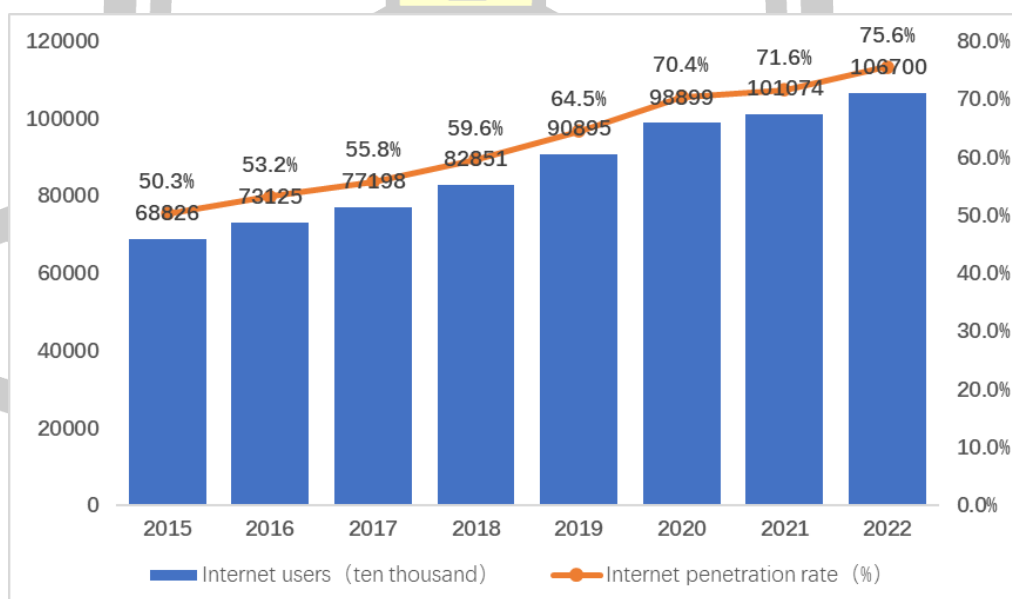


Figure 1 Chart of changes in the size of Internet users and Internet penetration rate

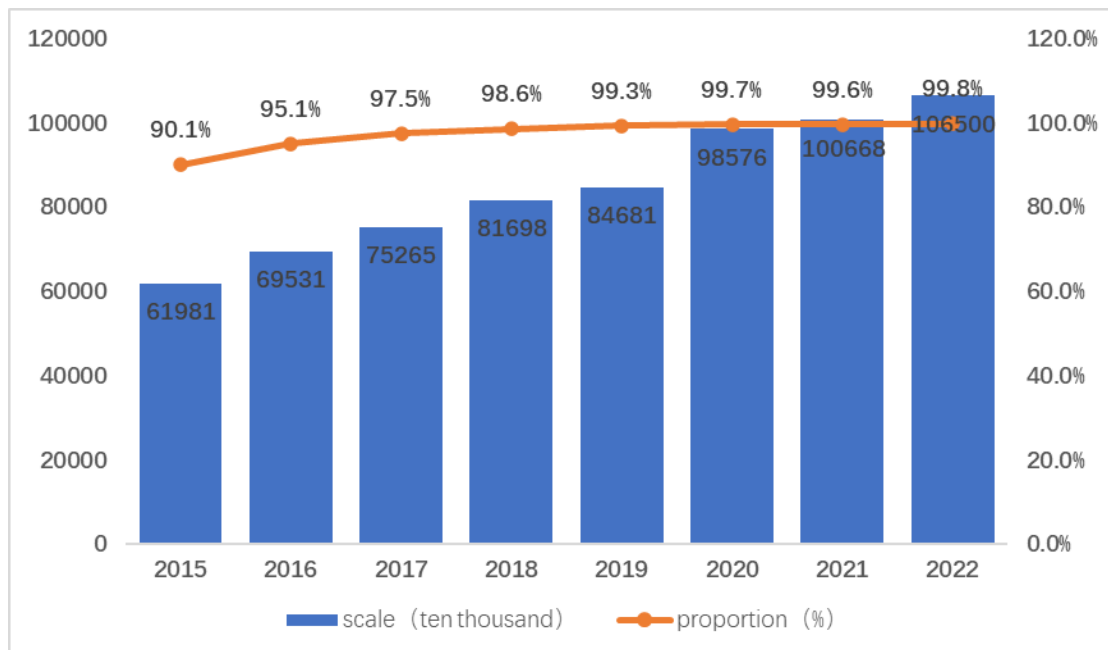


Figure 2 The scale and proportion of Chinese mobile Internet users

With the passage of time, the Internet has been integrated into people's daily life, and has become an indispensable and important element. Especially with the continuous development of the concept of "Web 3.0", people's interaction with Internet information has changed from passive acquisition to active creation, and this change has had a profound impact on Internet users and Web service providers. While Internet user groups browse web information, they are also unknowingly generating new Internet content. This process of content creation is no longer limited to consumption, but rather to the spontaneous incorporation of personal insights, ideas, and creativity into the Web. At the same time, Web service providers are actively pushing the boundaries of what is possible by launching various Web applications to meet the needs of users to create content and support this innovation and interaction. Many forms of social media platforms have emerged on the Internet, including social networking sites, Tiktok, WeChat, Microblog, and so on. These social media platforms provide important communication and sharing platforms for users to access information and communicate with each other in a variety of ways, as well as to express their views and ideas and derive valuable information from them. These platforms not only provide users with a window to connect with the world, but also

influence the dissemination of information and social interaction to a great extent. A comparison of the usage purposes of typical social apps, as shown in Figure 3.

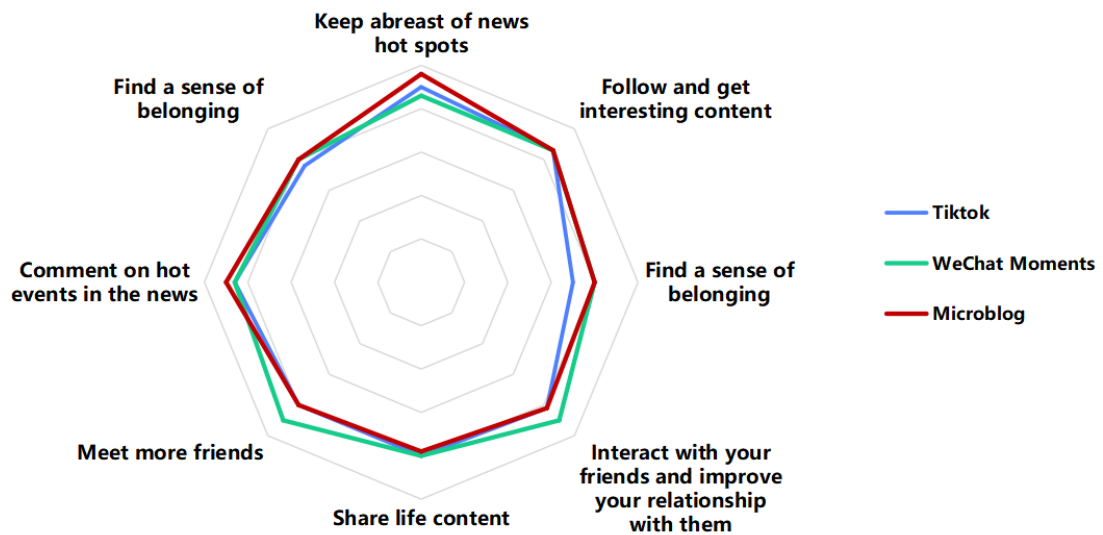


Figure 3 A comparison of the usage purposes of typical social apps

Microblog as the most popular social software in China today, is used as the main channel for information sharing and interactive communication among users [2]. It is also the largest social networking platform in China, accounting for 89% of all online social activities in China. Currently, it is known as the Chinese version of Twitter. Microblog brings a new way of interaction to users, allowing them to share information, make connections, disseminate content and access information in real time. Microblog has set up a set of rapid and comprehensive posting rules to enable users to share concise, real-time information instantly. In foreign regions where Twitter is the dominant social media, Twitter has grown rapidly due to its different interaction methods from other social platforms and the concept of providing users with a sense of convenience and speed, and has become the earliest microblog application recognized by Internet users as well as being widely used and promoted [3][4]. Sina's Q1 2023 earnings report shows that as of March 2023, microblog had 593 million monthly active users, a net increase of about 11 million year-on-year. Mobile users accounted for 95% of monthly active users. average daily active users in March 2023 were 255 million, a net increase of approximately 3 million year-over-

year. Number of monthly active users of microblog corporation from 1st quarter of 2015 to 1st quarter of 2023, as shown in Figure 4.

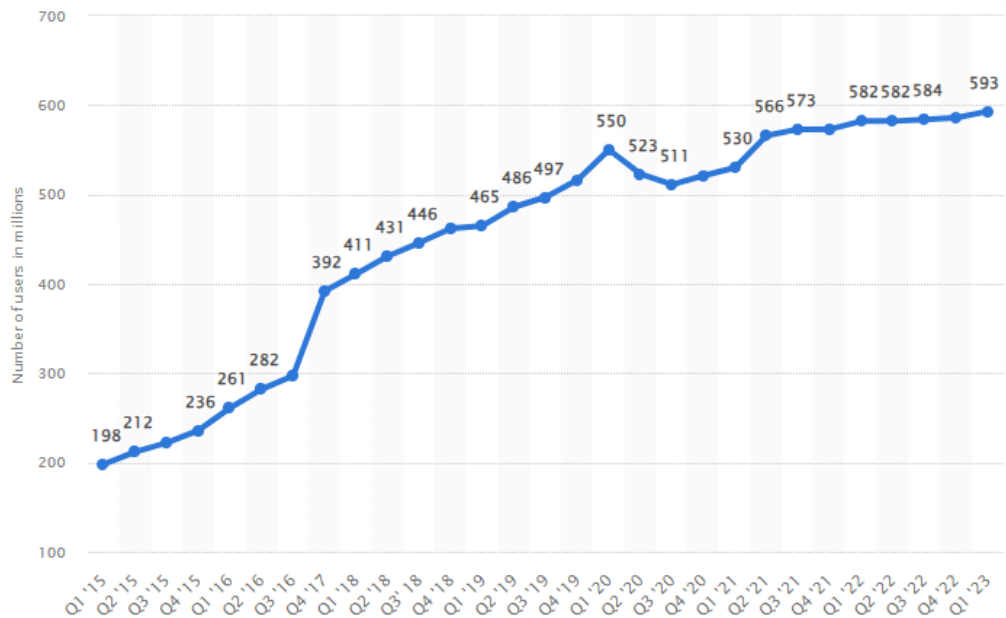


Figure 4 Number of monthly active users of microblog corporation from 1st quarter of 2015 to 1st quarter of 2023

With the rapid development of the mobile Internet, the mobile terminal platform represented by microblog is gradually integrating into people's lives and is closely connected with their daily lives. The rise of social media has accelerated the process of socialization, while the continuous evolution of the Internet has also prompted people's free time to be gradually filled by cell phones. This leads to fewer opportunities for face-to-face communication and a greater tendency for people to share their experiences through virtual platforms, such as social networks, and to reduce loneliness by interacting with others. People are increasingly willing to look to major information distribution platforms as their primary source of news compared to traditional ways of obtaining information (Such as newspapers, television and so on). Microblog has grown to have 60 different verticals with an increasingly robust ecosystem[5]. Microblog covers a wide range of everyday areas, which has attracted many companies, enterprises and social organizations. With the addition of these platforms, microblog has become a mainstream information dissemination and social media tool with a wider audience. Microblog has achieved comprehensive information

coverage in different fields, thus further strengthening its social influence[6].The microblog vertical ecosystem is shown in Figure 5.

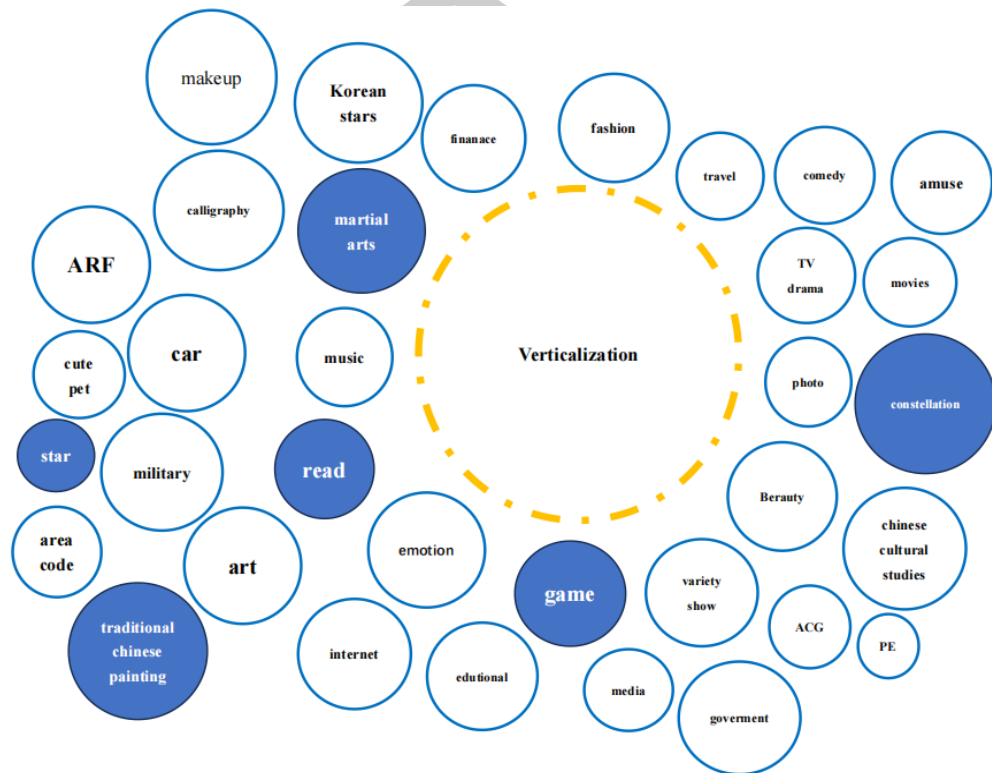


Figure 5 The microblog vertical ecosystem

The microblog platform has a huge user base of 255 million daily active users (DAU) every day, which leads to a constant influx of web data on the platform. The content of microblogs comes in various forms, including plain text, emoticons, pictures, videos, and so on. Users can express their emotions with microblog anytime, anywhere, and they can also retweet other people's microblog to express their own opinions[7]. However, the information on the microblog platform varies greatly, from users' preferences for popular celebrities, to their views on current political hotspots and national policies, to their comments on popular variety shows or the latest technological products, these short microblog textbooks and comment data, which are full of subjectivity, contain great commercial and social value. For example, businesses can collect users' comments or scores on a product to analyze its popularity and possible areas for improvement from a macro perspective; government agencies

can collect users' views on current political hotspots or national policies to analyze users' emotional tendencies in order to better grasp public opinion trends; and even some Internet celebrities can manipulate microblog hot searches to influence users' emotions and thus increase their attention. Even some internet celebrities can manipulate the hot search on microblog to influence the users' emotions, thus increasing their own attention. With the increasing number of users, the value of microblog information has become more representative. Under each popular event, there are various voices, and most of them are presented in the form of texts. These texts contain rich information about emotional tendencies, which are of great practical value and revelation for business, politics, education and other fields, so it becomes especially important to accurately grasp the emotional tendencies of users.

1.2 Research Objectives

This paper proposes a research on microblog text sentiment analysis based on web crawler, which mainly contains several aspects such as microblog data crawling, microblog data preprocessing, and microblog text sentiment analysis.

(1) Microblog data acquisition

On the microblogging platform, the data presents the characteristics of huge quantity and diverse forms. In order to realize efficient data acquisition, we choose to take the text data, which accounts for the largest proportion of microblog data, as the main crawling target. Web crawlers may encounter various anti-crawling difficulties when crawling microblog data, because websites take a series of measures to restrict crawler activities in order to protect data security and prevent malicious crawling. Based on the study of microblog web crawling workflow, we analyze the shortcomings of the existing microblog web crawling strategy and propose an improved microblog crawler crawling strategy.

(2) Microblog data preprocessing

In the process of data crawling, the obtained data are often interfered by various noises, which contain a large number of meaningless texts, symbols, links and emoticons. These noises will interfere with the analysis of the sentiment tendency of the text, so the data need to be cleaned. In order to obtain accurate results of sentiment

analysis, this paper adopts the technical means of regular expressions, jieba splitting and removal of stop words to process the data obtained by crawling. The purpose of these steps is to extract the "pure" data related to sentiment analysis.

(3) Sentiment analysis of microblog data

Microblog short text information contains a large amount of network language, emoticons, etc., and due to the long-time span of the content, the emotions of the posts posted by people often change due to time or the occurrence of certain special events. Therefore, traditional sentiment analysis methods are not suitable for this task. This article proposes a CNNs-Bi LSTM sentiment analysis method that integrates attention mechanism. It combines convolutional neural networks and bidirectional short-term memory networks to obtain keyword information in text through attention mechanism, efficiently and accurately realizing data temporal and semantic information mining. Through experimental verification using Microblog public opinion data, the results show that this method achieves higher accuracy compared to other benchmark models and can fully utilize multidimensional matrices to capture rich text features, with certain advantages.

1.3 Benefits of the Research

Microblog users mainly create content through texts, and these texts contain a wealth of useful information. In recent years, microblog text sentiment analysis has become a new research field that has attracted much attention. This is because microblog text data contain a lot of important information, which has important research value for a variety of tasks such as sentiment tendency and market research. In different application scenarios, these text data can be helpful for in-depth analysis and insights.

For government users, by analyzing the sentiment tendencies in microblog texts, the government can more accurately capture the emotional dynamics of the people and gain a deeper understanding of the emotional state of different regions and groups of people, thus providing real-time and accurate data support for policy making and livelihood research. In addition, microblog sentiment analysis can also help the government grasp social hotspots and changes in public opinion in a timely manner,

providing important references for responding to public opinion and maintaining social stability.

For corporate users, microblog text sentiment analysis can provide insight into consumers' emotional attitudes towards products and services. By analyzing the comments and evaluations of microblog users on specific commodities, enterprises can understand consumers' preferences, needs and feedback, and then optimize product design and improve service quality to better meet market demand. In addition, enterprises can use sentiment analysis to predict market trends, develop more accurate marketing strategies, and improve market competitiveness.

For individual users, microblog text can help to understand the user's inner emotions and record the emotional changes over time. Sentiment analysis of microblog text is important for individual mental health concerns and interventions. By analyzing the texts posted by individuals on microblogs, it is possible to understand their emotional changes and psychological states, providing clues for early detection of mood swings and psychological problems. Especially in the face of increasing mental health problems, microblog sentiment analysis can provide strong support for mental health professionals to intervene with patients in a timely manner and provide appropriate psychological counseling and support.

In summary, microblog text sentiment analysis is not only a promising research direction, but also a technological application that benefits multiple levels and fields. By digging deeper into the sentiment information in microblog text, we are able to provide more accurate and targeted decision-making support and services to governments, enterprises and individuals, thus maximizing the social benefits.

พหุ ประเด็น โท ชีเว

Chapter 2 Literature Reviews

2.1 Current Status of Data Acquisition Research

In foreign countries, research on the collection of Twitter microblogging data mainly focuses on studying the use of TwitterAPI. This is because TwitterAPI is relatively mature in foreign countries, and it is also supported by rich third-party APIs and detailed development documents. Therefore, most foreign Twitter data collection systems are based on the existing TwitterAPI for secondary development.

In their study, researchers such as Zeel Doshi developed a tool that can analyze Twitter sentiment tendencies and perform visual analysis by using a third-party API tool, Tweepy. On the other hand, researchers such as Java used Twitter's Search Application Programming Interface (API) to collect tweet data from it. They extracted discrete features about trending issues by categorizing the content of tweets and used it to build a system to recommend relevant content to target users.

The current methods on microblog data collection mainly include the following:

- (1) data collection based on the API provided by microblog
- (2) methods combining microblog API and web crawler technology
- (3) data crawling based on web crawler technology.

Web crawlers, as an important part of search engines, have been subject to continuous attention and research by scholars and well-known enterprises at home and abroad. As early as 1993, the world gave birth to the first web crawler, which was designed and developed by Matthew Gray [8] and named World Wide Web Wanderer. the initial design goal was to calculate the number of servers in the network, which was later used for web crawling. However, limited by the state of the art at the time, the crawler actually indexed only about 6000 web pages. Due to the limited network resources at the beginning, a standalone web crawler running independently was sufficient to fulfill the users' needs. However, with the continuous progress of network technology, the information carried in the network showed a geometric growth. The gradual emergence of distributed technologies to fill the deficiencies of stand-alone web crawlers in handling massive data crawling and analyzing in the era of big data.

The introduction of these techniques enables more efficient crawling and processing in the face of huge data volumes to meet the challenges of information explosion.

In 1994, Eichmann [9] at the University of Houston developed an Internet crawler called RBSE (Repository Based Software Engineering) with funding from NASA. The crawler utilizes the working principle of search engines with the goal of retrieving highly relevant documents. The core component is a crawler program called Spider, which is capable of creating and managing Oracle databases. By employing a breadth-first traversal strategy with limited depth, it traverses pages linked by `"/html"` or `"http:"`. One of the main functions of RBSE is to identify and store new resources or sources of information in the repository before notifying the user with One of the main functions of RBSE is to identify and store new resources or information sources in the repository and then notify the user about these new resources. This crawler system was designed and developed to help users better access and manage content on the Internet through efficient information retrieval. This approach contributed to the advancement of the information retrieval field at that time.

In 1999, Calton Pu and others [10] created a web crawler system using a search mechanism similar to that provided by WAIS (Wide Area Information System). This system allowed the crawler's triggers to be customized according to the user's needs and came with a push system that proactively pushed information about new resources crawled to the user. Subsequently, in 2001, IBM introduced WebFountain [11], a distributed web crawler system similar to the high-performance crawler Mercator [12]. WebFountain consists of a master node and multiple slave nodes that collaborate to accomplish the crawling task, and also possesses excellent scalability.

In 2002, Cho J and others [13] first categorized distributed crawlers and proposed a set of crawler evaluation criteria to compare traditional stand-alone crawlers with distributed crawlers, and the results showed that distributed crawlers are more efficient. Among them, the well-known distributed crawler system UbiCrawler [14] shows good compatibility and scalability in terms of performance. This system can be deployed on multiple computers, while a web page de-duplication mechanism is introduced in the crawler system to prevent crawling data information from duplicate web pages.

In 2004, Maggini M and others [15] proposed a strategy for controlling the bandwidth of the crawler after studying document importance based and topic based crawling strategies. This strategy is applicable to small search engines by monitoring the transfer rate statistics of each server to determine the optimal number of parallel downloads.

In 2006, Harth A and Umbrich J and others [16] proposed a semantic based distributed crawler system which consists of five phases such as crawling, detection, transformation, indexing and extraction. In this system, the crawled web content is converted into a directed markup graph and indexed for structured queries. Meanwhile, Matsuo Y and others [17] proposed a social network crawler system called "POLYPHONET" for detecting the relationship of data and information between people and obtaining documents shared on the network. The scalability and relevance recognition features of a simplified search engine crawler module are utilized in order to repeatedly identify each entity and crawl the relationships to obtain more accurate information about social network data.

In 2009, Matthew Hurst and Alexey Maykov [18] designed and implemented a web crawler for social media blogs. Since blogs and other social media differ from traditional web content in several ways, the system utilizes the time of posting a blog as a URL prioritization condition to ensure the quality and importance of the blog crawl. The system includes a scheduling subsystem and a fetching subsystem to issue URL requests one by one as the fetcher's capacity allows for load balancing.

In 2011, Lianjie and others [19] on the other hand, solved the problem of incomplete data collection due to API limitations, and successfully realized comprehensive and efficient access and data collection for Sina Microblog. Philipp Berger and Patrick Hennig and others [20] improved the feed-crawler to become a general and scalable of a blog crawler. Data crawling of blog posts was implemented through the MapReduce [21] and Apache Nutch [22] frameworks, and new analytics were explored, utilizing in-memory databases for support.

In 2013, Shunhua Chen and others [23] proposed a microblogging data collection scheme based on microblogging APIs, which successfully achieved efficient

microblogging data acquisition by finely controlling the number and frequency of API calls and employing distributed crawling techniques. Rafael Ferreira and Fred Freitas [24] further investigated distributed crawling of social media blogs, and proposed RetriBlog, an architecture-centered framework for crawlers that crawl blog data information. By adopting an architectural development approach, the complexity of development can be better controlled, thus facilitating a parallel development process and reducing system maintenance costs.

In 2014, Lavanya Pamupart and others [25] proposed a new approach to avoid web crawler crawling overload. Since it is often difficult for servers to respond in a timely manner when downloading large files or executing a large number of task requests per second, the method introduces a sequential streaming protocol (Leakey Bucket) that retrieves data in a sequential manner in order to ameliorate the overloading problem during data retrieval. Meanwhile, a mobbing unrestricted router (MURR) consisting of three phases: inbound, peak and outbound is proposed to allow crawling when the traffic is less than the peak, otherwise the server will be requested to extend the network resources in order to improve the crawling overload problem effectively.

In 2015, Mingxiang Zhong and others [26] used a browser-controlled multithreaded crawling approach, which leverages the browser kernel to parse dynamically loaded data in microblog pages, especially by processing the asynchronously loaded content of Ajax on microblog pages. This approach further enables user-centered data acquisition, allowing researchers to obtain more comprehensive information about user-related data.

In 2016, Gossen [27] proposed an open-source focused crawler called iCrawl, which aims to collect fresh content and relevant web page information related to specific topics in social networks. This crawler system seamlessly integrates web and social media to collect web and social media data in a single system. In addition, it utilizes newly generated data streams from social media to guide the crawler in order to extend the completeness of the content it crawls. Tested with the social platform Twitter as a source of fresh data, the results demonstrate iCrawl's ability to take full advantage of accurate focused crawling as well as the continuous input of fresh data

streams from social media when collecting real-world datasets. This system also has the ability to automatically adjust the crawling direction to adapt to the newly generated information sources. Lu Yang [28] proposed an innovative approach to simulate user logins on the Sina platform using P2P technology in order to obtain data related to hot microblogging topics. This approach successfully overcomes the limitation of the limited number of APIs provided by microblog to grab more data from the system. With this approach, they developed a highly stable crawler system that can work continuously for 24 hours, thus enabling round-the-clock collection of data on microblog. This innovative work achieved remarkable results in terms of scalability and stability, providing strong support for crawling and analyzing microblog data.

In 2017, Dani Gunawan [29] proposed a distributed focused web crawling approach. One computer was used as the crawler host and four crawler slaves with the same hardware were configured to achieve efficient web content crawling. Although adding threads increases the network load, the optimal number of threads needs to be determined to strike a balance between efficiency and load.

In 2018, Yunfang Liu [30] improved an accuracy-aware incremental crawling algorithm aimed at improving the effectiveness of web crawlers. Using the technique of mixed integer quadratic programming, an optimal refresh strategy was designed for those web links whose content changed. This refresh strategy aims to optimize the execution of the parallel crawler, thus improving its performance and also the quality of the page updates.

In 2019, Sekhar S [31] proposed a natural language processing based page relevance prediction algorithm called TextRank for focused web crawlers. This algorithm determines the degree of relevance of a web page to a topic based on the frequency of occurrence of the topic word in the top ranked sentences in the web page content. By ranking the sentences while ensuring that the content of the original web page data is correctly categorized, the algorithm is able to crawl web content around the topic without being marginalized. By comparing it to a web crawler with a breadth-first strategy, the experiment confirms the effectiveness of the TextRank algorithm and its ability to crawl around topics.

In 2020, Dr. G S Pradeep Ghantasala and others [32] proposed a method to compute web page similarity at high speed by combining the combined structure of PageRank web algorithm and Shark look web algorithm. Comparing the Shark-Search algorithm with the previous PageRank algorithm, it was found that the web crawler using this structure was able to scan more deeply into the web area where the similarity sites are located. At the same time, the crawler system stops crawling data information for the network area of irrelevant pages, which improves the crawler system's efficiency in crawling high-value web information.

In 2021, Rongrong Xie [33] proposed a simulation of web crawler-based big data crawling method for web pages. Xiangyu Long [34] Shengmin Zhang [35] faced the need of obtaining useful information and deeply optimized the crawler data collector by using Python software on the basis of generic web crawler technique. Lei Gang and others [36] proposed a public cloud-based microblogging web crawler and public opinion visualization and analysis experimental platform design, including the virtual environment construction of the public cloud platform, web crawler programming, Sina microblog comment data automatic collection, public opinion analysis and data visualization and other modules, which formed a comprehensive experimental project of the characteristics of the innovative practice of smart media communication, media integration and so on. The practical results show that the study is conducive to improving students' media literacy, news reporting practical ability and innovation ability.

In 2022, He Bo [37] proposed the maintenance method of crawler program in microblog based on Python in order to improve the stable operation ability of the microblogging platform and give users a better social experience. Sun Zhaoyu[38] proposed the design and implementation of microblog crawler program based on Python, using Python language to design the microblog crawler program, based on the study of the basic concepts and principles of web crawler, designed an application with four functional modules of configuration, crawling, storage, and analysis, which provides technical support for the regulation of media content and data analysis. Lu Lili [39] proposed the design and implementation of a social platform opinion user tracking system based on crawlers, through crawlers and deep learning methods, as

well as a big data analysis platform to realize the association mining for user inappropriate comment information, by crawling the microblogging social platform data to validate the system architecture and methods proposed in the paper, and achieved good results.

In 2023, Deng Xiaolu and others [40] proposed a Scrapy-based microblog data crawler study, exploring the implementation of a web crawler tool based on the Scrapy framework to accomplish data capture and other objectives. The tool searches for relevant microblog information based on one or more microblog keywords and writes the search results to a local file. Zhu Rongshen and others [41] proposed a distributed network oriented anti-crawler data orderliness study for the problem of disorder in anti-crawler data to reduce the difficulty of data subsequent processing. Sun Chuan-nang and others [42] proposed a study on keyword matching retrieval for web crawlers based on Python language. Combined with user preferences, user-related information is obtained from user labels, retrieval history, browsing behavior, etc., a user feature model is constructed, and the pseudo-code feature values of Python language are calculated. Determine the crawler path, obtain the keyword and non-keyword node feature weights, and realize the keyword matching retrieval of web crawler. Dolin Liu [43] proposed a distributed web crawler data collection algorithm under Scrapy framework to improve the data collection speed and avoid repeated collection.

2.2 State of the Art of Sentiment Analysis Research

Natural Language Processing (NLP) is one of the major directions in the field of artificial intelligence, which is a discipline that studies how to make computers understand human language. Nowadays, more and more Internet users can express their concepts and share their ideas and opinions on online platforms. Therefore, it is possible to analyze the emotional color and tendency of the speech of the users of major platforms, which is important for the government to understand the people's hearts and minds, and to serve the people; for merchants and enterprises, knowing where the customer's preferences and needs are will help merchants to improve their own services, and to launch new products; in the aspect of emotional catharsis, computers can also act as a window of emotional catharsis and may bring help to people who have repressed emotions. Suppressed emotions may be able to bring help.

Sentiment analysis is the use of computer algorithms to automatically detect emotional valence, or sentiment, from text or speech, usually by inferring positive, negative, or neutral emotional valence from a piece of text, or the number of stars in a product or movie review, or the degree of positivity indicated by a real number between 0 and 1 in a passage of text. More generally, sentiment analysis can infer a person's attitude toward a particular goal or topic, which can be an evaluative judgment such as positive, negative, or an emotion such as frustration, happiness, anger, sadness, excitement, etc. Sentiment analysis can also refer to inferring positive, negative, or neutral sentiment validity from a piece of text, or a product or movie review, or a real degree of positivity between 0 and 1. Sentiment analysis can also refer to the determination of a person's emotional state from his or her expression, whether or not the expression is directed at an external entity [44].

The concept of sentiment analysis has attracted extensive attention from scholars in the field of natural language processing since it was introduced by Nasukawa [45] in 2003. Currently, there are three mainstream sentiment analysis methods: one is based on sentiment lexicon; the other is based on traditional machine learning; and the third is based on deep learning.

2.2.1 Sentiment Classification based on Sentiment Dictionary

The idea of sentiment classification based on sentiment dictionary is relatively simple and easy to implement. The general process of this method is shown in Figure 6, after inputting text data, it is first subjected to a series of preprocessing operations, such as word division and deactivation, and then the constructed sentiment dictionary is used to traverse the preprocessed words for text matching, and the words are weighted according to the judgment rules, and the final output is the sentiment type of the text. Among them, the sentiment dictionary consists of two parts: words and corresponding weights. If the words after word separation and the words in the dictionary match, they will be weighted accordingly: the weights of positive words are cumulative to each other, the weights of negative words are cumulative to each other, the weight of negative words is -1, and the weights of adverbs of degree and negative words are multiplied with the weights of the words they modify [46][47].

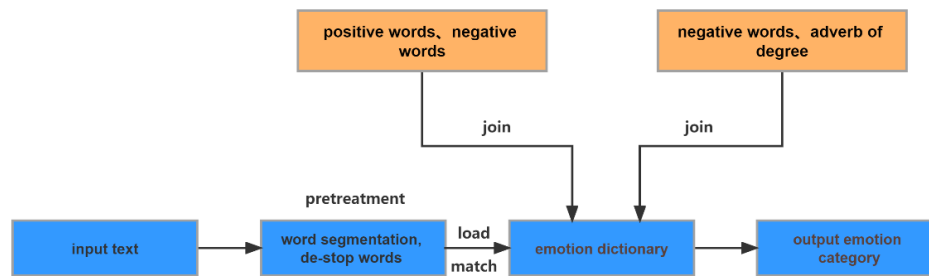


Figure 6 Emotion classification process based on emotion dictionary

H. Wang [48] proposed a new emotion classification model based on hybrid learning. In the first stage, the improved dictionary classification method is used to calculate emotion score in the whole data set, and the data with high or low scores are directly marked; in the second stage, the rest of the method is based on emotion dictionary and Bi-GRU fusion model to calculate emotion score, and the two stage hybrid frame makes the method effectively applied in microblog emotion classification. K. Jia [49] Based on the existing emotion dictionary, the paper improves the calculation method of emotion intensity of different emotion categories of emotion words in emotion dictionary. W. Tao [50] Research on the impact of the high dimension of comment text features and the implicitness of text expression on sentiment classification performance, a deep learning sentiment classification model based on topic and sentiment features is proposed for sentiment classification of comment texts. Y. Zhang [51] mainly divided the text comments of e-commerce from the three aspects of keywords, evaluation objects and emotional resources, and classifies them according to the constructed emotional dictionary. Xie Ruobing and others [52] proposed three improved skip-gram models based on Hownet for precise analysis of text sentiment trends. Hua Bei and others [53] compiled and merged several foundational sentiment lexicons, including HowNet and NTUSD. They incorporated emoticon dictionaries and internet vocabulary dictionaries and expanded new words using an improved SO-PMI algorithm, creating a relatively comprehensive Chinese foundational sentiment lexicon based on this foundation.

Ahmed and others [54] constructed a multilingual lexicon by learning sentiment clustering from sentence context, and experiments show that this method can well

improve the accuracy of the discrimination; Zhao Yanyan[55] crawled about 146 million microblogs by means of a crawler and carried out the work of classifying positive and negative words, and constructed a sentiment lexicon based on the text counting algorithm of 50,000 positive words and 50,000 negative words respectively. Zhang and others [56] further extended the sentiment dictionary by combining the degree adverb dictionary, the negative word dictionary, the Internet vocabulary dictionary and other related dictionaries; Chen Keyu and others[57] improved the sentiment dictionary and realized the sentiment classification work based on it, and obtained a high accuracy rate, but this method is only applicable to the hotel field, and the application in other fields is not effective. Wang Xuehe [58] and others statistically analyzed the crawled comments of microblogging breaking news in 2021, and constructed a sentiment dictionary of breaking news through the annotation of lexical properties, and then combined with SnowNLP, and the experiments showed that the model improved the accuracy of sentiment analysis of breaking news.

The semantic similarity-based sentiment dictionary construction method is used to measure the distance between sentiment words and positive and negative labels, according to this method L and others [59] proposed a point-by-point mutual information PMI (Point-wise Mutual Information) measure, which is able to judge the polarity of the sentiment words very well, but ignores the sentiment representation of the sentiment words in the contextual semantics, which makes the sentiment polarity judgment is prone to bias. Turney and others [60] proposed a phrase affective tendency calculation method of PMI -IR on the basis of PMI, to find the average value of affective tendency of text phrases as the result of judging affective polarity. This traditional affective dictionary based calculation method is more dependent on lexicality, and some words in the text contain multiple meanings of words or reverse expressions, which express different emotions in different contexts, but in the traditional affective dictionary, the emotion categories are only judged according to the emotion words, which cannot accurately express the emotion polarity of the words containing the contextual contexts. Therefore, Kang Yue and others [61] proposed the method of building 100,000 sentiment dictionaries fused into one large-scale sentiment

dictionary to deal with the sentiment classification problem when dealing with semantically complex Chinese data, which can well solve the above existing problems such as multiple meanings of the word sentiment words. Huanhuan Yuan and others [62], in order to reduce the dependence on manually constructed sentiment dictionaries and highlight the keywords' In order to reduce the reliance on manually constructed sentiment dictionaries, Huanhuan Yuan and others [62] proposed weighted word2vec and added an attention mechanism in the Long Short-Term Memory (LSTM) model to give full play to the role of keywords in sentiment categorization, so that the sentiment words can express the meaning of the original text more effectively, and the sentiment polarity of the text can be judged more accurately, but the effect of contextual structure on the sentiment polarity is also neglected. Based on the problems of poor adaptability and tedious manual operation of sentiment dictionaries, scholars have applied machine learning and deep learning methods to sentiment dictionaries in order to construct sentiment dictionaries that can improve the accuracy of sentiment categorization. Vaibhavi and others [63] applied Bayesian models to sentiment dictionaries, and the sentiment words in text data were classified and trained by the Plain Bayesian model, in order to reduce the domain irrelevant sex on the expression of emotion. Li Yongshuai and others [64] also applied neural networks to emotion dictionaries, and they used a bidirectional LSTM model to construct a dynamic emotion dictionary to improve the accuracy of text classification.

With the continuous progress and development of social networks, a large number of network neologisms are gradually generated on social platforms. Since the sentiment dictionary cannot be updated in time, it lacks the recognition of new words when using the sentiment dictionary approach for text sentiment classification. In addition, the classification accuracy of sentiment classification methods based on sentiment dictionaries is easily limited by the quality of the dictionaries, and overly relies on the inclusion of labeled sentiment words in the dictionaries, so the classification accuracy is easily affected and the classification accuracy is not high. In order to improve the classification accuracy, scholars began to introduce machine learning methods into the field of sentiment analysis and achieved good improvement.

2.2.2 Sentiment Classification based on Machine Learning

Sentiment classification using machine learning methods is mainly divided into supervised, unsupervised and semi-supervised, the most commonly used is the supervised sentiment classification method, whose general implementation process is to prioritize the use of a large number of emotionally labeled corpus to train the sentiment classification model, and then input the unlabeled text to be tested into the model to predict the sentiment category.

In supervised methods, the provided data are labeled. Supervised learning For each input, the learner is provided with a response target. Some of the common supervised methods are Naïve Bayes, SVM, KNN. In unsupervised methods the data provided is unlabeled and is automatically categorized based on the intrinsic attributes of the data. There are few applications of such methods in the field of sentiment analysis. Semi-supervised learning uses a small number of labeled and a large number of unlabeled samples to perform supervised or unsupervised learning tasks and falls between supervised and unsupervised learning. In semi-supervised methods, the classification results can be greatly improved by performing feature extraction on unlabeled text, which makes the problem effective when the number of labeled samples is small.

Many researchers have worked on sentiment analysis using supervised machine learning methods. Tang and others [65] proposed a regression SVM emotion classification model in order to visualize the expression of textual emotion intensity thresholds; Haque and others [66] used Linear SVM, Multinomial Naïve Bayes, Stochastic Gradient Descent, Randomized Gradient Descent, and Stochastic Sentiment Classification model to classify product reviews in the areas of "Mobile Phones and Accessories", "Music", and "Electrical Appliances" on Amazon. Gradient Descent, Random Forest, Logistic regression and Decision Tree methods for sentiment analysis, the best classification results were obtained using SVM on the music domain with an accuracy of 94.02%; Tarmer L and others [67] realized sentiment analysis on movie review dataset with the help of Naïve Bayes, SVM and Decision Tree algorithms, and the experimental results showed that the SVM algorithm had the highest accuracy rate under the task; Liu and others [68] compared the three classification methods of

Support Vector Machines, Bayesian models and n-tuple language models, the three feature selection methods of information gain, CHI statistics and document frequency and the three feature weighting methods of Boolean The experimental results showed that support vector machine, TF-IDF and information gain are the most effective recommendation algorithms in the three integrated recommendation algorithms.

The experimental results show that the support vector machine, TF-IDF and information gain algorithms have the best effect. Liu Yong [69] proposed an improved random forest model for text classification application, optimized for the voting method, using a combination of decision tree, predictive probability weighted voting and combining the random search and grid search strategy to optimize the hyper-parameters for continuous correction and adjustment, the method achieved good performance. Huang Lei and others [70] optimized the lexical annotation sequence, combined with the maximum entropy classification model, to determine the sentiment category of microblog text. Songyan Zhang [71] proposed a semi-supervised method fusing consistency regularity and popularity regularity for sentiment classification task.

However, traditional machine learning methods have some limitations in text feature extraction and cannot fully capture the contextual information of words. The training effect of the model is often highly dependent on the training corpus, while manual labeling of the corpus carries the risk of subjectivity, and the labeling process is cumbersome and time-consuming, requiring a large number of human resources [72]. In order to make up for the shortcomings of machine learning methods and improve the classification accuracy of sentiment analysis, research scholars have begun to explore the application of deep learning algorithms in the field of sentiment analysis, and have gradually achieved significant results. Soumya S [73] proposed fine grained sentiment analysis of malayalam tweets using lexicon based and machine learning based approaches, the machine learning algorithms such as Support Vector Machine (SVM) and Random Forest (RF) classifiers are used for sentiment classification of the dataset. Bingyun Lin [74] taking the “Viya's tax evasion” case as an example, firstly, the Latent Dirichlet allocation (LDA) topic model is used to construct a topic model of Microblog content in the case, and the optimal number of

topics is determined by perplexity. Then, the k-prototype algorithm and the improved Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm are respectively used to cluster Microblog users and analyze the similarities and differences between the various categories of users.

2.2.3 Sentiment Classification based on Deep Learning

Traditional machine learning-based methods often require a lot of time and resources for feature engineering and functional design, which makes the whole process seem complicated and expensive. Not only that, these methods often fail to capture the natural order structure and contextual information in textual data well, which leads to making it difficult to learn the semantic information of words. However, since 2010, we have gradually witnessed a transition from traditional machine learning models to deep learning models in the field of text categorization. The rise of deep learning methods has revolutionized this field. Compared to traditional machine learning methods, deep learning methods no longer need to rely on tedious manual design rules and complex feature engineering. Instead, they are able to automatically extract semantically meaningful expressions from text data, which makes text mining more accurate and efficient. Especially in the direction of sentiment categorization, deep learning is gaining momentum. Deep learning models are not only able to adapt to different types and lengths of text, but also capture contextual information, semantic relationships, and complex features, leading to remarkable results in sentiment classification tasks. This trend signals an exceptionally bright future for deep learning in the field of natural language processing, which will bring us more accurate and efficient methods for text categorization.

In the research and exploration of Artificial Neural Networks (ANN), the concept of deep learning has been introduced, aiming to simulate the neural structure of the human brain. With the continuous development of deep learning technology, its application in the field of natural language processing (NLP) is gradually favored. In the task of text sentiment categorization, deep learning-based methods mainly introduce deep learning models for determining text sentiment categories. One important advancement in deep learning is the introduction of Word Embeddings, a concept first introduced by Hinton in 1986 [75]. The word vector approach captures

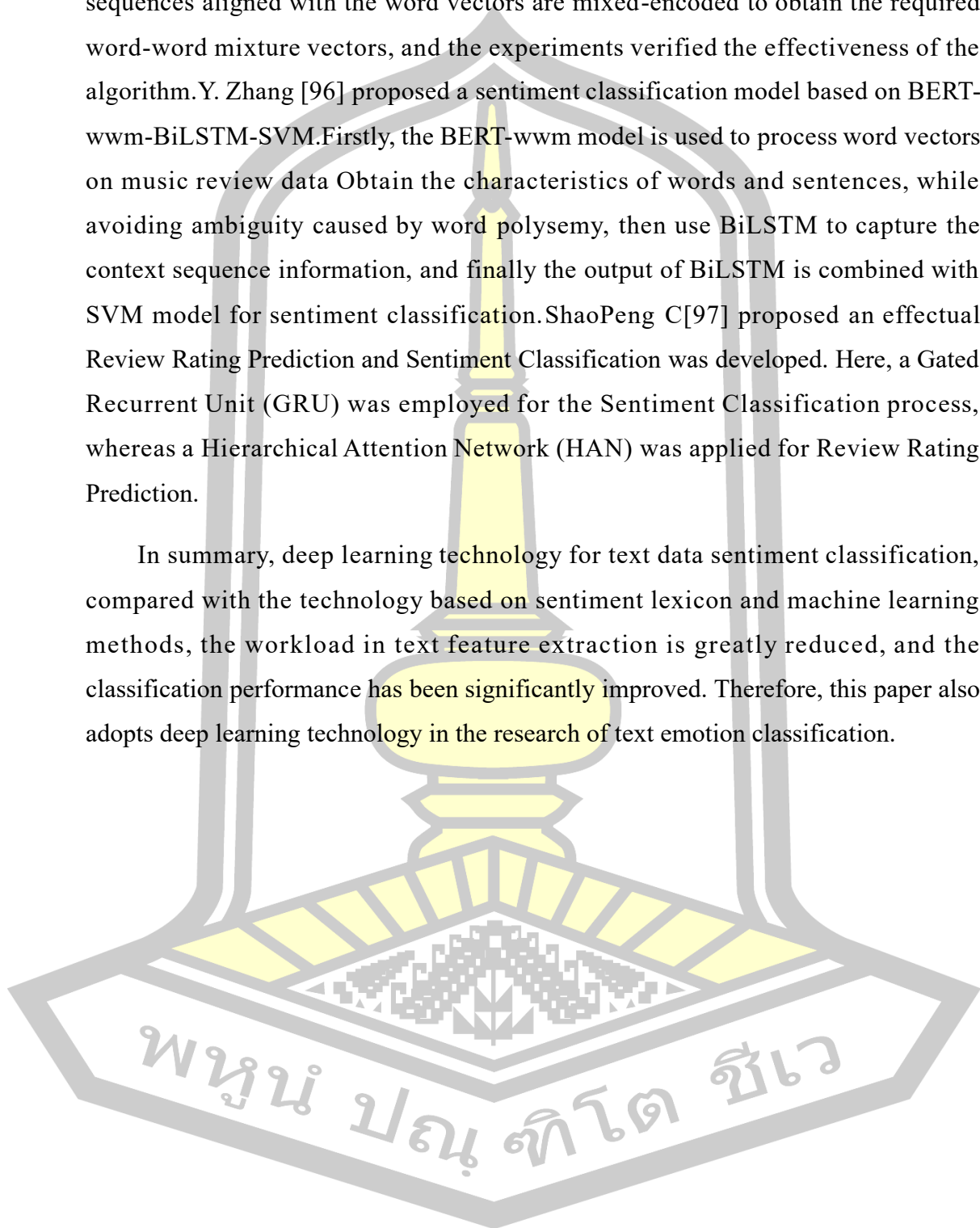
semantic information in text [76]. In 2013, Mikolov and others [77] proposed two variants of the word vector model: the CBOW (Continuous Bag of Words) model and the Skip-Gram model, which are collectively referred to as word2vec. These word vector techniques are able to quickly learn high-quality word vectors from a large amount of unlabeled data, which provides a textual representation in a more semantic representation of text. In addition to word2vec, there is another word vector model based on full-text word frequency, i.e., the GloVe (Global Vectors for Word Representation) model. the GloVe model not only takes into account the co-occurrence information between words, but also better captures the correlation between words [78].

In the field of Natural Language Processing (NLP), the most commonly used neural network models are Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN), respectively. Several studies have pointed out that CNNs are effective in capturing and extracting locally relevant information from text. Li J and others [79] constructed a sentiment classification model based on product reviews using CNN, which mainly focuses on labeling product feature words and applying word vector techniques to cluster these feature words. Experiments proved that the model has high accuracy in sentiment classification. Chen Ke and others [80] used multi-channel CNN to fuse the local features extracted from multiple channels after pooling layer, this model is able to obtain multi-dimensional local information in text sentences, and the experiment proves that it has effectiveness. Luo Fan and others [81] proposed a multilayer network model HRNN-CNN, which uses a two-layer recurrent neural network to model the text and introduces a sentence layer, and then combines the convolutional neural network to achieve the sentiment classification of long text; although the CNN performs well in capturing the local information, however, there are some shortcomings in dealing with the text that there are long sequential dependencies [82]. At this point, variants of Recurrent Neural Networks (RNN), such as Long Short-Term Memory Networks (LSTMs) and Gated Recurrent Units (GRUs), are advantageous. Ma and others [83], in an effort to integrate explicit and implicit knowledge, proposed an extension of the LSTM, known as the Perceptual LSTM, where the extended LSTM unit includes a separate output gate for insertion of

symbolic-level memory and conceptual-level inputs, which performed well on the two benchmark datasets; Fei and others [84] proposed a model based on long and short-term memory networks that can handle arrays, vectors and high-dimensional data for multi-topic classification, which was experimentally found to be able to reach 96.5% in terms of the accuracy metrics; Han, Jiansheng and others [85] used a bidirectional temporal convolutional network for sentiment categorization, whereas Zhu and others [86] used a long and short-term memory network to deal with comment statements that require long-distance dependency; Hu and others [87] found some keywords that contribute greatly to the semantics of the text, built a keyword vocabulary, and combined it with an LSTM model, and the experimental results show that this method outperforms the baseline LSTM in terms of accuracy; Wang and others [88] constructed a capsule-tree LSTM model in order to solve the problem of emphasis bias that exists in both the LSTM and tree LSTM. LSTM model. They introduced a dynamic routing algorithm as an aggregation layer, and constructed sentences by assigning different weights to nodes to represent their contribution to sentiment polarity. Li Weijiang and others [89] built a bi-directional LSTM based model to analyze textual information and emotional resources and assigned different channels to them to fully utilize the emotional information. Wang and others [90] proposed a fine-grained model for sentiment analysis, their approach combines CNN and Long Short-Term Memory (LSTM) neural network (LSTM) model, using CNN to obtain the local information of a sentence, and all the local information serves as inputs to the LSTM model. Tang and others [91] developed an algorithm to solve the problem of encoding the problem of intrinsic relationships between sentences in a document, a model combining CNN, LSTM and gated recurrent neural network was proposed, which showed good performance on four datasets. Alayba [92] proposed a model combining CNN and LSTM, which synthesized the advantages of both, solving the problem of missing local information as well as capturing long term dependencies in the text sequence. Zhang [93] proposed a model combining sentiment word vectors and CNN for sentence-level sentiment classification. Kai Zhou [94] combined a bidirectional long and short-term memory network (BiLSTM) with CNN to improve the performance of text sentiment analysis. Chen Xin and others [95] proposed a new text sentiment classification method using BiLSTM network and word fusion, mainly

through the pre-trained word2vec word vectors, the word vectors and the word vector sequences aligned with the word vectors are mixed-encoded to obtain the required word-word mixture vectors, and the experiments verified the effectiveness of the algorithm. Y. Zhang [96] proposed a sentiment classification model based on BERT-wwm-BiLSTM-SVM. Firstly, the BERT-wwm model is used to process word vectors on music review data. Obtain the characteristics of words and sentences, while avoiding ambiguity caused by word polysemy, then use BiLSTM to capture the context sequence information, and finally the output of BiLSTM is combined with SVM model for sentiment classification. ShaoPeng C[97] proposed an effectual Review Rating Prediction and Sentiment Classification was developed. Here, a Gated Recurrent Unit (GRU) was employed for the Sentiment Classification process, whereas a Hierarchical Attention Network (HAN) was applied for Review Rating Prediction.

In summary, deep learning technology for text data sentiment classification, compared with the technology based on sentiment lexicon and machine learning methods, the workload in text feature extraction is greatly reduced, and the classification performance has been significantly improved. Therefore, this paper also adopts deep learning technology in the research of text emotion classification.



Chapter 3 Methodology

3.1 Data Acquisition Technology

Currently, for microblog, there are two main data collection methods that are widely used. One is to utilize the API server provided by microblog for developers and directly call the API interface of microblog to complete data collection. The other is to use web crawler tools to parse the content of microblog pages to obtain data.

3.1.1 API-based Data Acquisition

The API-based data acquisition technique can collect the required data quickly and easily by directly interacting with the server's API. The data obtained by this method is usually returned in a formatted form, which makes parsing easier. This approach can significantly reduce development costs. The flow diagram for calling the API to get the data is shown in Figure 7.

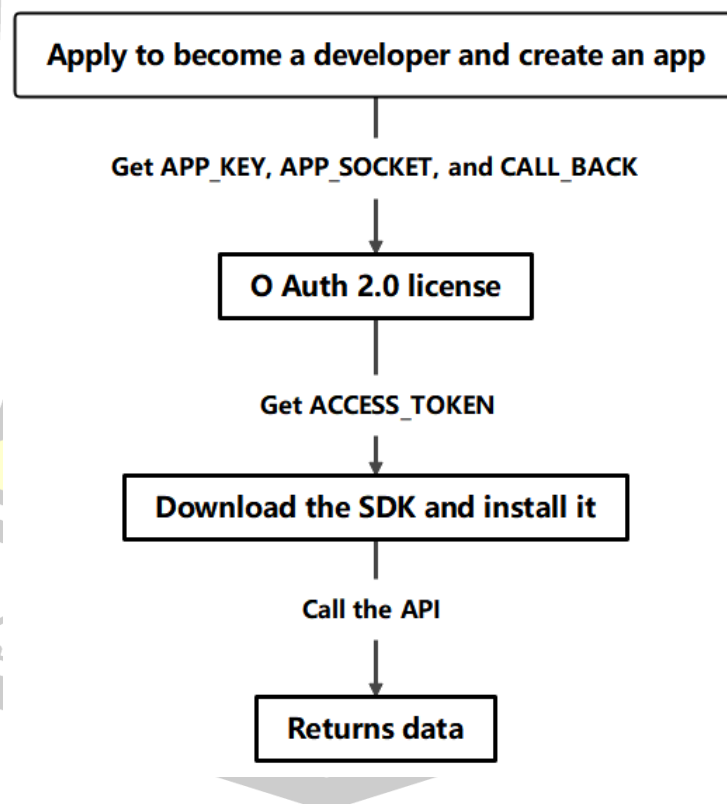


Figure 7 Flow chart of data acquisition based on API

To use the API call method to get data, first need to understand the API call process in detail. When using microblog API, the first step is to obtain user authentication, which is also known as user authorization. Currently, the main authentication method used by the microblogging open platform is OAuth 2.0, which is a commonly used authentication protocol. The operation mechanism of OAuth 2.0 is shown in Figure 8.

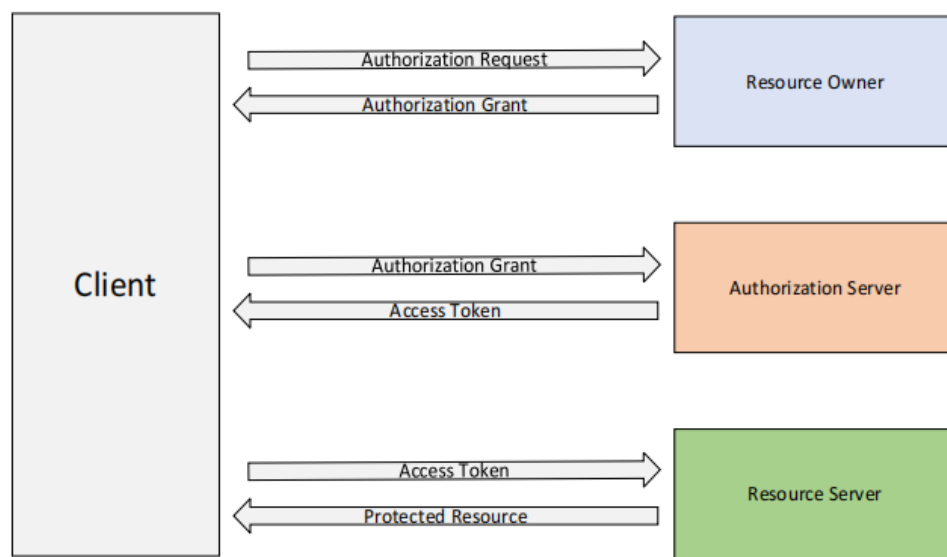


Figure 8 The operation mechanism of OAuth 2

Client is a third-party application developed by the user, which needs to be created and completed by the user; Resource Owner, which refers to the microblogging user, is the resource owner who owns the data; Authorization Server, which is the server that provides access to the API and is responsible for handling the user's authorization request; Resource Server, which is the server that stores the data and is also the target server that we access through API calls for data collection; Resource Server, which is the server that stores data and is also the target server that we access through API calls for data collection. Resource Server, this is the server that stores the data and is also the target server that we access through API calls for data collection.

It is clear from the flowchart that to achieve the goal, i.e. to access the protected resources in the Resource Server, the first step is to upload the Access Token to the Resource Server. The Access Token is returned to the user by the Sina Authorization

Server after the user has completed the authorization, and it is a necessary condition for the user's application to carry out the operation.

In order to obtain the user's permission to operate, the user's application first needs to send an authorization request to the user himself/herself. In response, the user will return an Authorization Grant, which the user's application needs to submit to the authorization server in order to obtain an Access Token and complete the authorization process. When interacting with the API, the Access Token plays a role similar to that of a flag, conveying to the resource server that the user's application has been authorized by the user and needs to access the data in the resource server. Once the user application submits the user's pass to the resource server, the resource server verifies the validity of the user's authorization. At this point, the entire process of accessing microblogging data from the API server is successfully completed.

The method of utilizing the API for access and obtaining data directly from the resource server is usually simpler in terms of engineering implementation. In addition, the data returned by API calls is often presented in a structured JSON format, so there is no need for additional fine-grained matching of the data, which greatly reduces the data processing time.

However, there are multiple limitations in obtaining data from microblogging resource servers via API calls. For example, there is a limit on the number of times a user with low privileges can access the resource server per day. In addition, the amount of data that can be obtained by a single API call is also limited, and users with low privileges can only obtain partial data and cannot obtain complete data coverage. Therefore, for users with lower privileges, it is difficult to obtain a large amount of data when calling microblogs using the API server.

3.1.2 Web Crawler Based Data Acquisition

A web crawler is an automated program or script that continuously extracts information from Internet sites according to a specific strategy. In other words, it utilizes programming to collect the required data on a Web page to achieve the goal of automated data crawling. In essence, a web crawler simulates the behavior of a

browser opening a web page in order to extract the required data from the web page. The basic workflow of a crawler is shown in Figure 9.

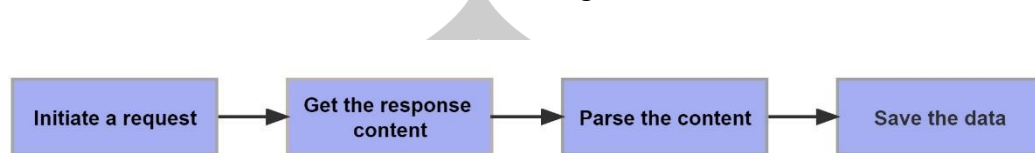


Figure 9 The basic workflow of a crawler

First, by using the HTTP library, we send a request to the target site, that is, we make a request. This request can contain additional header information, and once the request has been successfully sent, we wait for a response from the server. If the server returns an HTTP status code of 200, the request was successful and the server was able to respond properly. At this point, we get a response that contains the content of the page we need to retrieve. This response can be of various types, such as HTML, JSON strings, or binary data. Once the response is successfully fetched, we can choose to use regular expressions or page parsing libraries to disassemble the content and analyze it to extract the required data. Once the required data is obtained, we can choose to store it or process it further. The data can be stored in a variety of ways, either in a database, as a text file, or even as a file in a specific format. The form of storing data is very flexible, and you can choose the most suitable storage method according to the actual needs.

3.1.2.1 Data Crawling

Both requests and urllib are standard libraries for making HTTP requests in the Python programming language. They provide an easy way to send HTTP requests, fetch data from the network, and interact with a server. urllib is a module built into the Python standard library, divided into sub-modules such as urllib.request, urllib.parse, and urllib.error. urllib is available in Python 2.x and urllib has some different uses in Python 2.x and Python 3.x. requests is a third-party library that is more user-friendly and feature-rich than urllib. It encapsulates the details of HTTP requests in a more concise way, making it easier to send requests and process responses. the requests library supports a variety of HTTP methods (GET, POST, PUT, DELETE) and can handle cookies, headers, session hold, and more. In addition, it can also easily handle

JSON data and file uploads and other operations. In this paper, the requests library is used for data crawling.

3.1.2.2 Data Parsing

We can parse and extract valuable data from a web page or a new list of URLs with the help of a web parser (a tool used to parse web pages). Python supports a number of techniques for parsing web pages, namely Regular Expressions, XPath, BeautifulSoup, and JSONPath. for parsing text, there is Regular Expressions; for parsing HTML/XML, there is XPath, BeautifulSoup, and Regular Expressions; for parsing JSON, there is XPath and BeautifulSoup determine the path to a given node based on the hierarchical structure of the HTML/XML document, so they are better suited to handling data with a more obvious hierarchy. Regular expressions are based on the characteristics of the text to match or find the specified data, it can deal with any format of the string document, similar to the effect of fuzzy matching. JSONPath is specialized in data parsing of JSON documents. Python provides different modules or libraries to support different web parsing techniques, as shown in figure 10.

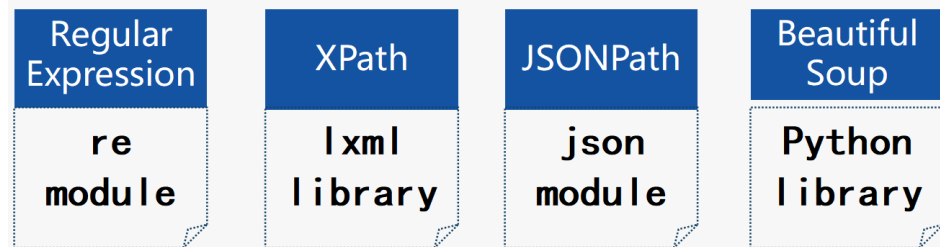


Figure 10 Different data parsing methods

The lxml library is a Python web page parsing library that supports HTML and XML parsing, and supports XPath parsing, which is very efficient. lxml library is not a built-in standard library in Python, so you need to install it before you use it, and the installation method is similar to that of the requests library. lxml library is not a built-in standard library in Python, so you need to install it before you use it, and the installation method is similar to that of the requests library. lxml library is also called BeautifulSoup4 library. XPath is an XML path language used to determine the location of a part of an XML tree structure. XPath technology is based on the XML tree structure, and is able to traverse nodes (elements, attributes, etc.) in the tree

structure. BeautifulSoup4 library, also known as the Beautiful Soup library or bs4 library, is used for parsing HTML or XML documents. BeautifulSoup4 library is not a built-in standard library in Python, and needs to be installed before using it. It needs to be installed before you can use it, in a similar way to the requests library. Regular expressions are powerful tools for working with strings. They use predefined patterns to match a class of strings with common characteristics, and are mainly used to quickly and accurately find and replace complex strings. In Python, the main use of re module to realize the operation of regular expressions. Javascript Object Notation, JSON is a lightweight text data exchange format. It represents data through a combination of objects and arrays, which is simple but highly structured. Python provides the json library to read and write JSON files. json library is a built-in standard library in Python, which can be used without additional installation.

3.1.2.3 Data Storage

Crawled data can be stored not only to the file, but also to the database. MySQL is a relational database, it will save the data in different two-dimensional table, each column in the table is a field, each line is a record; MongoDB is a non-relational document database, its data storage form similar to the JSON object, which can contain other documents, arrays of fields and arrays of documents, the form is very flexible. Python 3.x version provides pymysql library to connect to MySQL databases. PyMongo is used for MongoDB development tools, is the recommended way to operate the MongoDB database in Python. In this paper, the MongoDB database is selected.

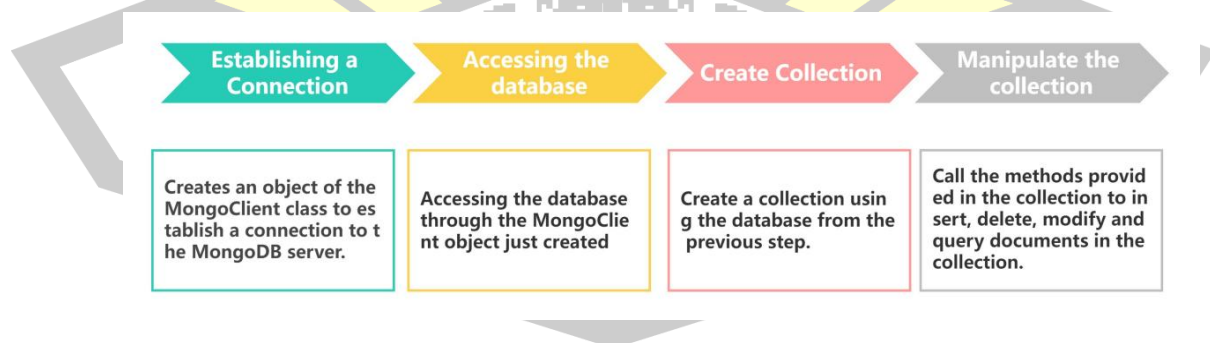


Figure 11 Basic flow of using the PyMongo library

3.2 Data Processing Technology

Before starting data mining, it is usually necessary to process the acquired data appropriately, a process known as data preprocessing, and it plays an important role in the whole data mining process. This is because most of the data collection methods may have some loose controls during the data collection process, which can lead to outliers in the data (e.g., negative income), irrational combinations of data in the real world (e.g., the animal species is "cat" and the fur color is "rainbow colors"), and missing data. Failure to carefully screen for these issues prior to data analysis can lead to misleading results. Therefore, not only is it necessary to have a large amount of data, but it is also necessary to ensure the quality of the data before analyzing it. This means that the data first needs to be processed to eliminate erroneous values and anomalies and to ensure the accuracy of the data. Through data preprocessing, data can be cleaned and normalized to make it more compatible with the needs of the analysis. This step is a critical part of ensuring the accuracy and reliability of data analysis. Therefore, it is essential to introduce the data preprocessing step before further analyzing the data.

3.2.1 Data Cleaning

The raw data we collect often contains a lot of irrelevant and noisy information, which is also called "dirty data". In microblogging data, these dirty data may include various characters such as "@", "V", etc., as well as a variety of emoticons such as " (; 'Д `)", "ε-( 'V ` ;)", "^ _ ^", etc., and even some common emoticons and meaningless URL links. These dirty data not only can't accurately reflect the user's real intention, but also interfere with text processing such as performing sentiment analysis, thus reducing the accuracy and processing efficiency of the sentiment classification model. In order to solve this problem, we need to clean and delete these dirty data. In Python, there are some tool libraries such as Pandas, Numpy, etc. that can help us with data processing and cleaning. In addition, regular expressions can be utilized to achieve the goal of data cleaning so that this extraneous information can be eliminated and the data can be made cleaner and more useful. Such processing can improve the quality and accuracy of subsequent analysis and ensure that we are focusing on what is truly meaningful.

3.2.2 Text Segmentation

In text mining, text disambiguation is a fundamental step. In some languages, textual disambiguation is necessary because it transforms a continuous stream of text into separate words or tokens. For example, English words have clear grammatical boundaries, so word segmentation is usually not necessary, but word form reduction may be needed because there are many forms of word changes in English. Chinese, on the other hand, must be subjected to lexical processing for text analysis because it has no clear intervals between words, and the text needs to be broken down into meaningful words. Three major problems that often occur:

(1) Recognition of new words: new words refer to the newly appeared words that are not recorded in the existing thesaurus. For example, in the sentence "李想的性格十分活泼", the name "李想" is a neologism, which cannot be accurately recorded in the thesaurus, and therefore may be incorrectly segmented as "李/想/的/性格/十分/活泼".

(2) Inconsistent word boundaries: some words have ambiguous word division, for example, "人民代表大会代表" can be divided into a separate word or "人民/代表/大会/代表". In this case, the words are cut in an inconsistent way, resulting in ambiguity.

(3) Ambiguity of cut words: Sometimes the position of cut words can lead to ambiguity, for example, "西安人多" can be divided into "西安/人多" or "西安人/多". Although these ways of word cutting are grammatically and logically correct, they are not always accurate. In this case, it is necessary to choose the most suitable mode of word segmentation by taking the context into consideration.

In conclusion, Chinese word segmentation is a challenging task that needs to take into account issues such as neologism recognition, word boundary inconsistency, and tangential ambiguity. In order to obtain more accurate results, we need to consider the context and select the appropriate lexical model according to the actual needs.

Currently, there are many excellent Chinese participle tools, including jieba participle, LTP participle of Harvard University, and Stanford CoreNLP of Stanford University, and in this paper, we choose to use jieba participle to realize the Chinese participle task. jieba participle is not only easy to call in the Python library, but also supports importing customized dictionary and removing deactivation, which makes it very powerful. jieba not only easy to call in the Python library, but also supports importing customized dictionaries and removing deactivated words, making it very powerful.

jieba provides four modes: Precise mode: cut the sentence precisely, mainly used for text analysis, suitable for detailed cutting of text. Full mode: recognizes all possible words in a sentence, faster, but may cause ambiguity. Search Engine Mode: based on the exact mode, secondary slicing of longer words to improve recall, suitable for applications such as search engines. Paddle mode: based on the domestic PaddlePaddle deep learning framework, it can further improve the effect by training the sequence annotation network model to realize the word segmentation.

In this paper, we choose to use the jieba participle tool, which is not only powerful in terms of functionality, but also easy to integrate into the Python environment, providing an effective solution for the task of participle in Chinese text.

3.2.3 Remove Stop Words

Deactivated words refer to words that have no actual meaning in the text, such as conjunctions, prepositions, auxiliaries, pronouns, numerals, and other words, such as "she", "he", "it", "the", "ah", "ah", "and", "and", as well as the numbers 0-9, the special symbols "¥", "#", "%", etc. Although these words appear frequently in the text, they are not always used in the text. Although these words appear frequently in the text, they do not help substantially in tasks such as text sentiment analysis, but may increase the burden of subsequent processing and have a greater impact on the processing time of the sentiment classification model. In order to improve the system performance, we need to eliminate these deactivated words from the text.

Nowadays, there are many well-organized and open deactivation word lists available for us to use directly, such as HIT and Baidu deactivation word lists are

common choices. Of course, we can also build a deactivation word list suitable for our own domain according to our own needs. In the jieba participle system, provides the relevant API interface, we only need to prepare their own deactivation file path into the corresponding code, you can automatically eliminate these deactivated words in the participle process. Doing so helps to reduce the complexity of processing and improve the efficiency and accuracy of the sentiment classification model.

3.3 Text emotion classification

3.3.1 The types of different emotions

On various social platforms, there are many emotional texts involved. We will analyze the authenticity of the product based on its comments and consider whether it meets our own needs; We will write blogs, brief books, and even establish our own websites to share what we have learned and felt; We will express our opinions and opinions on social focal issues through our social circle and Microblog. It contains countless emotional messages. In this emerging era, people's ways of expressing emotions have undergone tremendous changes not only in channels, but also in language habits and structures.

Emotional classification, proposed by Bo Pang et al., is an important task in the field of natural language processing. The research on sentiment analysis was first established on the basis of document level analysis. Pang and Snyder extended the bipolar classification of emotions to multi-level classification. Due to the continuous research, exploration, and innovation of many experts and scholars in this field, emotion classification related algorithms have become a hot research topic both domestically and internationally. Emotion classification algorithms mainly include sentiment classification algorithms based on dictionaries and rules, sentiment classification algorithms based on machine learning, and sentiment classification algorithms based on neural networks.

Emotional classification or opinion mining is a fundamental task in natural language processing (NLP), used to determine the polarity of text emotions. Binary classification, such as positive and negative emotions, can include fine-grained classification of emotions more generally. Natural language, like numbers, spreads and records information. The process of information dissemination and reception

corresponds exactly to the encoding and decoding of information, and text encoding and decoding are important links in natural language processing. On various emerging social platforms, people express their opinions in a variety of sentences, many of which cannot be strictly analyzed according to grammar and syntax rules, and sometimes even contradictory analysis may occur; The same word may have vastly different meanings in different contexts. In different eras, people have different language expressions and preferences, and analyzing the meaning expressed by words based on contextual context is more flexible and feasible.

The most basic analysis unit for text sentiment classification is sentence, while the smallest unit of a sentence is word [98]. Because there are language processing regions in the brain, it is easy for humans to make judgments about emotions. However, for computers to make judgments on the emotions of text, algorithms and rules must be designed for text sentiment classification, and various constraints must be proposed [99]. Text is different from images in that behind images are pixel values, while text is a product of human history and modern culture [100]. Therefore, it is necessary to symbolize sentences first.

For a sentence M with a length of L , to determine whether the sentence belongs to emotion category a , it is necessary to calculate the probability that sentence M belongs to emotion category a . Abstract the classification of text emotions into the following formula:

$$P(a|M). \quad (3.1)$$

Among them, there are two important tasks: representing sentence M and, calculating the probability p . The representation method of sentences directly affects the solution of classification probability, which relies on neural network design.

3.3.2 Discrete representation of words

To represent a sentence, the first step is to preprocess the sentence, including removing punctuation marks, segmenting the sentence into words, segmenting words based on spaces in English, and segmenting words in Chinese. After preprocessing, sentences are represented as a list of words. The initial representation method for words was single hot encoding, which required creating a word set for all words in the sentence, constructing a dictionary T for key value pairs, and recording $|T|$ as the number of words in the dictionary.

After unique encoding, each word is represented as a $|T|$ dimension vector, which is only 1 at the corresponding position in the word index, and all other positions are zero. Record the unique heat code as f , and the dimensional change process of sentence M is as follows:

$$R^L \xrightarrow{f} R^{L \times |T|}. \quad (3.2)$$

In actual text classification, due to the large size of the corpus and dictionary, it can be seen from Eq. (2) that the unique hot coding will make the sentence represented as a high-dimensional sparse matrix. Moreover, unique encoding cannot distinguish the importance of different words in a sentence, which makes it impossible to distinguish between high-frequency words and low-frequency words.

Using sentences as the basic unit code, a dictionary is established for sentences. The sentences are represented as $|T|$ dimensional vectors, and the frequency of corresponding words appearing in the dictionary is recorded at each position. The dimensional change process of sentence M is as follows:

$$R^L \xrightarrow{f} R^{|T|}. \quad (3.3)$$

3.3.3 Distributed table of words

The sentence representations constructed by DHC, BOW, and TF-IDF are sparse and high-dimensional. In the field of natural language processing, traditional machine learning algorithms are based on these representation methods. After the proposal of word2vec, sentence level sentiment classification algorithms have shifted from encoding with sentences as the basic unit to encoding with word embeddings as the basic unit. Word2vec is based on the distribution hypothesis: words with similar contexts have similar semantics.

The byproduct of the word2vec task is $W_{|T| \times H} \in R^{|T| \times H}$, which is the word vector of the dictionary, and according to the index of the dictionary, the word embeddings corresponding to the words can be found. For the unique hot code θ_i , its word embedding is:

$$W_i = \theta_i^T \cdot W^{|T| \times H}. \quad (3.4)$$

The entire process embeds words from high-dimensional sparse representation to low-dimensional dense representation. In a single hot coding space, two vectors are linearly independent and orthogonal; In word2vec embedding space,

there is a concept of distance between two vectors. This process maps words from high-dimensional, discrete, and sparse representations to low-dimensional, continuous, and dense representations.

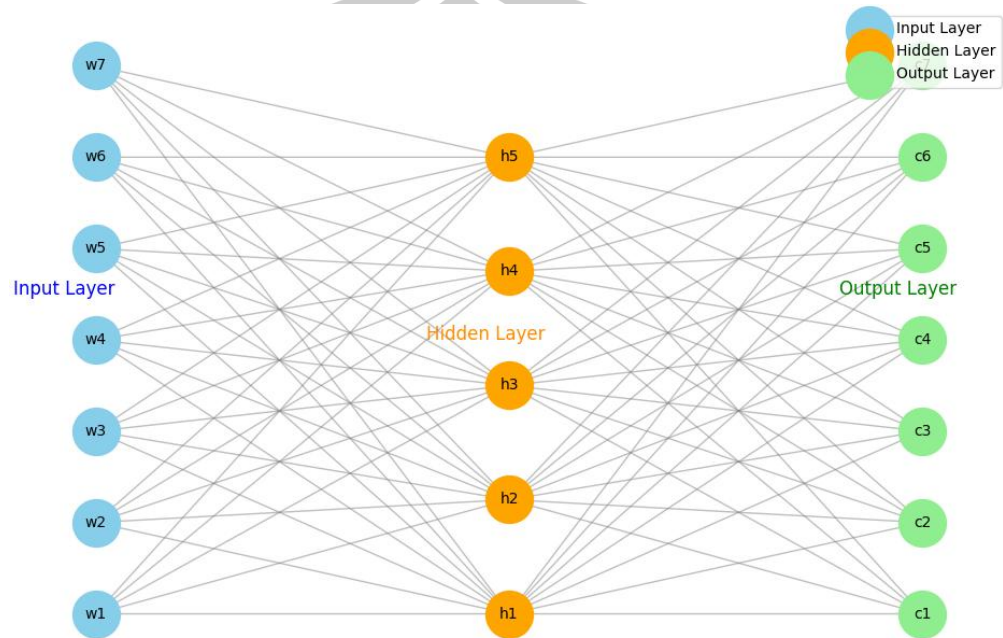
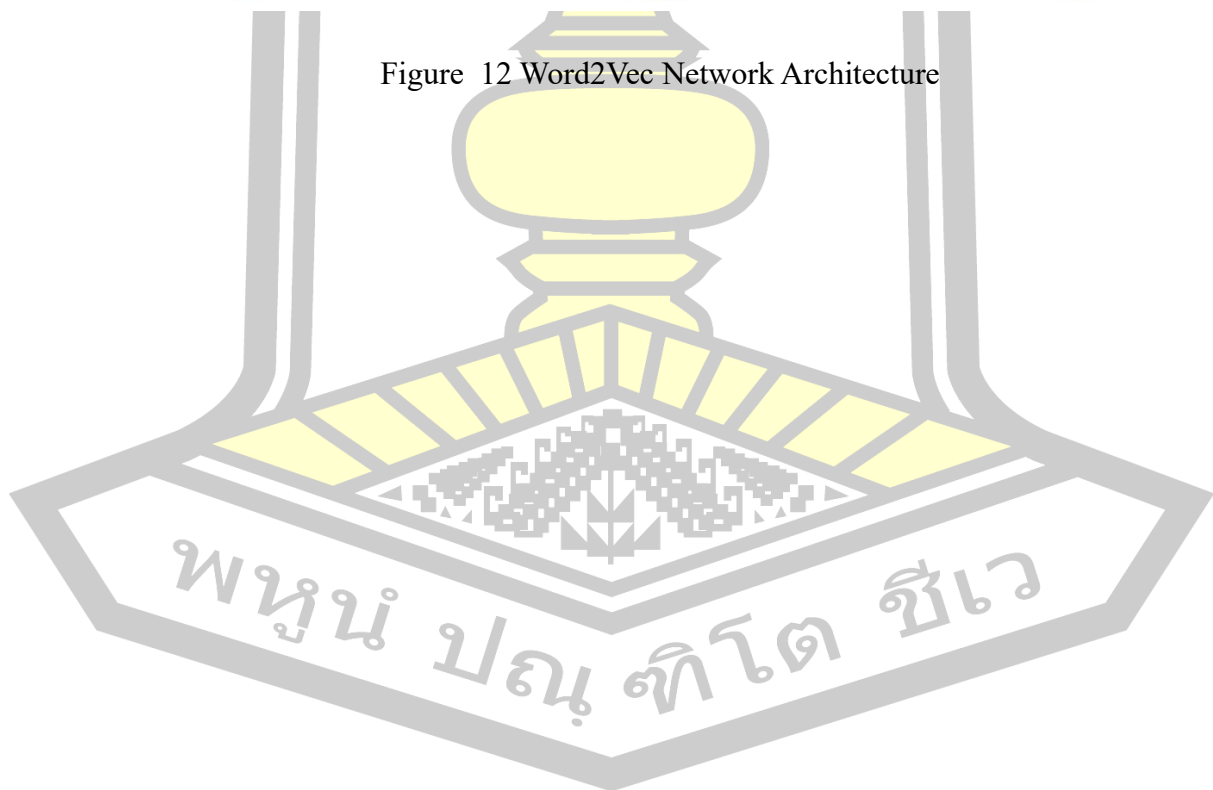


Figure 12 Word2Vec Network Architecture



Chapter 4 Improved Microblog Data Acquisition and Processing Methods

Web crawlers may encounter various anti-crawling difficulties when crawling microblog data, because websites take a series of measures to restrict crawler activities in order to protect data security and prevent malicious crawling.

(1) Login restriction: microblog restricts the access of non-logged-in users, and you need to use a logged-in account to access and get the data.

(2) CAPTCHA: microblog will require users to enter a CAPTCHA code when the number of accesses is high or abnormal. This can be handled automatically by using CAPTCHA recognition technology, but may increase the complexity of crawling.

(3) Dynamic rendering: microblog uses JavaScript to load content dynamically, making it difficult to crawl for direct access to complete data.

(4) IP blocking: microblog may monitor frequently requested IP addresses and block those considered to be crawlers.

(5) Request frequency limitation: microblog may limit the frequency of requests sent by the same IP within a short period of time.

In this paper, we propose an improved method for collecting and processing microblogging data with respect to login restriction, captcha, dynamic rendering, IP blocking, and request frequency limitation. The experimental environment is shown in Table 1.

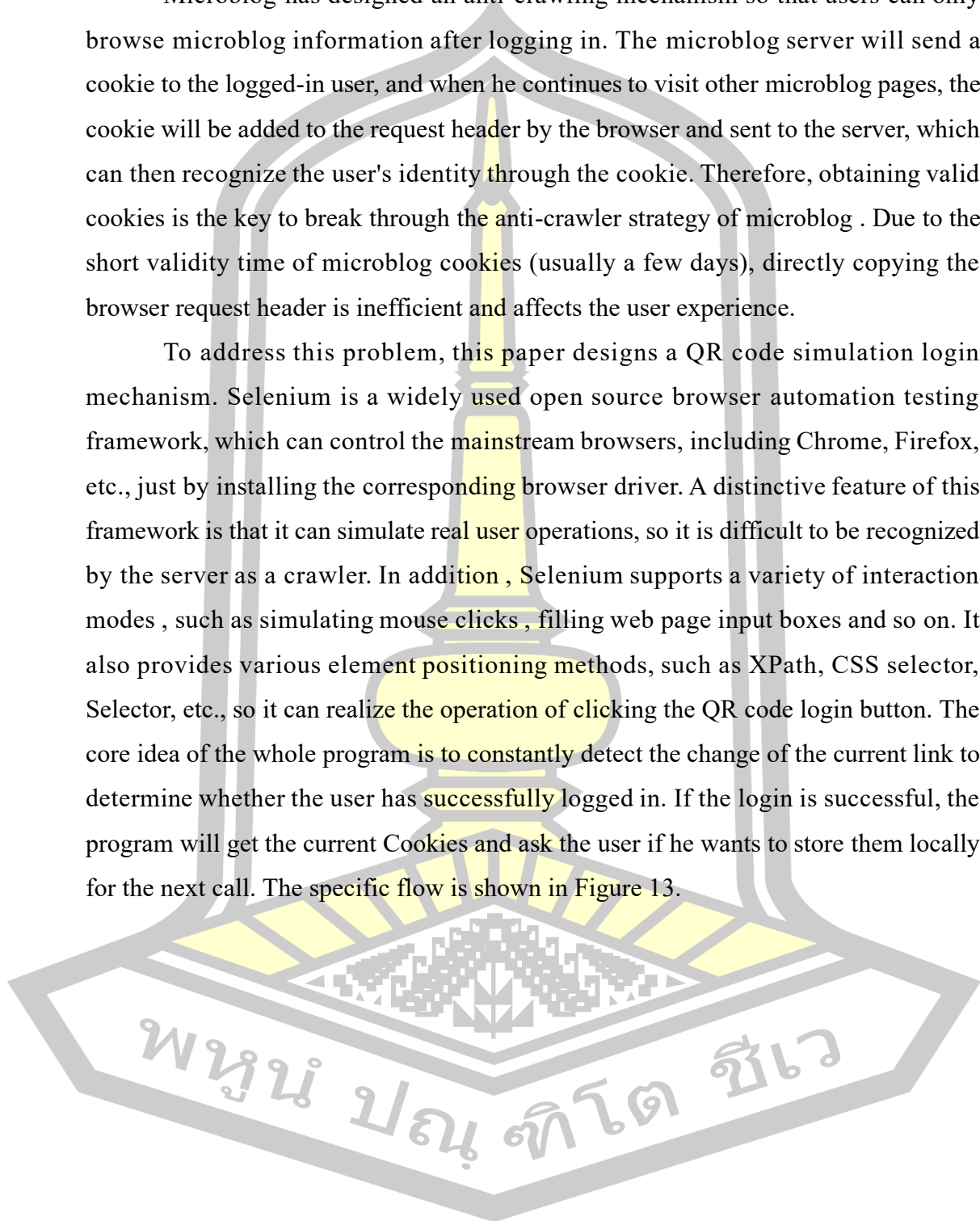
Table 1 Experimental Environment

Experimental Language	Development Tool	Operating System	Data Storage	Dependency Package
python	Pycharm	Windows	MongDB	os time undetected_chromedriver selenium seleniumwire pymongo wordcloud Jieba bs4

4.1 Microblog Simulation Login

Microblog has designed an anti-crawling mechanism so that users can only browse microblog information after logging in. The microblog server will send a cookie to the logged-in user, and when he continues to visit other microblog pages, the cookie will be added to the request header by the browser and sent to the server, which can then recognize the user's identity through the cookie. Therefore, obtaining valid cookies is the key to break through the anti-crawler strategy of microblog. Due to the short validity time of microblog cookies (usually a few days), directly copying the browser request header is inefficient and affects the user experience.

To address this problem, this paper designs a QR code simulation login mechanism. Selenium is a widely used open source browser automation testing framework, which can control the mainstream browsers, including Chrome, Firefox, etc., just by installing the corresponding browser driver. A distinctive feature of this framework is that it can simulate real user operations, so it is difficult to be recognized by the server as a crawler. In addition, Selenium supports a variety of interaction modes, such as simulating mouse clicks, filling web page input boxes and so on. It also provides various element positioning methods, such as XPath, CSS selector, Selector, etc., so it can realize the operation of clicking the QR code login button. The core idea of the whole program is to constantly detect the change of the current link to determine whether the user has successfully logged in. If the login is successful, the program will get the current Cookies and ask the user if he wants to store them locally for the next call. The specific flow is shown in Figure 13.



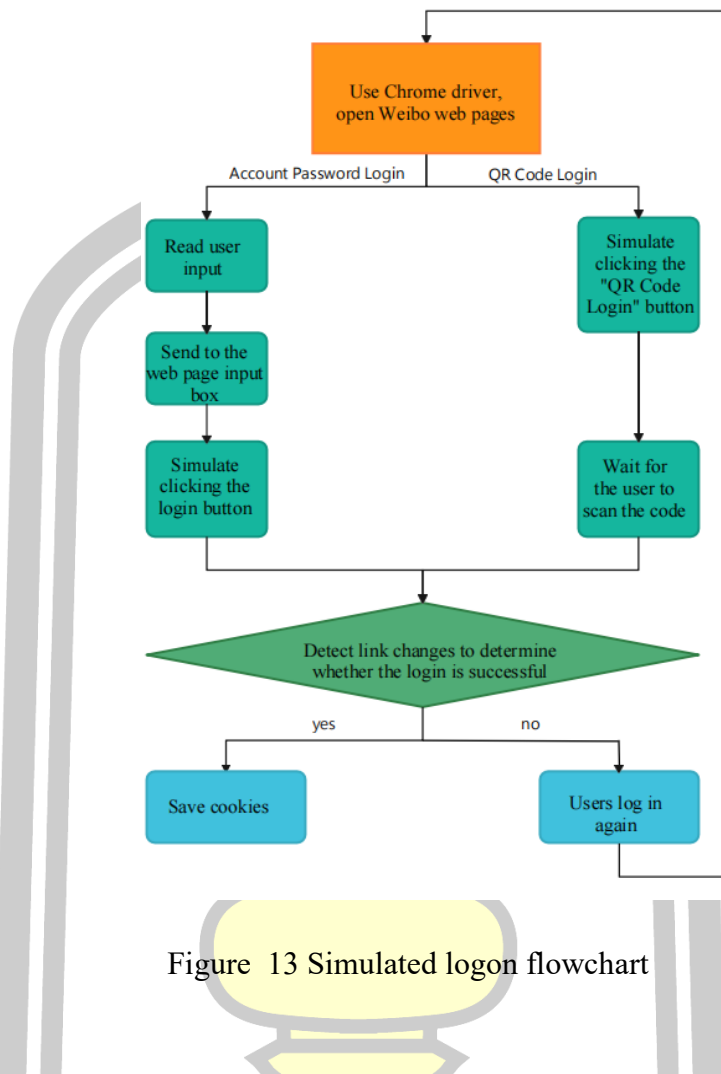


Figure 13 Simulated logon flowchart

The code is as follows:

```

def py_Microblog(self):
    self.driver.maximize_window()
    self.wait.until(expected_conditions.visibility_of_element_located((By.CLASS_NAME, "LoginCard_btn_Jp_u1")))
    self.driver.find_element(By.CLASS_NAME, "LoginCard_btn_Jp_u1").click()
    while True:
        try:
            locate = (By.CLASS_NAME, "Ctrls_avatarItem_3LrJN")

```

```

        self.wait.until(expected_conditions.visibility_of_element_lo
        cated(locate))

        break
    except:
        print("Please use your cell phone to scan the code to log in")

```

4.2 Microblog Data Crawling

Microblog has imposed access frequency limits on each IP address. To circumvent this restriction, we can create an object of the ChromeOptions class from the Selenium library. Within this object, we can utilize the `add_argument()` method to input properly formatted and valid proxy IP addresses. Once this configuration is applied to the corresponding browser driver settings, we can access web pages using the proxy IP addresses. By employing multiple distinct proxy IP addresses, we can effectively evade anti-scraping measures, thereby enhancing program stability and extraction efficiency. The proxy functionality is disabled by default, but it can be enabled by initializing it through the constructor. By modifying the `agent_ip` parameter to True and supplying the proxy IP address, the proxy functionality can be utilized seamlessly.

The code is as follows:

```

def __init__(self, agent_ip=False, proxy=""):
    if agent_ip:
        if proxy != "" and ":" in proxy:
            chrome_options = webdriver.ChromeOptions()
            chrome_options.add_argument('--proxy-server='+ proxy)
            self.driver = uc.Chrome(options=chrome_options)
        else:
            print("Please fill in the correct proxy ip format : ip:port")
            os.system(exit(0))

```

else:

```
self.driver = uc.Chrome()
```

Using the proxy IP to access the microblog home page, and logged in successfully, into the logged in user home page, through the cycle selenium corresponding to the browser driver to provide the `execute_script()` method, run the JavaScript code, simulate the page to scroll down, and constantly get the new html source code, and then through the `page_source` attribute to get the source code of the page, and at the same time, run the data parsing method that has already been written, `wb_data_obtain()`, to carry on the data parsing, and ultimately complete the crawling of the data.

The code is as follows:

```
def get_wb_data(self, rate=5):
    username=(By.CSS_SELECTOR, ".ALink_default_2ibt1.head_cut_2Zcft.head_n
ame_24eEB>span")
    self.wait.until(expected_conditions.visibility_of_element_located(username))
    for i in range(rate):
        self.wb_data_obtain(self.driver.page_source)
        self.driver.execute_script(f"window.scrollTo(0,{i*3000})")
        time.sleep(0.5)
    self.driver.close()
    return self.data_list
```

4.3 Microblog Data Parsing

Define the content of this crawler: username-username, content-content, time-publishing time, forward-forwards; comment-comments; like-likes. After analyzing the microblog pages, it is found that the information related to the users who published the microblogs are all under the same tag of the class name, and the information related to the content of the microblogs are also under the same tag. The webpage source code for the username is shown in Figure 14.

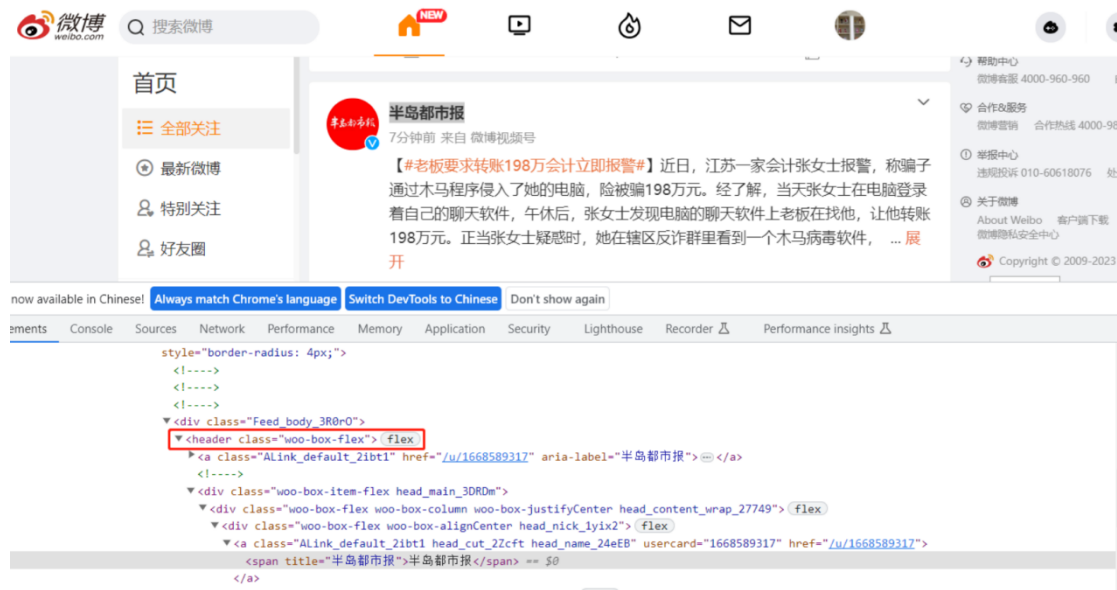


Figure 14 The webpage source code for the username

Further analysis of the label, found that it is divided into two categories, one is the required user name, release time, these are in a class named "woo-box-flex" under the header tag, the content of a class named "detail_wbtext_4CRf9" under the div label, other retweets, comments and likes are in a footer tag, and the value of an attribute aria-label in the label just corresponds to the value of the "woo-box". 4CRf9" div label, the other retweets, comments and likes in a footer label, and the value of an attribute aria-label in the label just corresponds to the value of "forwards", "comments" and "likes"., "comments" and "likes", so you only need to get the value of the aria-label attribute in the corresponding footed label, you can see that the three values are separated by ",", so use .split(",") method to split the values into a list, and then extract the corresponding values by looping through the list and depositing them into their respective lists. The BeautifulSoup library is used here to parse the content and extract it. The webpage source code for the repost is shown in Figure 15.



Figure 15 The webpage source code for the repost

The code is as follows:

```
bs = BeautifulSoup(self.driver.page_source, 'lxml')

username = bs.select(".ALink_default_2ibt1.head_cut_2Zcft.head_name_24eEB
>span")

content = bs.select(".detail_wbtext_4CRf9")

time = bs.select(".head-info_time_6sFQg")

footer = bs.select("footer")

forward = []

comment = []

like = []

for i in footer:

    string = i["aria-label"]
```

```
list = string.split(",")

forward.append(list[0])

comment.append(list[1])

like.append(list[2])
```

In order to facilitate the storage of data as well as the beauty and accuracy of the extracted data need to be integrated, the use of .strip () method to process the data, remove spaces, returns and other useless characters, filter out duplicate data, and finally the processed data to the json format to store the data_list, to complete the integration of data. Then the data in the "content" is saved separately in the "content.txt" local file to facilitate subsequent operations.

The code is as follows:

```
for i in range(min(min(len(username), len(content)), len(time))):

    a = True

    content_a = content[i].text.strip().replace("\n", "")

    if len(self.data_list) != 0:

        for j in self.data_list:

            if username[i].text == j["username"] and content_a == j["content"]:

                a = False

                break

    if a:

        if username[i].text != "" and content[i].text != "" and time[i].text != "":

            self.data_list.append(

                {

                    "username": username[i].text,
```

```

        "content": content_a,

        "time": time[i].text,

        "forward": forward[i],

        "comment": comment[i],

        "like": like[i]
    }
)

with open("content.txt", "a+", encoding="utf-8") as f:
    f.write(content_a)

```

4.4 Microblog Data Storage

In this paper, the crawled data is stored in the database, and the crawled related microblogging data and other contents are stored in MongoDB located in the cloud, so as to facilitate the later data analysis and data mining. Specific operations are as follows: create a MongoClient class object client, establish a connection with the MongoDB server; access the database through client.pydata; use db.pyinfo to create a collection; call the insert_many method in the collection to insert the crawled data data_list into the collection. The crawled data is shown in Figure 16.

The code is as follows:

```

client = pymongo.MongoClient(host='localhost', port=27017)
db = client.pydata
collection = db.pyinfo
collection.insert_many(data_list)
client.close()

```

	_id ObjectId	username String	content String	time String	forward String	comment String
1	ObjectId('64ec1e53e98a92a7d77...	"填鸭式教育受害者"	"网友投稿：商场买零食发现日本品...	" 15小时前 "	"10"	"45"
2	ObjectId('64ec1e53e98a92a7d77...	"百车全说Autotalk"	"奥迪Q2L，男士勿入？"	" 2小时前 "	"24"	"52"
3	ObjectId('64ec1e53e98a92a7d77...	"皮肤医生康凯"	"在我们身边总有那么一些人，他们...	" 8-26 11:00 "	"0"	"1"
4	ObjectId('64ec1e53e98a92a7d77...	"大飞Music"	"很多粉丝都在私信问#法郎#的价格...	" 2022-9-29 "	"308"	"202"
5	ObjectId('64ec1e53e98a92a7d77...	"傻子歪歌"	"有疤很烦？疤痕#修护#贵吗？如果...	" 2022-9-24 "	"143"	"152"
6	ObjectId('64ec1e53e98a92a7d77...	"撸车师兄"	"2023 BIRTV展 主要是去见见各位...	" 8-24 22:22 "	"1"	"2"
7	ObjectId('64ec1e53e98a92a7d77...	"大飞Music"	"2023 BIRTV展 主要是去见见各位...	" 8-24 22:22 "	"1"	"2"
8	ObjectId('64ec1e53e98a92a7d77...	"大眼看电影"	"很多粉丝都在私信问#法郎#的价格...	" 2022-9-29 "	"308"	"202"
9	ObjectId('64ec1e53e98a92a7d77...	"撸车师兄"	"有疤很烦？疤痕#修护#贵吗？如果...	" 2022-9-24 "	"143"	"152"
10	ObjectId('64ec1e53e98a92a7d77...	"傻子歪歌"	"啊啊啊啊啊啊虽然有预感到，该...	" 8-25 09:52 "	"40"	"31"
11	ObjectId('64ec1e53e98a92a7d77...	"思想聚集"	"为什么打孔划线的刹车盘，越来越...	" 8-25 09:39 "	"2"	"0"
12	ObjectId('64ec1e53e98a92a7d77...	"望京甜不辣"	"地图什么呢 "	" 23小时前 "	"15"	"225"
13	ObjectId('64ec1e53e98a92a7d77...	"高山青VIP"	"你的大学老师是你够不到的高度 "	" 昨天 09:00 "	"1140"	"716"

Figure 16 The crawled data

4.5 Microblog Data Processing

During the data crawling process, a lot of irrelevant and noisy information is collected, such as emoticons and meaningless URL links, which are called "dirty data". These dirty data can interfere with subsequent text processing tasks, such as sentiment analysis, and reduce the accuracy and efficiency of the model. In order to deal with this problem, we need to perform data cleaning and preprocessing, which mainly includes the steps of word segmentation, deletion of deactivated words and word frequency statistics. Among the many lexical libraries, jieba lexical library is the most effective and powerful. It provides three modes of participles: precise mode (removing redundancy), full mode (returning all possible participles), and search engine mode (second cutting of long words on the basis of precise mode), which can satisfy a variety of participles' needs. Since microblogging adopts network language, there are many meaningless stop words, the program in this paper will read the pre-compiled stop word dictionary: ChineseStopWord.txt to clean the word list returned by jieba, and keep only the key words. The program will read the preorganized stop word dictionary: ChineseStopWord.txt to clean the word list returned by jieba and keep only the key words. The flowchart of the preprocessing module is shown in Figure 16. The data after the segmentation process is shown in Figure 17.

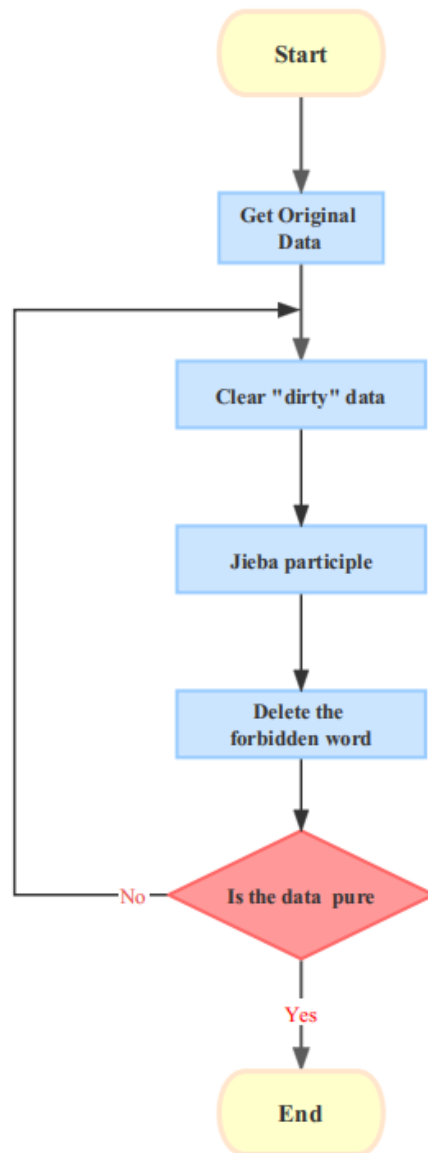


Figure 17 The flowchart of the preprocessing module

The code is as follows:

with open("content.txt", "r", encoding="utf-8") as f:

text = f.read()

jieba_list = jieba.lcut(text)



Figure 18 The data after the segmentation process

4.6 Microblog Data Visualization

The visualization of text data with the help of word clouds is especially suitable for displaying keywords in microblog posts. Word clouds can present keywords in a visual way so that users can quickly gain insight into the social hotspots and related information involved in that keyword. This is achieved by using the WordCloud library in Python, which is capable of automatically counting word frequencies, and it only requires the developer to pass the processed keyword data in string format to the WordCloud function to generate a word cloud map. At the same time, this library also gives the user the ability to customize the shape of the word cloud map, text hue, text size and other parameters, so as to meet different visualization needs. In this paper, we generate a word cloud map by further processing the processed keywords through the WordCloud function. This map will show the importance and frequency of keywords visually, providing users with a clear overview, which helps to better understand and convey the social hotspots in microblogs. The data visualization is shown in Figure 19.



Figure 19 The data visualization

The code is as follows:

with open("content.txt", "r", encoding="utf-8") as f:

```
text = f.read()
```

```
jieba_list = jieba.lcut(text)
```

```
ww = wordcloud.WordCloud(
```

```
background_color='white',
```

width=1280,

height=720,

```
max_words=6000,
```

```
min_font_size=10,
```

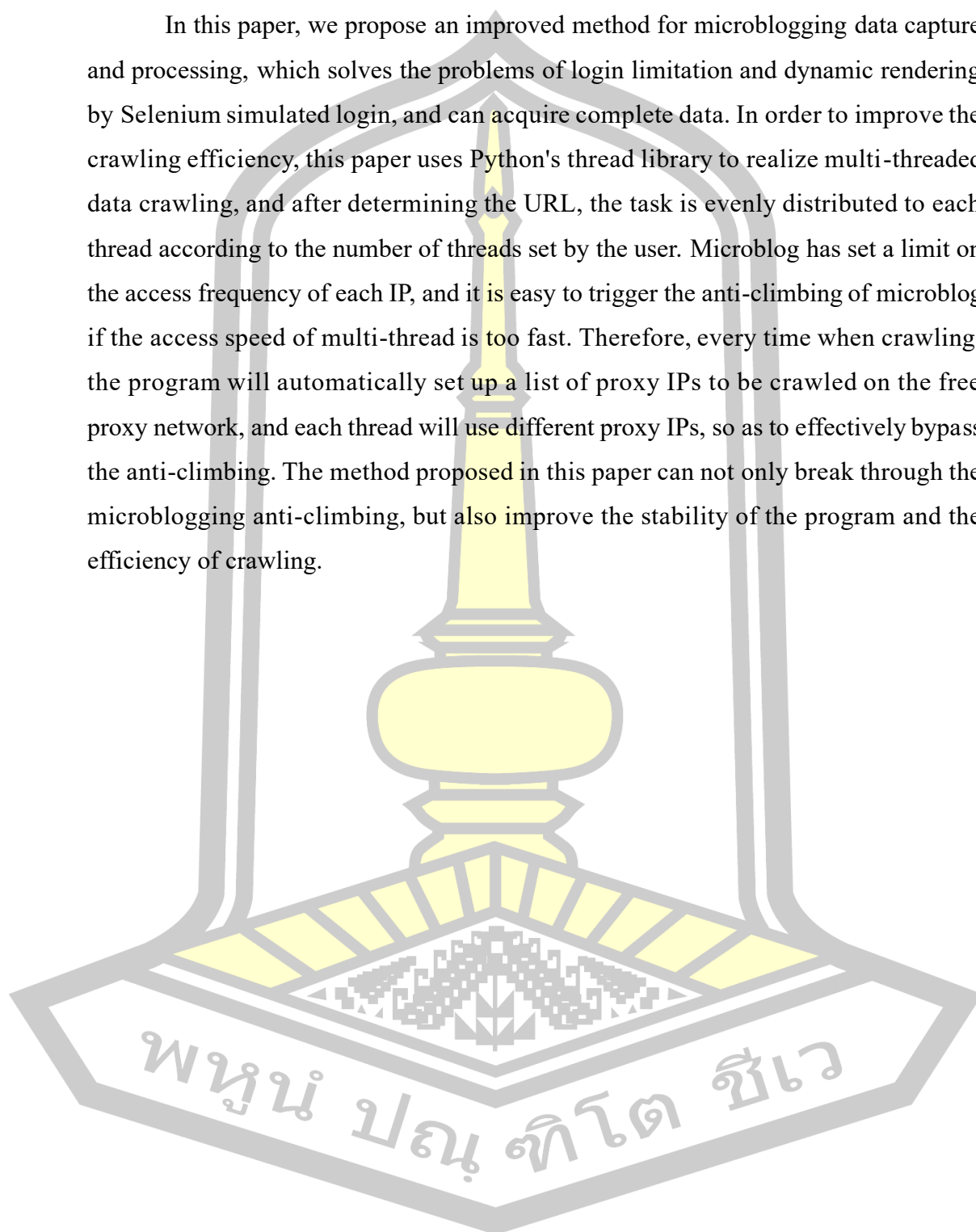
```
font_path="YeZ.ttf"
```

)

```
ww.generate(" ".join(jieba_list))
```

ww.to_file('wb.png')

In this paper, we propose an improved method for microblogging data capture and processing, which solves the problems of login limitation and dynamic rendering by Selenium simulated login, and can acquire complete data. In order to improve the crawling efficiency, this paper uses Python's thread library to realize multi-threaded data crawling, and after determining the URL, the task is evenly distributed to each thread according to the number of threads set by the user. Microblog has set a limit on the access frequency of each IP, and it is easy to trigger the anti-climbing of microblog if the access speed of multi-thread is too fast. Therefore, every time when crawling, the program will automatically set up a list of proxy IPs to be crawled on the free proxy network, and each thread will use different proxy IPs, so as to effectively bypass the anti-climbing. The method proposed in this paper can not only break through the microblogging anti-climbing, but also improve the stability of the program and the efficiency of crawling.



Chapter 5 A CNN-BiLSTM algorithm for Microblog emotion classification with attention mechanism

5.1 The overview of CNN-BiLSTM algorithm

5.1.1 CNN Network

As a special type of feedforward neural network, CNN has been widely used in the field of natural language processing by scholars in recent years [101]. Its basic structure is divided into three parts: input layer, convolutional layer and pooling layer, and fully connected layer. Convolutional layer feature extraction first represents the text in the form of a word vector matrix, and then scans through convolutional check matrices of different sizes [102]. During the scanning process, the parameter values of the filter composed of convolutional kernels are fixed and unchanged. After filtering, a new feature map is mapped, and all elements on the feature map come from the filter with consistent parameters. The CNN network architecture is shown in Figure 20.

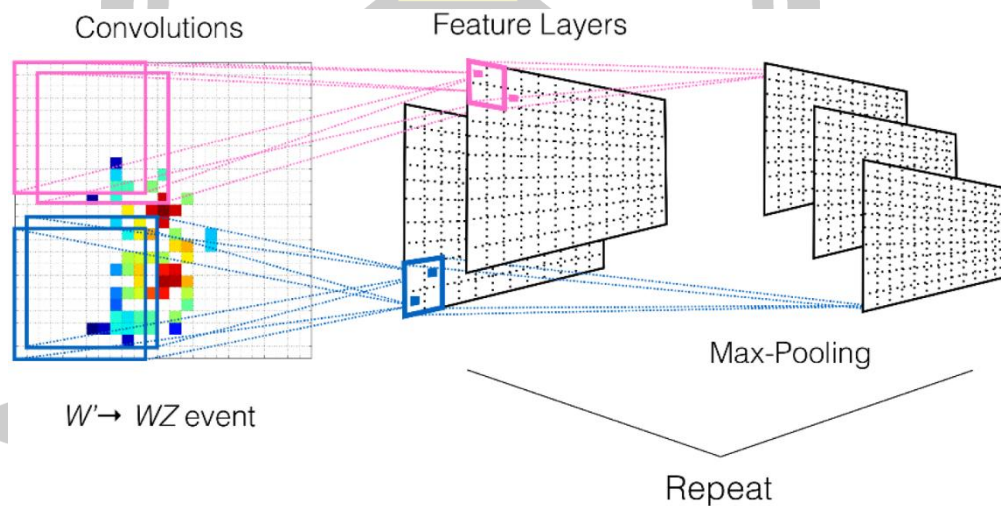


Figure 20 CNN network architecture

5.1.2 BiLSTM neural network

LSTM is a special type of recurrent neural network (RNN) composed of cell units and three gates. Cell units are the core computing power that records the current computing state, while forgetting gates, input gates, and output gates regulate the information flow in and out of the storage unit [103]. forgetting gates

clear useless information from the storage unit, input gates select input information from the current storage unit, and output gates determine the final output of the information. When conducting sentiment analysis on Microblog texts, it is often necessary to consider the impact of contextual semantic features on the overall emotional state of the text. However, the general LSTM model only focuses on unidirectional semantic relationships in the text, which ignores the impact of semantic features on the overall state. The Bi LSTM model is composed of LSTM networks in both positive and negative directions, which fully capture contextual information based on two different orders and can mine more comprehensive text semantics. The BiLSTM network architecture is shown in Figure 21.

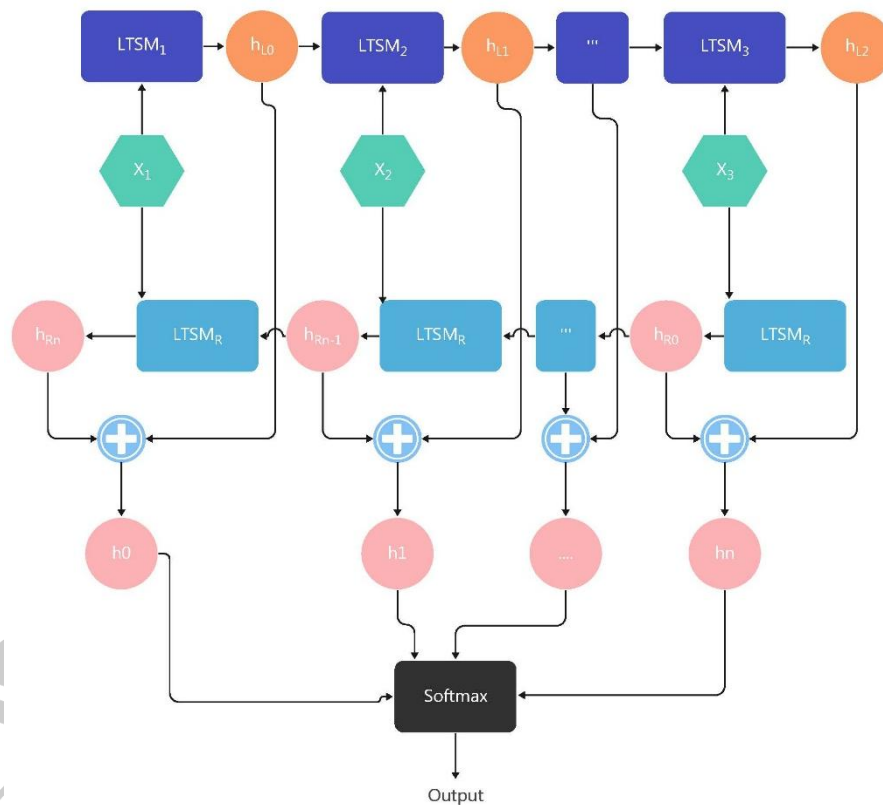


Figure 21 Bi LSTM network architecture

5.1.3 The joint mechanism of CNN-BiLSTM algorithm

This article constructs a Microblog sentiment classification model based on convolutional neural network CNN and bidirectional short-term memory network Bi

LSTM. On the basis of Bi LSTM and CNN, an attention mechanism was introduced, and a multi-channel model combining convolutional networks and bidirectional long and short term networks was proposed for sentiment polarity analysis of Microblog texts [104]. CNN-BiLSTM Joint Mechanism is shown in Figure 22.

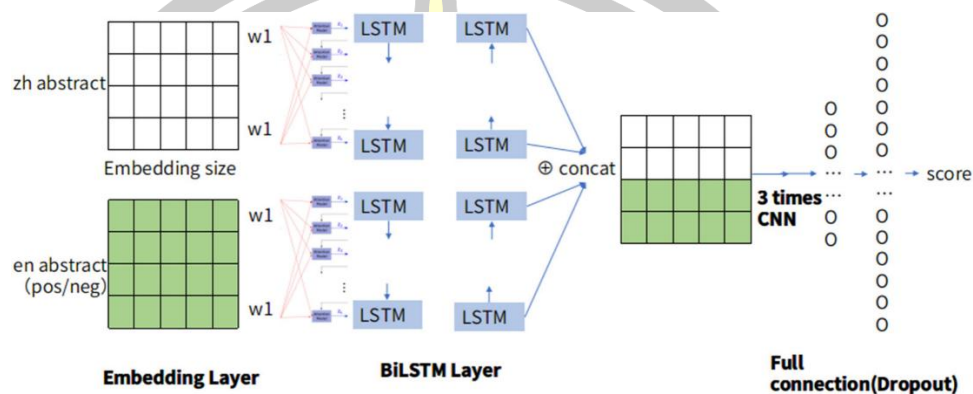


Figure 22 CNN BiLSTM joint mechanism

5.2 Attention mechanism

Emotional classification, as an important task in the field of NLP, aims to determine the emotional polarity of text, including positive and negative emotions [105]. After reading a large number of references, this article found that improving the accuracy of emotion classification algorithms and determining whether the classification algorithm has good performance can be achieved from three aspects: whether the algorithm has the ability to combine context, and how much context can be learned; Does the algorithm have attention and can it focus on learning; Does the algorithm have memory and store sequential information.

The attention mechanism originates from human research on vision [106]. Due to the bottleneck in information processing, humans selectively focus on some information while ignoring others, a mechanism known as attention mechanism. When used for sentence level text sentiment classification, the attention mechanism is usually placed at the decoding end, which weights and sums the input sequence over the entire time range.

The existing sentiment classification algorithms based on neural networks mainly include sentiment classification algorithms based on recurrent neural

networks, sentiment classification algorithms based on convolutional neural networks, and sentiment classification algorithms based on attention mechanisms. The self-attention mechanism belongs to a special attention mechanism, which can serve as both an encoder and a decoder. When the sentence length is long, even LSTM may experience the phenomenon of gradient vanishing and information dissemination being hindered. In this case, using self-attention mechanism as the encoder can achieve cross distance learning, and the classification effect is better than RNN. The attention mechanism can be expressed as the following formula:

$$A(Q, S) = \sum_{i=1}^L S(Q, K_i) \times V_i \quad (5.1)$$

where, S refers to the input information, including sentence M and label y, Q as the query vector, K and V appear in pairs, S(Q, K) refers to the attention weight value of V. A(Q, S) is the attention force vector weighted and summed by the attention mechanism. This formula represents the way the attention mechanism decodes sentence M. The attention mechanism architecture is shown in Figure 23.

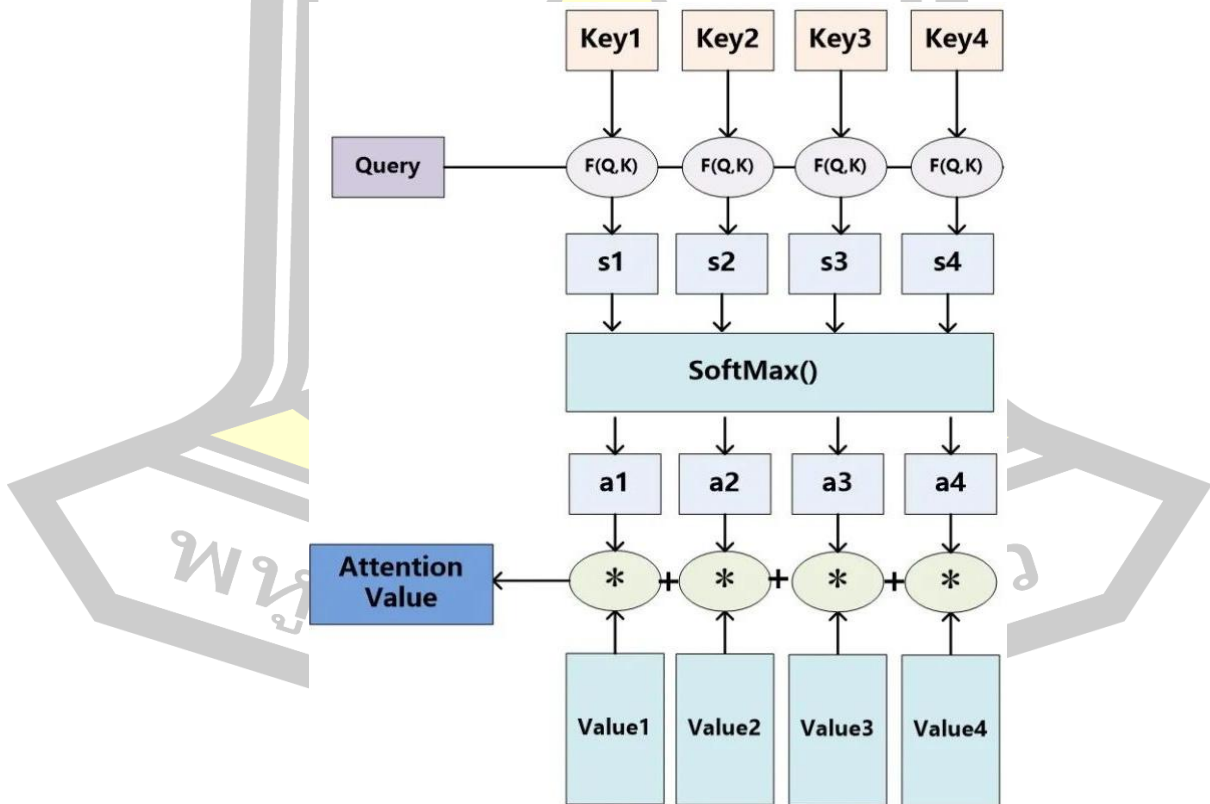


Figure 23 The attention mechanism architecture

5.3 Data sources and processing

5.3.1 Construction of Microblog dataset

The data crawled includes usernames, Microblog content, etc. Delete the usernames and retain only the Microblog content. Based on this dataset, a new Microblog sentiment polarity dataset was constructed. Partial Microblog text data was selected from the dataset and sentiment polarity was annotated manually. To ensure the quality of data annotation, cross and repeated labeling was used, and text with inconsistent labeling in two rounds was deleted. A total of 116922 pieces of data were organized. In order to avoid unsatisfactory experimental results due to imbalanced sample sizes, a total of 58461 text data with positive sentiment expressions and 58461 data with negative sentiment expressions were collected. The labeled corpus dataset achieved a uniform distribution of positive and negative sentiment polarity in the samples.

5.3.2 Dataset preprocessing

Data preprocessing is an indispensable preliminary work for text sentiment polarity analysis, which mainly includes data filtering, text segmentation, and removing stop words. In the constructed labeled corpus dataset, there are many non-textual data, labels, and special characters. Considering optimization efficiency, storage space is saved, and word vector representation accuracy is improved. Therefore, it is necessary to remove these useless information and filter it mainly through regular expression matching.

5.3.3 Training process

In the model proposed in this article, the Input input layer is the first layer, which reads the cleaned dataset. The Microblog text data processed by word segmentation is passed into the second layer Embedding layer. During the vectorization process, the Word2Vec model is used, and the Embedding layer embeds the corresponding vector based on the incoming words, which is then transmitted into the multi-channel model. The third layer of the model is a multi-channel feature extraction layer, which enters the BiLSTM model channel and can simultaneously capture semantic information in both positive and negative directions. Multiple CNN models are designed to extract local features of sentences at different scales, and

using convolutional kernels of different sizes can better extract features from different dimensions to achieve semantic information extraction.

The word vector dimension trained using the Word2Vec model is 350, and the filters for the three convolutional channels are 1×350 , 2×350 , and 3×350 , respectively. The pooling layer adopts Max Pooling operation to eliminate weak features. The BiLSTM model adopts L2 regularization processing to control the model complexity and avoid overfitting as much as possible. Introducing attention mechanism in the fourth layer to extract more important feature information from each channel; The splicing layer summarizes the features output by all channels to obtain richer feature information; The summarized features are passed into the fully connected layer, and a random deactivation mechanism is added to reduce redundancy and improve the model's generalization ability. Finally, the sentiment polarity category of the text is determined based on the Softmax classifier in the output layer.

5.4 Experimental results and analysis

5.4.1 The benchmark model

To evaluate the effectiveness of the algorithm proposed in this article, some benchmark models will be selected for comparison in the same experimental environment. The benchmark model selected in this article is as follows.

- 1) Logistic Regression Model (LR): It mainly solves binary classification problems and is a classic machine learning classification model.
- 2) Support Vector Machine (SVM): Use TF-IDF to represent text words, and use SVM algorithm to determine emotional polarity.
- 3) LSTM and Bi LSTM models: Use LSTM to capture unidirectional semantics or Bi LSTM to extract sentence context features, and use softmax classifiers for sentiment analysis.
- 4) CNN-Att and BiLSTM-Att models: CNN extracts local feature information from text, or uses Bi LSTM to obtain contextual semantic information, and then calculates the attention weight of the model's output features using Attention. Finally, it enters the fully connected layer and outputs with the classifier.

5) CNN-BiLSTM model: The segmented sentences are vectorized through word embedding, and then input into the CNN. The extracted local features are further input into Bi LSTM, and the final result is obtained through a classifier.

Table 2 Experimental comparison results

Algorithm	Accuracy (%)	Recall (%)	F-value (%)
LR	76.85	76.21	76.39
SVM	81.14	80.76	80.95
LSTM	83.19	82.89	83.05
CNN-Att	83.84	83.50	83.70
Bi LSTM	84.97	84.50	84.83
Bi LSTM-Att	87.08	86.99	87.02
CNN-Bi LSTM-Attention	88.58	88.44	88.52

The experimental results of the proposed algorithm and several other neural network model methods on Microblog sentiment dataset are shown in Table 1. From the experimental results, it can be seen that traditional machine learning models such as LR and SVM have poor experimental results. LR models are relatively simple, and their models themselves cannot perform feature filtering, resulting in poor data fitting performance; Although the classification performance of the SVM model is better than that of the LR model, it only performs a single weighted average for the word vector information in the sentence. Therefore, traditional machine learning methods are not suitable for the current needs. In the Microblog sentiment polarity analysis experiment, it is evident that the test results of the deep learning model are superior to traditional machine learning models. The experimental results of the Bi LSTM model have higher accuracy compared to the LSTM model, indicating that bidirectional units have significant advantages in serialized data processing. After incorporating attention mechanism into the CNN network, its classification performance is better than that of the LSTM model. The CNN network learns the word vector representation of text through convolutional window sliding, which can effectively extract the overall features of sentences. Moreover, the attention

mechanism can focus on semantically related vocabulary in declarative memory, better capturing emotional words and assigning them higher weight values.

It can be seen that the CNN BiLSTM algorithm has more advantages over traditional machine learning algorithms in grasping Microblog text features. The text features extracted by this algorithm are more comprehensive than the information captured by a single CNN or BiLSTM network. The introduction of attention mechanism makes the model pay more attention to the emotional part of the text, and the allocation of weight information makes the text representation more rich in emotional feature information. This advantage is also reflected in the comparison between BiLSTM and Bi LSTM-Att models, where the text representation is consistent, but the model has been effectively improved under the attention mechanism.

5.4.2 The optimality of the architecture

To further explore the advantages of the design of each part of the model proposed in this article, ablation experiments were conducted. The proposed model was decomposed and BiLSTM network, multi-channel CNN network, and Attention mechanism were removed to verify the effectiveness of the removed part. To control the influence of parameters on the experiment, the hyperparameters were set the same for each group of experiments. The experimental results are shown in Table 3 (w/o represents without).

Table 3 Ablation test results

Model	Accuracy (%)	Recall (%)	F-value (%)
w/o word2vec	87.91	87.92	87.91
w/o CNNs	86.87	86.83	86.87
w/o BiLSTM	89.26	89.11	89.08
w/o Attention	89.25	89.18	89.13
CNN-Bi LSTM-Attention	89.44	89.42	89.39

From the experimental results, it can be seen that each part of the model's structure significantly improves the performance of the model. Under the same conditions, the word embedding layer trained with Word2Vec has an improved

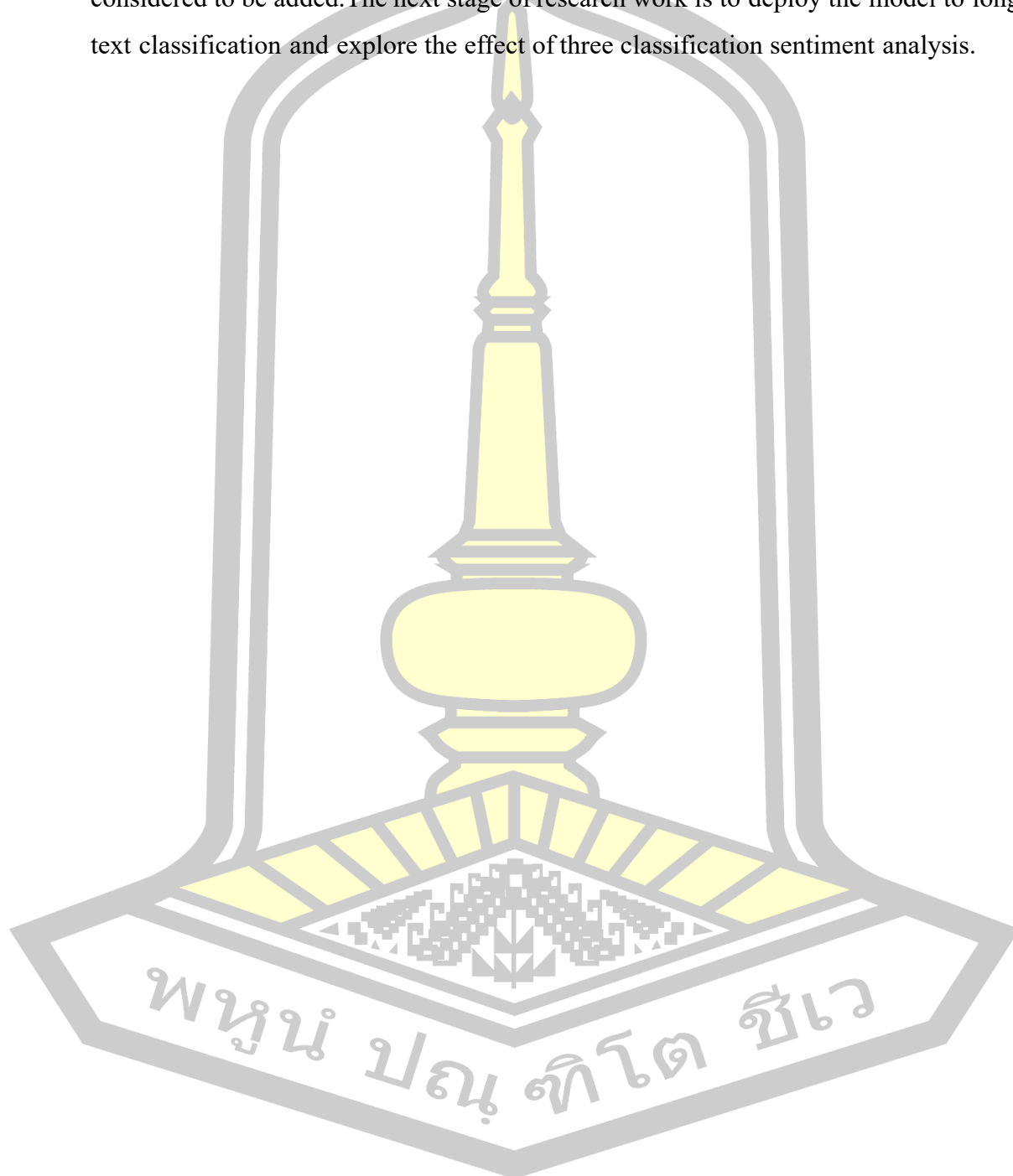
F-value compared to the method of randomly initializing word vectors, indicating that Word2Vec has a good effect on the training of word vectors. When the attention mechanism is not added, the model can achieve good results by only utilizing the feature extraction advantages of CNNs and BiLSTM. However, the attention mechanism can amplify the differences in text features in the final results, achieving further mining of text features. The embedding of CNNs and BiLSTM is beneficial for improving model performance. The introduction of BiLSTM model can increase the richness of semantic information, effectively capturing long-distance dependencies and contextual information in text; After removing the CNN network with multiple channels, local features of multi-scale text were ignored, resulting in a significant decrease in F-value. This also indicates that increasing the structure of CNNs has a positive effect on sentiment polarity analysis models.

Therefore, the analysis results of the model in this article are better than other models. On the one hand, it fully utilizes the advantages of CNN and BiLSTM models, which can extract multi-scale local feature information in the text while parsing contextual semantic information; On the other hand, by constructing an attention mechanism to obtain more implicit information, the impact of non-key vocabulary on the model is reduced. Multiple features are fused in a multi-channel form to achieve sentiment enhancement of word vectors, making semantic features more correlated with sentiment polarity labels, thus achieving the best sentiment classification performance.

5.5 Conclusions

A Microblog sentiment analysis method is proposed based on CNN, BiLSTM, and attention mechanism. This method focuses on multi-scale text input features and achieves the importance allocation of different dimension text features through attention mechanism. The experimental results demonstrate that the multi-channel CNNs BiLSTM sentiment classification algorithm based on attention mechanism performs better than other benchmark models on Microblog datasets and can effectively identify and analyze the emotions of netizens' Microblog posts in online public opinion.

There are still some shortcomings in the data and methods of this study. The research data is limited to Microblog short text, and the classification effect of the model on long text data is not discussed. At the same time, neutral text is not considered to be added. The next stage of research work is to deploy the model to long text classification and explore the effect of three classification sentiment analysis.



REFERENCES

- [1] China Internet Network Information Center. The 51st Statistical Report on Internet Development in China [R]. Beijing: China Internet Network Information Center, 2023.
- [2] Liu Y, Wang X J, Chen Z S, et al. Evaluating Digital Health Services Quality via Social Media[J]. IEEE Transactions on Engineering Management, 2023.
- [3] JGhahremani Y, Amiri B. A novel simplicial complex representation of social media texts: The case of Twitter[J]. Chaos, Solitons & Fractals, 2023, 173: 113642.
- [4] Liu Y, Wang X J, Chen Z S, et al. Evaluating Digital Health Services Quality via Social Media[J]. IEEE Transactions on Engineering Management, 2023.
- [5] Balick A .Social Media, Identity, and Careful Culture: How Online Social Networks Limit Identity, Amplify Difference, and Diminish Social Cohesion[J].Psychoanalytic Inquiry, 2023, 43(1):24-35.
- [6] Tang Lingyan, XIONG Congcong, Wang Yuan, et al. A review of short text Sentiment Tendency Analysis based on Deep learning [J]. Journal of Computer Science and Exploration, 2021, 15(5):18.
- [7] Zeng L , Li R Y M .Construction safety and health hazard awareness in Web of Science and Weibo between 1991 and 2021[J].Safety science, 2022(152-):152.
- [8] Kustron, Konnie,G.Searching the World Wide Web.[J]. Records Management Quarterly, 1997: 8-14.
- [9] Selberg E, Etzioni O. Multi-Service Search and Comparison Using the MetaCrawler, 2000: 195-208.
- [10] Liu L, C, Tang W. Continual queries for Internet scale event-driven information delivery[J]. Knowledge and Data Engineering, IEEE Transactions on 1999.
- [11] Heydon A, Najork M. Mercator: A scalable, extensible Web crawler [J]. World Wide Web-internet& Web Information Systems, 1999, 2(4): 219-229.
- [12] Najork M, Heydon A. High-Performance Web Crawling [J]. Massive Computing, 2001.
- [13] Cho J, Garcia-Molina H. Parallel crawlers[C]. Proceedings of the 11th

International Conference on World Wide Web. New York, NY, USA: ACM, 2002: 124-135.

[14] Boldi P, Codenotti B, Santini M. UbiCrawler: a scalable fully distributed Web crawler [J]. *Software Practice & Experience*, 2004, 34(8): 711-726.

[15] Maggini M, Gori M, Sarti L. Exact and approximate graph matching using random walks[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005, 27(7):1100-1111.

[16] Harth A, Umbrich J, Decker S. MultiCrawler: A Pipelined Architecture for Crawling and Indexing Semantic Web Data [J]. *International Conference on the Semantic Web*, 2006: 259-271.

[17] Matsuo Y, Mori J, Hamasaki M. POLYPHONET: an advanced social network extraction system from the Web[C]// *Proceedings of the 15th international conference on World Wide Web*, WWW 2006: 23-26.

[18] Hurst M, Maykov A. Social streams blog crawler[C]//2009 IEEE 25th International Conference on Data Engineering. IEEE, 2009: 1615-1618.

[19] Lian Jie, Zhou Xin, Cao Wei, et al. Sina microblog data mining program[J]. *Journal of Tsinghua University (Natural Science Edition)*, 2011, (10): 1300-1305.

[20] Berger P, Hennig P, Bross J. Mapping the Blogosphere--Towards a universal and scalable Blog-Crawler[C]//2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing. IEEE, 2011: 672-677.

[21] Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters [J]. *Communications of the ACM*, 2008, 51(1): 107-113.

[22] Khare R, Cutting D, Sitaker K. Nutch: A flexible and scalable open-source web search engine [J]. *Oregon State University*, 2004: 32-42.

[23] Chen Shunhua, Wang Xiaotong, Hao Zhifeng, et al. Distributed crawling technology based on microblog API[J]. *Telecommunication Science*, 2013, 29(8): 146-150.

[24] Ferreira R, Freitas F, Brito P. RetriBlog: An architecture-centered framework for developing blog crawlers[C]// Elsevier Ltd. Elsevier Ltd, 2013:1177-1195.

[25] Pamulaparty L, Rao CVG, Rao M S. A novel approach for avoiding overload in the web crawling[C]//2014 International Conference on High Performance Computing and Applications (ICHPCA). IEEE, 2014: 1-6.

[26] Zhong Mingxiang, Tang Jintao, Xie Songxian, et al. A microblog data crawling method based on dynamic web page parsing[J]. Ship Electronic Engineering, 2015, (10): 95-99.

[27] Gossen G, Demidova E, Risse T. iCrawl: Improving the freshness of web collections by integrating social web and focused web crawling[C]//Proceedings of the 15th ACM/IEEE-CS Joint Conference on Digital Libraries. 2015: 75-84.

[28] Lu Yang, Li Huakang, Sun Guozi. A distributed microblog crawler system based on P2P technology[J]. Journal of Jiangsu University (Natural Science Edition), 2016, 037(003):296-301.

[29] Gunawan D, Amalia A, Najwan A. Improving data collection on article clustering by using distributed focused crawler [J]. Data Science: Journal of Computing and Applied Informatics, 2017, 1(1): 1-12.

[30] Liu Fangyun, Zhang Zhiyong, LI Yuxiang. Research on distributed parallel incremental crawling technology based on Hadoop[J]. Computer Measurement and Control, 2018, 26(10):8. DOI:CNKI:SUN:JZCK.0.2018-10-058.

[31] Sekhar SRM, Siddesh G M, Manvi S S, et al. Optimized focused web crawler with natural language processing based relevance measure in bioinformatics web sources[J]. Cybernetics and Information Technologies, 2019, 19(2): 146-158.

[32] GSD, Ghantasala P, Chinnam V S, et al. Range of web Crawling from HTTP Parse and HTML Requests as Static Digraph and Web Pages [J]. VenkataSai and Karukuri, Raju and Sripad, Sharath, Range of web Crawling from HTTP Parse and HTML Requests as Static Digraph and Web Pages (August 8, 2020), 2020.

[33] Xie Rongrong, XuHui,Zheng Shuaiwei,et al.Simulation of webpage big data crawling method based on web crawler [J]. Computer Simulation, 2021, 38(6):439-443.

[34] Long Xiangyu. Design of Data Crawling Program Based on Web Crawling Technology[J]. Technology and Market, 2021, 28(10): 41-43.

[35] Shengmin Zhang, Aiju Wang . Research and Design of Distributed Multi-Topic Web Crawler Based on Python [J]. Journal of Kaifeng University, 2021,

35(1):93-96.

[36] Lei Gang, Lai Yu, Yu Mengyu. Design of Weibo Web Crawler and Public Opinion Visualization Analysis Experimental Platform Based on Public Cloud[J]. Laboratory Research and Exploration, 2021, 40(10): 273-276+294.

[37] He Bo. A Python-based approach for maintaining crawler programs in Sina Weibo[J]. Software, 2022, 43(02): 52-54.

[38] Sun Shouyu. Design and implementation of Sina Weibo crawler program based on Python[J]. Science and Technology Information, 2022, 20(12): 34-37.

[39] Lu Lili. Design and realization of user tracking system for social platform opinion based on crawler[J]. Computer Knowledge and Technology, 2022, 18(01): 26-28.

[40] Deng Xiaolu, Yao Song. Scrapy-based data crawler for Sina Weibo[J]. Modern Information Technology, 2023, 7(03): 44-47.

[41] ZHU Rongshen, SUN Chuan-nang, PAN Hong. A python based distributed network anti-crawler data ordering study[J]. Computer Simulation, 2023, 40(05): 426-429+447.

[42] SUN Chuan-nang, ZHU Rong-shen, LI Xiu. Research on web crawler KMR based on Python language[J]. Computer Simulation, 2023, 40(03): 504-507.

[43] Liu Dolin, Lv Miao. Simulation of distributed web crawler data collection algorithm under Scrapy framework[J]. Computer Simulation, 2023, 40(06): 504-508.

[44] Mohammad S M. Sentiment analysis: Automatically detecting valence, emotions, and other affectual states from text[M]//Emotion Measurement. Woodhead Publishing, 2021: 323-379.

[45] Nasukawa T, Yi J. Sentiment analysis: Capturing favorability using natural language processing[C]//Proceedings of the 2nd international conference on Knowledge capture. ACM, 2003: 70-77.

[46] Chen L C, Lee C M, Chen M Y. Exploration of Social Media for Sentiment Analysis Using Deep Learning[J]. Soft Computing, 2020, 24(11): 8187-8197.

[47] Pan D, Yuan J, Li L, et al. Deep neural network-based classification model for Sentiment Analysis[C]//2019 6th International Conference on Behavioral, Economic and Socio-Cultural Computing (BESC). IEEE, 2019: 1-4.

[48] H. Wang and D. Zhao, Emotion analysis of microblog based on emotion dictionary and Bi-GRU, 2020 Asia-Pacific Conference on Image Processing, Electronics and Computers (IPEC), Dalian, China, 2020, pp. 197-200.

[49] K. Jia and Z. Li, Chinese Micro-Blog Sentiment Classification Based on Emotion Dictionary and Semantic Rules, 2020 International Conference on Computer Information and Big Data Applications (CIBDA), Guiyang, China, 2020, pp. 309-312.

[50] W. Tao, "Emotional classification model of review text based on topic and sentiment characteristics," 2020 2nd International Conference on Economic Management and Model Engineering (ICEMME), Chongqing, China, 2020, pp. 1057-1061,

[51] Y. Zhang, J. Sun, L. Meng and Y. Liu, Sentiment Analysis of E-commerce Text Reviews Based on Sentiment Dictionary, 2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), Dalian, China, 2020, pp. 1346-1350.

[52] NIU Y, XIE R, LIU Z. Improved word representation learning with sememes[J]. ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers), 2017, 1: 2049–2058.

[53] Hua Bei, Peng Xuechun. Research on microblog sentiment classification method based on dictionary and rule set[J]. Journal of Taiyuan Normal University: Natural Science Edition, 2021, 20(04): 48-54.

[54] Ahmed M, Chen Q, Li Z. Constructing Domain-Dependent Sentiment Dictionary for Sentiment Analysis[J]. Neural Computing and Applications, 2020, 32(18): 14719-14732.

[55] Yanyan Zhao, Bing Qin, Qiuhui Shi, Ting Liu. Construction of large-scale emotion lexicon and its application in emotion classification [J]. Journal of Chinese Information, 2017, 31(02): 187-193.

[56] Zhang S, Wei Z, Wang Y, et al. Sentiment analysis of Chinese micro-blog text based on extended sentiment dictionary[J]. Future Generation Computer Systems, 2018, 81: 395-403.

[57] Ke-Yu Chen. Research on Text Sentiment Classification in Hotel Domain [D]. Chongqing University, 2017.

[58] Wang Xuehe, Zhao Hua. An automatic method for constructing sentiment

lexicon oriented to the propensity analysis of emergencies[J]. Journal of Ningxia University (Natural Science Edition), 2021, 42(3): 290-294.

[59] Lin L, Li J, Zhang R, et al. Opinion mining and sentiment analysis in social networks: A retweeting structure-aware approach[C]. Proc of 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing, 2014.

[60] Turney Peter D. Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews[J]. Proceedings of the Association for Computational Linguistics, 2022, 07: 471-424.

[61] KANG Yue, XUE Huizhen, HUA Bin. Fine-grained commodity evaluation analysis for deep learning networks[J]. Computer Engineering and Applications, 2021, 57(11): 8.

[62] Huanhuan Yuan, Yongli Wang, Xia Feng, et al. Sentiment Analysis Based on Weighted Word2vec and Att-LSTM[P]. Computer Science and Artificial Intelligence, 2018-12-08.

[63] Vaibhavi N, Patodkar N P, Shaikh I R. Sentimental analysis on twitter data using naive bayes[C]. Proceedings of the 6th Post graduate conference for computer engineering, 2017.

[64] Y.S. Li, L.M. Wang, Y.M. Chai, et al. Research on dynamic emotion dictionary construction method based on bidirectional LSTM[J]. Small Microcomputer Systems, 2019, 40(03): 503-509.

[65] Tang Xiao-bo, Yan Cheng-xi. A sentiment classification model based on SPIPRO principle and support vector machine[J]. Information Studies: Theory & Application, 2013, 36(1): 98-103. (in Chinese)

[66] Haque T U, Saber N N, Shah F M. Sentiment analysis on large scale Amazon product reviews[C]//2018 IEEE international conference on innovative research and development (ICIRD). IEEE, 2018: 1-6.

[67] İlhan Tarimer, Adil Çoban, Arif Emre Kocaman. Sentiment Analysis on IMDB Movie Comments and Twitter Data by Machine Learning and Vector Space Techniques.[J]. CoRR, 2019, abs/1903.11983.

[68] Cao HT. Research on sentiment analysis of Chinese microblogs based on PAD model: Dalian University of Technology; 2013.

[69] Liu Z, Liu R. Empirical research on sentiment classification of Chinese

microblogs based on machine learning [J]. Empirical study on Chinese microblog sentiment classification based on machine learning [J]. Computer Engineering and Application. 2012;1.

[70] HUANG Lei, LI Shoushan, ZHOU Guodong. Research on microblog sentiment recognition method based on syntactic information [J]. Computer Science, 2017, 44(2): 244-249.

[71] Zhang Songyan. Semi-supervised deep learning and its application in sentiment analysis [D]. Shanxi University, 2021. DOI:10.27284/d.cnki.gsxiu.2021.001692.

[72] WANG Ting, YANG Wenzhong. A review of research on text sentiment analysis methods [J]. Computer Engineering and Applications, 2021, 57(12): 11-14.

[73] S. S and P. K. V, Fine Grained Sentiment Analysis of Malayalam Tweets Using Lexicon Based and Machine Learning Based Approaches, 2021 4th Biennial International Conference on Nascent Technologies in Engineering (ICNTE), NaviMumbai, India, 2021, pp. 1-6.

[74] B. Lin, X. Zhu and J. Hu, Machine Learning-based Weibo user group profiling under hot events, 2022 International Conference on Culture-Oriented Science and Technology (CoST), Lanzhou, China, 2022, pp. 283-287.

[75] Hinton G E. Learning distributed representations of concepts [C]//Proceedings of the eighth annual conference of the cognitive science society. 1986, 1: 12.

[76] Mikolov T, Yih W, Zweig G. Linguistic regularities in continuous space word representations [C]//Proceedings of the 2013 conference of the north american chapter of the association for computational linguistics: Human language technologies. 2013: 746-751.

[77] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space [J]. arXiv preprint arXiv:1301.3781, 2013.

[78] Pennington J, Socher R, Manning C D. Glove: Global vectors for word representation [C]//Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014: 1532-1543.

[79] J. Li, H. Li. Research on feature extraction and sentiment classification of short text review products based on deep learning [J]. Intelligence Theory and

Practice,2018,41(02):143-148.

[80] Chen Ke,Liang Bin,Ke Wende,Xu Bo,Zeng Guochao. Sentiment analysis of Chinese microblogs based on multi-channel convolutional neural network[J]. Computer Research and Development,2018,55(05):945-957.

[81] Luo Fan,Wang Houfeng. Combining RNN and CNN hierarchical networks for sentiment classification of Chinese text[J]. Journal of Beijing University (Natural Science Edition),2018,54(03):459-465.DOI:10.13209/j.0479-8023.2017.168.

[82] Liu Gaojun,Zhao Ximing. Hierarchical microblog sentiment analysis based on BGRU-CNN[J]. Journal of Northern Polytechnic University Journal,2019,31(02):68-76.

[83] Yukun Ma, Haiyun Peng, Tahir Khan, et al. Sentic LSTM: A hybrid network for targeted aspect-based sentiment analysis[J]. Cognitive Computation, 2018, 10(4): 639-650.

[84] TENG Fei, ZHENG Chaomei, LI Wen. Multidimensional topic model for oriented sentiment analysis based on long short-term memory[J]. Journal of Computer Applications, 2016, 36(8): 2252-2256.

[85] Jiansheng Han, Jie Chen, Peng Chen, Jie Liu, Dezhong Peng. Sentiment classification of Chinese text based on bidirectional temporal deep convolutional networks[J]. Computer Applications and Software,2019,36(12):225-231.

[86] Zhu X, Sobihani P, Guo H. Long short-term memory over recursive structures[C]//International Conference on Machine Learning. 2015: 1604-1612.

[87] Hu Fei, Li Li, Zhang Zi-li, et al. Emphasizing essential words for sentiment classification based on recurrent neural networks[J]. Journal of Computer Science and Technology, 2017, 32(4): 785-795.

[88] Wang J, Yu L, Lai K R, et al. Investigating dynamic routing in tree-structured lstm for sentiment analysis [C]. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 2019: 3430-3435P

[89] Li Weijiang, Chia Fang. Sentiment analysis based on multichannel bidirectional long- and short-term memory networks[J]. Journal of Chinese Information, 2019, 33(12): 119-128

[90] Wang J, Yu L C, Lai K R, et al. Dimensional sentiment analysis using a regional cnn-lstm model [C]. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, Association for Computational Linguistics, 2016: 225-230P

[91] ang D, Qin B, Liu T. Document modeling with gated recurrent neural network for sentiment classification[C]//Proceedings of the 2015 conference on empirical methods in natural language processing. 2015: 1422-1432.

[92] Alayba A M, Palade V, England M, et al. A combined CNN and LSTM model for arabic sentiment analysis[C]//International Cross-Domain Conference for Machine Learning and Knowledge Extraction. Springer, Cham, 2018: 179-191.

[93] ZHANG Zhi-Hua. Research on sentiment word vector and text sentiment analysis based on deep learning [D]. East China Normal University,2016.

[94] ZHOU Kai. Research on text sentiment analysis algorithm based on deep learning[D]. Guizhou University,2019.

[95] Chen X,Yang S,Yao YH. Text sentiment classification method based on BLSTM algorithm and word fusion[J].Journal of China University of Metrology,2020,31(02):225-232.

[96] Y. Zhang and J. Liu, "Research on Chinese Sentiment Classification Based on BERT-www-BiLSTM-SVM," 2023 IEEE International Conference on Control, Electronics and Computer Technology (ICCECT), Jilin, China, 2023, pp. 832-837.

[97]ShaoPeng C,Xiaoke W,Shunan Z, et al. Effect of daily new cases of COVID-19 on public sentiment and concern: Deep learning-based sentiment classification and semantic network analysis.[J]. Zeitschrift fur Gesundheitswissenschaften = Journal of public health,2023.

[98] J. Yu, J. Jiang, and R. Xia, "Entity-sensitive attention and fusion network for entity-level multimodal sentiment classification," IEEE/ACM Transactions on Audio, Speech, and Language Processing, Vol. 28, No. 9, pp. 429–439, Jan. 2020.

[99] K. Anuratha and M. Parvathy, "Twitter sentiment analysis using social-spider lex feature-based syntactic-senti rule recurrent neural network classification,"International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,Vol.30, No. Supp01, pp. 45–66, Jun.2022.

[100] R. Duwairi, M. N. Al-Refai, and N. Khasawneh, "Feature reduction

techniques for Arabic text categorization,” *Journal of the American Society for Information Science and Technology*, Vol. 60, No. 11, pp. 2347–2352, Jul. 2009.

[101] M. C. Q. Farias, P. H. de Castro Oliveira, G. Dos Santos Lopes, C. J. Miosso, and J. A. Lima, “The Influence of magnetic resonance imaging artifacts on CNN-based brain cancer detection algorithms,” *Computational Mathematics and Modeling*, Vol. 33, No. 2, pp. 211–229, Jan. 2023.

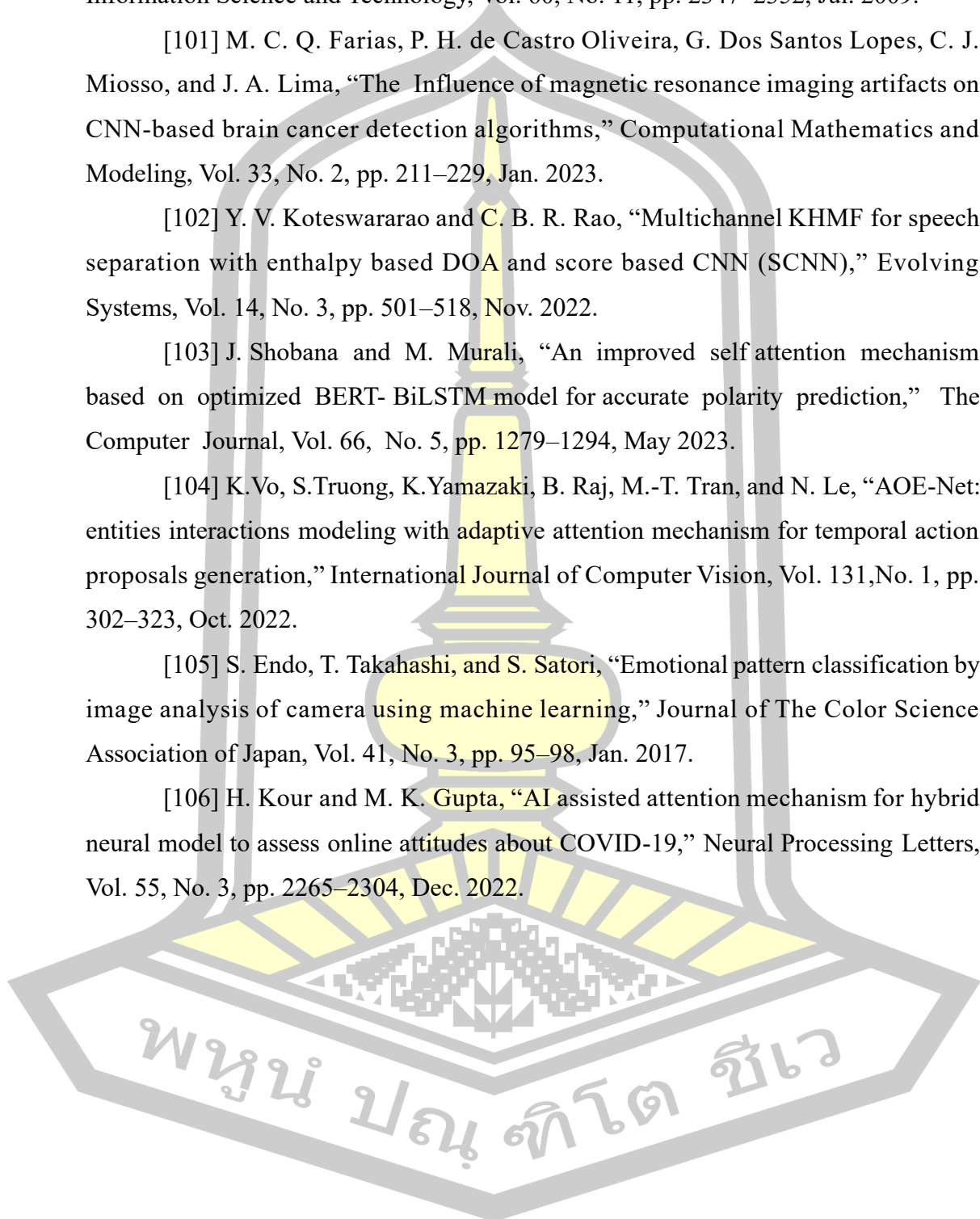
[102] Y. V. Koteswararao and C. B. R. Rao, “Multichannel KHMF for speech separation with enthalpy based DOA and score based CNN (SCNN),” *Evolving Systems*, Vol. 14, No. 3, pp. 501–518, Nov. 2022.

[103] J. Shobana and M. Murali, “An improved self attention mechanism based on optimized BERT- BiLSTM model for accurate polarity prediction,” *The Computer Journal*, Vol. 66, No. 5, pp. 1279–1294, May 2023.

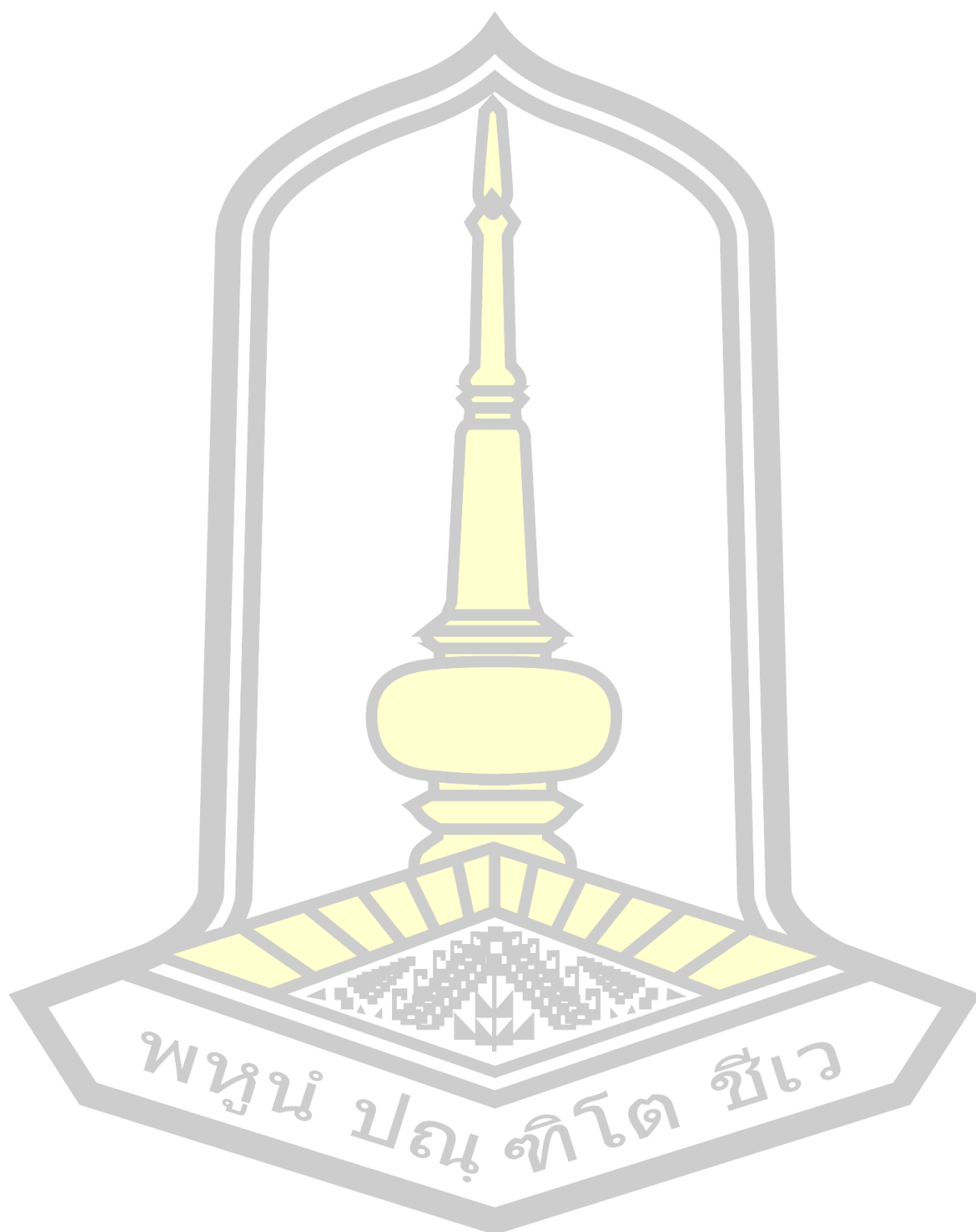
[104] K. Vo, S. Truong, K. Yamazaki, B. Raj, M.-T. Tran, and N. Le, “AOE-Net: entities interactions modeling with adaptive attention mechanism for temporal action proposals generation,” *International Journal of Computer Vision*, Vol. 131, No. 1, pp. 302–323, Oct. 2022.

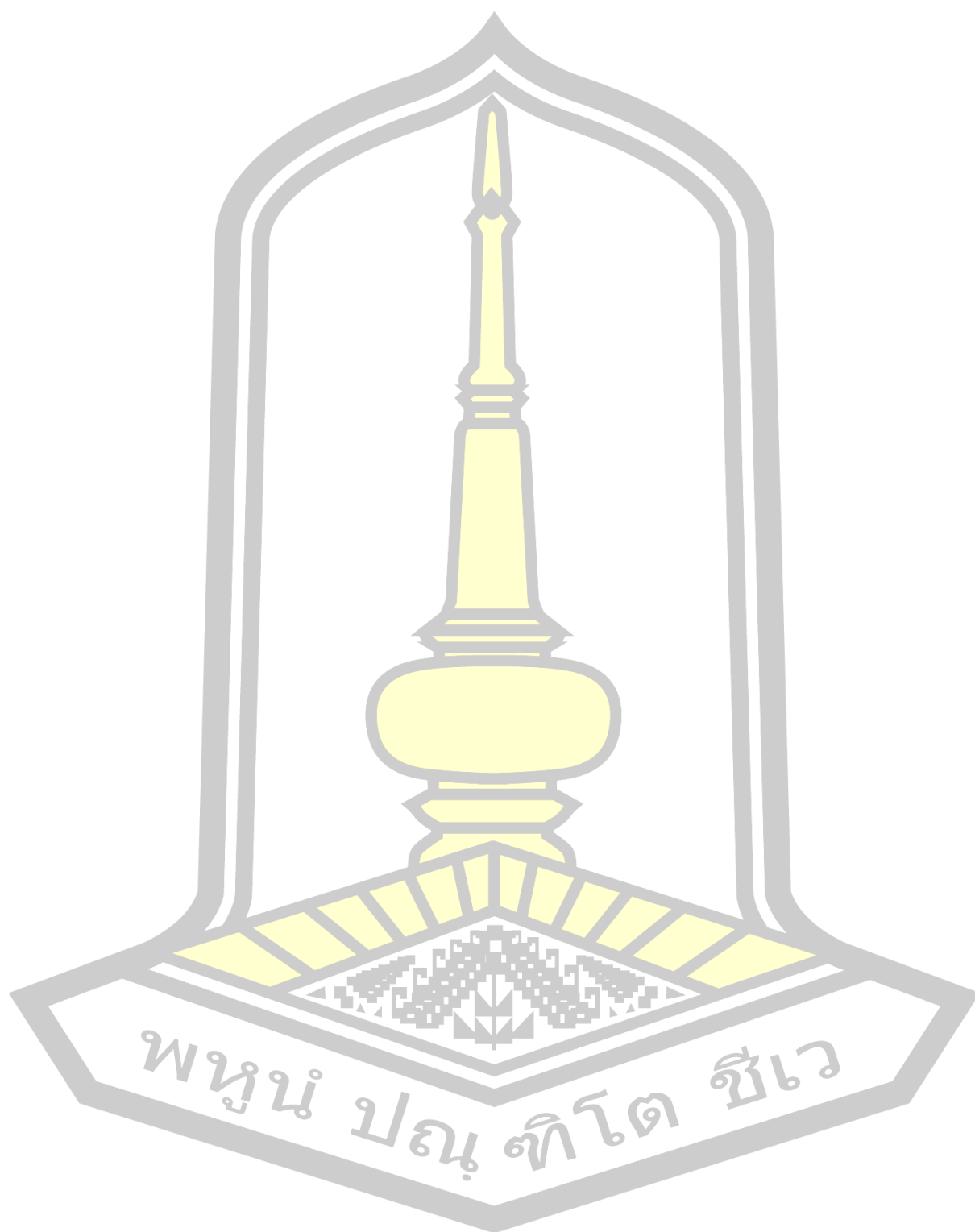
[105] S. Endo, T. Takahashi, and S. Satori, “Emotional pattern classification by image analysis of camera using machine learning,” *Journal of The Color Science Association of Japan*, Vol. 41, No. 3, pp. 95–98, Jan. 2017.

[106] H. Kour and M. K. Gupta, “AI assisted attention mechanism for hybrid neural model to assess online attitudes about COVID-19,” *Neural Processing Letters*, Vol. 55, No. 3, pp. 2265–2304, Dec. 2022.



REFERENCES





BIOGRAPHY

NAME	Xinyue Feng
DATE OF BIRTH	15 March 1990
PLACE OF BIRTH	Shanxi, China
ADDRESS	Yundonghai Street, Sanshui District, Foshan City, Guangdong Province
POSITION	Foshan, Guangdong
PLACE OF WORK	Foshan, Guangdong
EDUCATION	2008Xinjiang Haiquan Middle School 2012 Becheior Degree(B.Eng), north university of china 2015 Master Degree(M.Eng), north university of china 2025 Doctoral Degree(Ph.D),Electrical and Computer, Mahasarakham University

