令和 4 年度

博士論文

題目

Research on Emotional Text Generation and Multi-label Textual Emotion Detection

(感情テキスト生成とマルチラベルテキスト感情検出に関する研究)

報告者

周　洋阳　(シュウ　ヨウヨウ　/　Zhou　Yangyang)

徳島大学 大学院 先端技術科学教育部

システム創生工学専攻 知能情報システム工学コース

Contents

Abstract 1

Chapter 1 Introduction 4

1.1 Motivation and Significance 4

1.2 Outline and Contributions 6

1.3 Thesis Organizations 11

Chapter 2 Background 12

2.1 Text Generation 12

2.2 Textual emotion detection 14

2.2.1 Mainstream Methods 14

2.2.2 Other Strategies 16

Chapter 3 Emotional Text Generation 19

3.1 Task Definition 19

3.2 Methodology 19

3.2.1 Model for Text Generation 19

3.2.2 Retrieval Strategy in Reranking 21

3.3 Experiment Setups 23

3.3.1 Datasets 23

3.3.2 Evaluation Metrics 24

3.3.3 Baselines 25

3.3.4 Experimental Details 25

3.4 Experimental Results 26

3.5 Discussion 29

3.6 Summary 30

Chapter 4 Multi-label Textual Emotion Detection 32

4.1 Task Definition 32

4.2 Methodology . . . 32
4.2.1 Encoder and Classifier for Detection . . . 32
4.2.2 Contrastive Strategy in Encoding . . . 33
4.2.3 Prompt Engineering in Encoding . . . 35
4.2.4 Consistency Strategy in Training . . . 37
4.2.5 Emotional Dictionary Matching in Preprocessing . . . 39
4.2.6 Commonsense Knowledge Inference in Preprocessing . . . 42
4.2.7 Topic Model Clustering in Preprocessing . . . 43
4.2.8 Multi-label Loss Function for Detection . . . 45
4.3 Experiment Setups . . . 47
4.3.1 Datasets . . . 47
4.3.2 Evaluation Metrics . . . 49
4.3.3 Baselines . . . 50
4.3.4 Experimental Details . . . 51
4.4 Experimental Results . . . 53
4.5 Discussion . . . 60
4.5.1 About Contrastive Strategy . . . 60
4.5.2 About Fine-tuning . . . 61
4.5.3 About Prompt Engineering . . . 63
4.5.4 About Multi-label Loss Function . . . 65
4.5.5 About Neurosymbolic Approaches . . . 67
4.6 Summary . . . 71

Chapter 5 Conclusion and Future work 72
5.1 Conclusion . . . 72
5.2 Future work . . . 75

Acknowledgment 76

Bibliography 77

Table List

3.1 An example of manual evaluation. 24
3.2 The evaluation result and the safe-response statistic of like emotion. 26
3.3 The evaluation result and the safe-response statistic of sadness emotion. 26
3.4 The evaluation result and the safe-response statistic of disgust emotion. 27
3.5 The evaluation result and the safe-response statistic of anger emotion. 27
3.6 The evaluation result and the safe-response statistic of happiness emotion. . . 28
4.1 The experimental results of contrastive strategies on the Ren-CECps dataset. In this paper, for the metrics of Micro F1, Macro F1 and AP, higher values indicate better performance; For the metrics of CE and RL, lower values indicate better performance. 53
4.2 The experimental results of contrastive strategies on the NLPCC2018 dataset. . 54
4.3 The ablation experimental results of prompt consistency on the Ren-CECps dataset. 54
4.4 The ablation experimental results of prompt consistency on the NLPCC2018 dataset. 56
4.5 The experimental results of neuro-symbolic approaches on the Ren-CECps dataset. 57
4.6 The experimental results of neuro-symbolic approaches on the NLPCC2018 dataset. 58
4.7 The confusion matrices for six emotions on the NLPCC2018 test set. True represents the ground truth, and Pred represents the predicted result. 63
4.8 The results of experiments with the weight of the consistency training on the two datasets. 67

Figure List

1.1 Overview of CERG architecture. The input below, from left to right, is emotion, post and response. The model includes 3 embedding layers and 12 transformer blocks. The current position predicts the next character. 7

1.2 An example of multi-label textual emotion detection. We incorporate the prompting method in the detection process to make the model more accurately output the emotions implied in the input text. 9

2.1 The baseline model from our previous work. Posts, responses and emotions are concatenated by different encoding layers. The decoder is used to predict the next characters. 12

3.1 The left diagram is the structure of the transformer block, the right matrix is an example of the self-attention mask. 20

3.2 An example of using the retrieval method to select a better response in inference. The model predicts 2 candidate responses by beam search method. The response with a lower beam score has a higher retrieval score. 21

3.3 The distribution of different emotions in the data set. 23

4.1 Overall framework of the model with contrastive strategy added to the encoding part. 33

4.2 Overall framework of the model with contrastive strategy added to the classification part. 34

4.3 The overview of prompt engineering. The original emotion labels are first changed into synonyms, then they are randomly combined into two sets of labels, and finally they are shuffled into a pair of prompts. 36

4.4 The overview framework of the prompt tuning with consistency training model. The input consists of a prompt and a text, and the output is a multi-label classification result. 37

4.5 The overview of the consistency training strategies. A training data point yields four predictions for a pair of prompts represented by the blue box and the yellow box, respectively. 39

4.6 Histogram of the number of words for different emotions in the emotional dictionary. 40
4.7 An example of matching keywords with an emotional dictionary and aiding emotion detection. 41
4.8 An example of converting the number of emotional words into a vector to aid emotion detection. 41
4.9 An example of using commonsense knowledge inference to aid emotion detection. 43
4.10 An example of using a topic category to aid emotion detection. 44
4.11 An example of using topic words to aid emotion detection. 45
4.12 The emotion correspondence between DLUTE and the two datasets. The Ren-CECps dataset is on the left and the NLPCC2018 dataset is on the right. . . . 47
4.13 Histogram of the number of texts for different emotions in the Ren-CECps dataset. 48
4.14 Histogram of the number of texts for different emotions in the NLPCC2018 dataset. 49
4.15 The emotional correlation coefficient matrix of the Ren-CECps test set. 50
4.16 The emotional correlation coefficient matrix learned by PTC-MTED. 62
4.17 The Macro F1 score performance of the PTC-MTED model with different sizes of NLPCC2018 training data. 64
4.18 The emotional correlation coefficient matrix of the Ren-CECps test set. The top is the matrix of BERT-FT, and the bottom is the matrix of $D_{wn}K_{drn}T_{cw}$. . . . 66
4.19 Histogram of proportionally normalized emotions in Ren-CECps texts of two topics. The vertical coordinate refers to the proportion of the emotion in the texts of the topic compared to that in the whole dataset. 69
4.20 A case study of a visualization that explains the neurosymbolic approach by SHAP. The effect of the symbolic information in emotion detection is at the top. The effect of text information in emotion detection is at the bottom. Numbers with absolute values over 0.03 are shown above the words. 70

Abstract

Recently, emotional computing has received more and more attention from scholars. Machines with emotions can give us a better human-computer interaction experience. The expression and detection of emotions are two important aspects of machines with emotions. This paper is devoted to exploring machine generation of text with emotion and detection of all emotions contained in the text.

Text as a carrier of emotion is one of the important research topics in the field of affective computing. The dialogue system is an important part of this research. So far, most of the mature dialogue systems are task-oriented based, while non-task-oriented dialogue systems still have a lot of room for improvement. We propose a data-driven non-task-oriented dialogue generator "CERG" based on neural networks. This model has the emotion recognition capability and can generate corresponding responses. We try to concatenate the post and the response with the emotion, then mask the response part of the input text character by character to emulate the encoder-decoder framework. We use the improved transformer blocks as the core to build the model and add regularization methods to alleviate the problems of overcorrection and exposure bias. In the training process, we mask the response part of the input text character by character to emulate the encoder-decoder framework to prevent the leakage of information during inference. We replace the characters with the BERT model predicted characters at random positions of the input text, which will improve the robustness of the model without disrupting the training parallelism. We introduce retrieval methods in the inference process. We calculate the weight scores of similar posts and responses together with beam search, which can make the predicted responses more in line with the context.

The data we adopt comes from the NTCIR-14 STC-3 CECG subtask. The data set contains 6 emotion categories and the corresponding 1.7 million Chinese Weibo post-response pairs. After concatenating emotion, post and response, we employ three embedding layers including token, position and segment embedding layers and 12 transformer blocks for representation. To train the model with the conventional optimizer, we adjust the position of the layer normalization in the transformer blocks. We adopt a hard voting manual metric to evaluate the generative ability of our model. The coherence, fluency, and emotional relevance scores of our model in

the manual evaluation are higher than the model without the retrieval method and the baseline model. The proportion of safe and commonplace responses has also been greatly reduced. The results of the manual evaluation show that our proposed model can make different responses to different emotions to improve the human-computer interaction experience. This model can be applied to lots of domains, such as automatic reply robots of social application.

Textual emotion detection is also playing an important role in the human-computer interaction domain. Texts often contain more than one emotion. The purpose of this paper is to explore the detection of all possible emotions in a text. The mainstream methods of textual emotion detection are extracting semantic features and fine-tuning by language models. Current methods of textual emotion recognition mainly use large-scale pre-trained models fine-tuning. However, these methods are not accurate enough in the semantic representation of sentences. Contrastive learning has been shown to optimize the representation of vectors in the feature space. Therefore, we introduce the contrastive strategies to the textual emotion recognition task. Weintroduce the contrastive strategies for multi-label textual emotion classification tasks. Based on the large-scale pre-trained model BERT, we propose two approaches: using self-supervised contrastive learning before fine-tuning the model, and using contrastive training on the same inputs during fine-tuning.

Due to the information redundancy in semantics, it is difficult for these methods to accurately detect all the emotions implied in the text. The prompting method has been shown to make the language models more purposeful in prediction by filling the cloze or prefix prompts defined. Therefore, we design a prompting approach for multi-label classification. To stabilize the output, we design two consistency training strategies.

Neural networks are replacing symbolic approaches as better methods for textual emotion detection due to their powerful feature extraction capabilities. However, neural networks are prone to overfitting during training because of the small amount of emotion detection data. Based on experience or knowledge, symbolic approaches can fit a small amount of data by low-dimensional features and also outperform neural networks in terms of interpretability. We design three models combining symbolic approaches with neural networks for detecting all potential emotions from texts in this article. Due to the importance of emotional words in detection, we retrieve these words from texts by an emotional dictionary approach; we predict the reaction and describe the state of the subject to help detect emotions by a commonsense knowledge inference

approach; we cluster texts by a topic model since texts with similar topics may have similar emotions. In this article, we employ three symbolic approaches to assist a neural network model in detecting emotions.

We experiment with the effectiveness of the strategies on two multi-label emotion classification datasets: Ren-CECps and NLPCC2018. The experimental results demonstrate that using the contrastive strategy in the classification part is more effective in improving the accuracy of emotion recognition than using the contrastive strategy in the encoding part. Our proposed prompt tuning with consistency training for multi-label textual emotion detection (PTC-MTED) model achieves Macro F1 scores of 0.5432 and 0.5269, respectively. The experimental results indicate that our proposed method has significant effectiveness in the multi-label textual emotion detection task. The experimental results show that the symbolic approaches improve the fitting process, improve the interpretability and increase the accuracy of neural networks. This indicates that neurosymbolic methods are effective in the multi-label textual emotion detection task.

Keywords: Emotional text generation, Multi-label textual emotion detection, Large-scale pre-trained language model, Retrieval strategy, Contrastive strategy, Prompt engineering, Neurosymbolic approach.

Chapter 1 Introduction

1.1 Motivation and Significance

One of the research motivations in affective computing is to give machines emotional intelligence [1]. Machines with emotional intelligence can better interact with humans [2]. Emotion generation and emotion detection are two of the important aspects in emotional intelligence [3]. Text is one of the ways of human-computer interaction.In this article, we focus on emotion generation and emotion detection in text.

Machine responses with emotions can improve the experience during human-computer interaction. The dialogue system has been receiving much attention since the Turing Test was proposed [4]. The dialogue system responds to the topics or instructions thrown by the user by simulating human beings [5].

Based on whether the dialogue system can achieve a specific goal, it can be divided into 2 types: task-oriented and non-task-oriented dialogue systems (or chatbot) [6]. Task-oriented dialogue systems are generally used in closed domains like ticket purchase, ordering, and customer service [7]. There are 2 main types of task-oriented dialogue systems: pipeline-based and end-to-end methods. A chatbot is generally used in open domains such as psychotherapy applications [8]. There are 3 main types of chatbot: rule-based, retrieval-based, and generation-based methods. So far, due to the application of slot filling and other technologies, the task-oriented dialogue system is more mature than the chatbot [9]. With the continuous advancement of big data and deep learning technologies, we can build a data-driven chatbot [10]. The Chinese Weibo involved in this article can be regarded as some non-task-oriented dialogue.

Emotion recognition is the process by which a machine identifies human emotion. The response of the machine based on the user's emotional state can enhance the user's experience. Krakovsky et al. believe that emotion detection is one of the skills that machines can imitate and even surpass humans [11]. Emotion detection can be applied to several real-world scenarios. For example, it can be used in intelligent customer service systems to improve user satisfaction [12], as well as in user satisfaction analysis systems to develop better sales strategies [13]. Machines usually recognize the emotion by acquiring emotion-induced physiological signals or behavioral

information [14]. In this article, we focus on emotion detection from the text.

Textual emotion detection can be classified into word level, sentence level, chapter level and target level according to the research objectives [15]. Building an emotion dictionary is a common method for word-level emotion analysis [16]. The information of each word in a sentence/chapter is pooled or concatenated to represent the sentence-level/chapter-level emotion as a whole [17]. The target-level target refers to the emotion of an entity or an attribute [18]. Regardless of the level, words that contain emotional tendencies are the most critical factors in the process of emotion detection [19]. The mainstream method of classification tasks like textual emotion detection at this stage is to first fine-tune or extract features from the text by using language models and then classify them [20]. With these language models, as well as the designed semantic tasks, we can obtain the feature representation of the text [21]. For example, the bidirectional encoder representations from transformers (BERT), predict whether the sentences are contextually relevant at the first token position [22]. A large-scale pre-trained language model (PTM) can learn information between contexts from large amounts of text [23]. The state-of-the-art results for the textual emotion detection task are almost obtained by these models [24].

1.2 Outline and Contributions

Dialogue generation is closely related to the technology of natural language generation. Natural language generation is a process that transforms structured data into natural language. In the domain of deep learning, the sequence-to-sequence (seq2seq) framework is often used in dialogue generation [25]. This framework consists of an encoder and a decoder, which is a kind of end-to-end learning algorithm. The encoder of seq2seq converts the input sequence into a hidden state vector. The decoder converts the vector into an output sequence, then adopts the output of the previous step as the input of the next step. With the increase of sequence length, the problem of gradient disappearance may appear in the calculation. Seq2seq avoids this problem by using long short-term memory instead of original recurrent neural networks [26]. Because the recurrent neural network cannot do the parallel calculation, the training speed is slow. The transformer model proposed by Google Brain parallelizes this calculation process by the multi-head self-attention mechanism, which greatly improves the calculation efficiency [27]. Thus it has become the most commonly used model in the seq2seq framework in recent years. There is some work dedicated to improving the accuracy of translations or the quality of generated sentences. Some researchers are committed to improving the accuracy of translations or the quality of generated sentences by disrupting parallel computing [28]. We try to figure out a method to improve the quality of generated responses without disrupting parallel computing.

Existing data-driven non-task-oriented dialogue systems tend to generate a safe and commonplace response [29], for example, "I don't know". We introduce the retrieval method into the non-task-oriented dialogue system, aiming to alleviate this problem.

The key to improving the human-computer interaction experience is to make the dialogue system empathetic. Affective computing is the study that can recognize and simulate human affects [30]. Affective computing can improve the user-friendliness of the system [31]. Lots of scholars research dialogue system and affective computing respectively. Few studies have linked these two aspects [32] [33]. Different emotions used in the same sentence usually express different meanings. This is one of the difficulties of natural language processing technology.

Chinese Weibo emotional response is a task to study how to properly combine affective computing to a chatbot. The data set we adopt is from the NTCIR-14 STC-3 CECG subtask, which

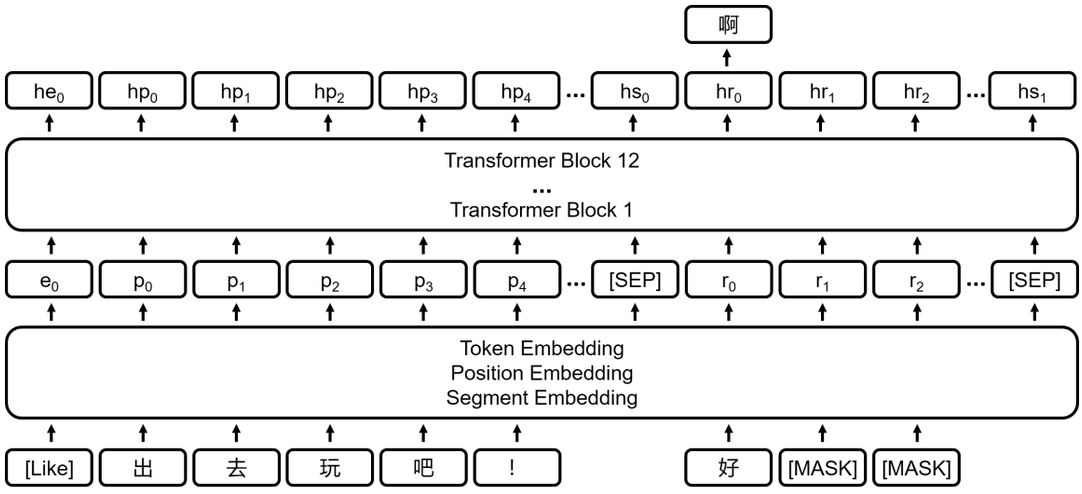

Figure 1.1 Overview of CERG architecture. The input below, from left to right, is emotion, post and response. The model includes 3 embedding layers and 12 transformer blocks. The current position predicts the next character.

is constructed from Chinese Weibo posts and replies [34]. This data set contains 6 different emotions: like, sadness, disgust, anger, happiness and other. We aim to find a way to incorporate affective computing into dialogue generation.

How to combine emotional computing with dialogue generation is a challenge. Zhou et al. proposed a memory-network-based emotional chatting machine, which introduced emotional factors into a Chinese dialogue generation system [35]. We once proposed the P&E2R model based on the LSTM network [36]. On this basis, we build a new model to improve the effect of emotional response generation. Unlike our previous work, we use the same embedding layers to deal with the emotion, the post and the response, as shown in Figure1.1. Besides, the encoder and decoder are no longer established separately. We directly employ multi-block transformers, while masking the response part of the input text character by character to avoid information leakage. Based on the teacher-forcing method [37], we add regularization methods such as character replacement to alleviate the problems of overcorrection and exposure bias while ensuring the parallel training of the model. Apart from the beam search method, we employ the retrieval method to improve the semantic relevance of generated responses in inference.

This model has made great progress in the emotional response generation. The coherence, fluency and emotional relevance scores of our model in manual evaluation, are higher than the

model without using the retrieval method and the baseline model. The proportion of safe and commonplace responses has also decreased significantly. These results indicate the effectiveness of our model. The model can be applied to the automatic reply of social applications like Chinese Weibo, and emotional chatting robots.

Some scholars regard emotion detection as a single-label classification task [38]. As a matter of fact, a sentence may contain multiple emotions. We regard emotion detection as a multi-label classification task [39].

Treating pooled or concatenated word vectors as sentence vectors will inevitably lose information. To solve this problem, some language models tend to design semantic tasks to train feature representations of whole sentences [21]. Large-scale pre-trained language models, such as bidirectional encoder representations from transformers (BERT), predict whether the sentences are contextually relevant at the first token position [22]. Semantic text similarity measures the meaning similarity of sentences. In this kind of task, the state-of-the-art results are almost obtained by large-scale pre-trained language models [40].

In large-scale pre-trained language models, the representation of word vectors is correlated with word frequency, resulting in uneven distribution of word vectors in the feature space [41]. The idea of contrastive learning is to represent the features uniformly in the feature space [42]. Therefore, Gao et al. introduce contrastive learning into the training process of large-scale pre-trained language models, which in turn achieves uniform distribution of sentence vectors in the feature space [43].

Besides emotional information, textual representations also contain other semantic information like contextual information, which may cause classification difficulties. The prompting method further applies the language model to various natural language processing tasks by means of inserting prompt as cloze or prefix into the input sample [44]. By introducing the prompting method in the textual emotion detection task, we can abstract the emotional information from the semantic space. For example, we can ask the language model to fill the blank with an emotion-bearing word. However, there is often more than one kind of emotion in the text. As shown in Figure 1.2, a good detection model should be able to detect multiple emotions. Therefore, we need to redesign the prompt and the model for the multi-label emotion detection task.

We attempt to prompt each emotion and let the model determine whether the emotion appears

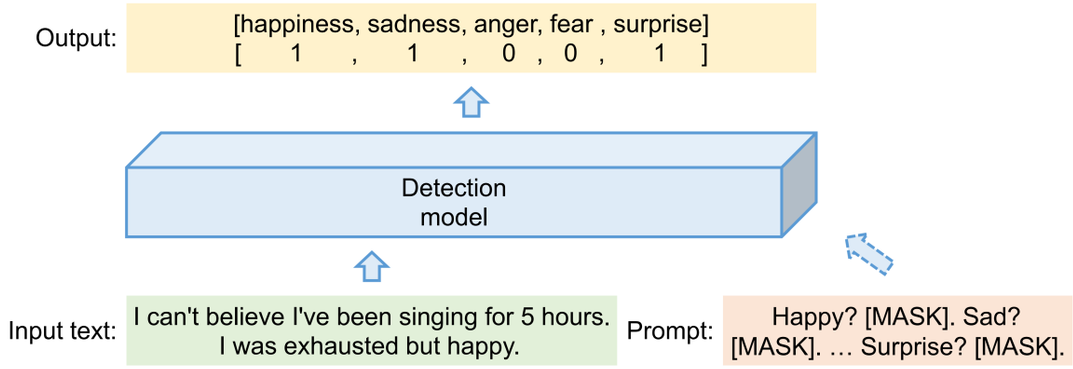

Figure 1.2 An example of multi-label textual emotion detection. We incorporate the prompting method in the detection process to make the model more accurately output the emotions implied in the input text.

in sequence. The representations are different between synonyms in the semantic space [45]. The different emotion-bearing words are biased with different datasets, which results in the semantics of the emotion-bearing words not fully covering the real situation in the space. Therefore, we try to ensemble the prompt with synonyms to improve the model' s ability to recognize ambiguous emotions.

The perturbations at the input such as adding noise often cause the unstable output of the model [46]. Consistency training has been shown to alleviate instability caused by these perturbations [47]. In addition, regularization methods such as dropout can cause inconsistencies between the training model and the prediction model. We adopt two kinds of consistency training strategies to make the output of a pair of synonym prompts tend to be consistent.

Due to the small amount of data for the textual emotion detection task, the neural network models are prone to overfitting [48]. Based on experience or knowledge, the symbolic approach can fit small datasets with refined (low-dimensional) features [49]. The symbolic approach here refers to the process by which machines deal with symbolic codes in an algebraic or logical way [50]. For example, an expert system simulates an expert in a specific field to answer questions by formulating rules or retrieving from a knowledge base [51].

In the process of emotion detection, words that contain emotional tendencies are the most critical factors [19]. Therefore, Li et al. identified emotions through the emotional dictionary [16]. Words that imply different kinds of emotions can be collected and be used to build an

emotional dictionary. When an emotional word appears in the input text, we believe that the text may imply this emotion. Besides, Minsky et al. found that emotions can be detected by knowledge [52]. For example, the machine can infer what mental state the person is in by using a knowledge graph.

Although the above symbolic methods are currently less accurate than neural network methods for emotion detection, it does not mean that they are ineffective. Neural networks often require additional methods to cope with outliers that are very different from the training data [53]. And the symbolic approach is superior to the neural network in terms of interpretability [54]. The combination of symbolic approaches and neural networks may perform better [55]. For example, the topic model can use statistical methods to cluster texts into several abstract topics [56]. These topics cannot be used directly for emotion classification, but we speculate that texts with similar topics may imply similar emotions.

Our contributions can be summarized as follows:

- We propose a Chinese emotional response generator CERG, and the results on the Chinese Weibo dataset show that our model is effective. Without disrupting the parallel computing, we improve the robustness of the model by using the masking and regularization methods. We introduce the retrieval method BM25 into the inference process, which greatly reduces the probability of generating safe and commonplace responses, and improves the diversity and contextual relevance of responses. We directly concatenate posts, masked responses with emotions, and adopt the embedding layers with shared weight to generate emotion-related answers, which is different from other models.
- We introduce contrastive strategies in the training phase of multi-label emotion recognition from text to further improve the recognition accuracy based on large-scale and training language models. We propose a prompt tuning with consistency training for multi-label textual emotion detection model. We combine multiple symbolic approaches with neural networks to improve the fitting process and the classification results of the model on multi-label textual emotion detection tasks.

1.3 Thesis Organizations

In this paper, we utilize multiple methods and strategies to improve the model's ability to generate text with sentiment, and to detect sentiment from text. The effectiveness of these methods and strategies is illustrated by our multiple evaluation metrics on multiple datasets. The remainder of this paper is organized as follows.

Chapter 2: Background

We review the knowledge of text generation and classification, including:

- commonly used text generation methods and strategies
- commonly used text classification methods
- a variety of strategies that may help classification model training and prediction

Chapter 3: Emotional Text Generation

We give a definition of the sentiment text generation task and propose a method for generation and a strategy for retrieval. We validate the effectiveness of this method and strategy on the dataset.

Chapter 4: Multi-label Textual Emotion Detection

We define a multi-label textual emotion detection task and then propose different methods in terms of contrastive learning, prompt engineering, and neurosymbolic approaches. Five evaluation metrics on two datasets demonstrate the effectiveness of these methods.

Chapter 5: Conclusion and Future work

We summarize the main contents of this thesis and give meaningful directions for future work.

Chapter 2 Background

2.1 Text Generation

In the NTCIR-14 STC-3 CECG subtask, we proposed the P&E2R model and got ranking second, as shown in Figure 2.1. After embedding the posts and responses with a shared weighted layer, we encode them by the recurrent neural networks. The embedding emotions are concatenated with the former features. The probability distribution of the current word is generated by a recurrent neural network decoder. This model is simple but effective. We introduce the idea of concatenation in this article. The disadvantage of this model is that the calculation of the recurrent neural network depends on the hidden state of the previous time, and it cannot be parallelized, which is very time-consuming.

Li et al. proposed the UniLM model [57]. The authors employ the transformer as the core of this model and make it parallel to improve calculation efficiency. Also, they adopt a special mask method to skillfully combine the encoder and decoder. Although we do not adopt the pre-trained model from UniLM in our article, we introduce the idea of the attention mask method to improve the speed of the generator.

There are still some problems with this method. The teacher-forcing method is the key technology to ensure that the transformer model can completely calculate all tokens in parallel during the training process. Zhang et al. pointed out that the ground truth word is used during model training, but once the predicted word is wrong in a certain position in the inference

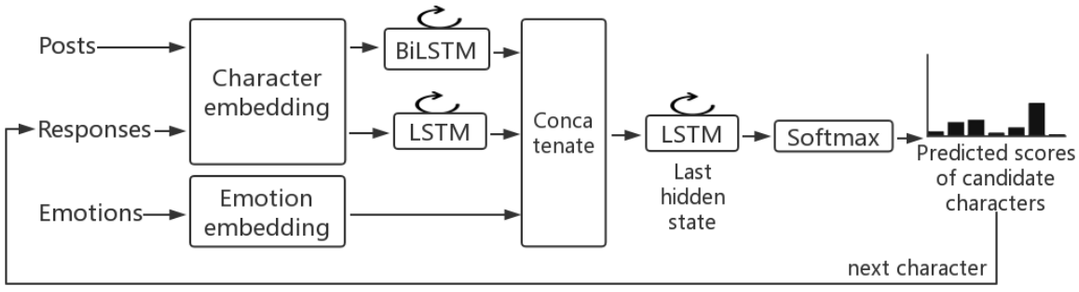

Figure 2.1 The baseline model from our previous work. Posts, responses and emotions are concatenated by different encoding layers. The decoder is used to predict the next characters.

process, the output of the model will deviate from the predetermined direction [58]. This will cause the exposure bias problem. The author proposed the word-level oracle and the sentence-level oracle method to solve the overcorrection problem brought by the teacher-forcing method. This method will disrupt the parallel computer system of the transformer model. We try to avoid disrupting the parallel computing mechanism and uses a variety of regularization methods like predicted character replacement to make the model more robust.

In addition, we also employ a beam search method in the inference process [59]. Beam search is a search algorithm that explores a graph by expanding the most promising node in a limited set. On the basis of that, we use the BM25 method and selects the most semantically relevant response among the k alternatives [60]. BM25 is a ranking function to estimate the relevance of documents to a given search query. We adopt this method to find the responses of the n closest posts and calculate their similarity to the predicted responses. The experiments show that using this retrieval method can make the responses more in line with the context.

2.2 Textual emotion detection

Textual emotion detection has recently become a hot topic due to its commercial and academic potential [48]. Textual sentiment analysis is generally performed with positive and negative in the tasks like product comments analysis [61]. Besides, emotions can be classified into various categories at fine-grained levels. Some scholars regard emotion analysis as a single-label classification task [38]. As a matter of fact, a piece of text may contain multiple emotions. Therefore, we tend to regard textual emotion analysis as a multi-label classification task [39]. In this article, the goal of the task is to detect all the possible emotions in the textual expression.

2.2.1 Mainstream Methods

Methods for textual emotion detection can be classified into four categories: **rule-based** methods, **classical learning-based** methods, **deep learning** methods, and **hybrid** methods [24].

The **rule-based** methods are well interpretable. These methods usually rely on preprocessing steps including tokenization, lemmatization, POS tagging, and stop words removal [62]. The emotion rules can be extracted by statistics or linguistic concepts. Lee et al. constructed an annotated emotion cause corpus and calculated the distribution of cause event types for each emotion class to recognize emotion cause events [63]. The results show that the rule-based methods are very sensitive to text quality. Therefore, we try to use rule-based methods to assist the neural network model to detect emotions instead of recognizing them directly in this paper.

Textual sentiment analysis is generally performed with positive and negative in the tasks like product comments analysis [61]. The use of a sentiment dictionary can be considered as a rule-based method when dealing with the textual sentiment analysis tasks. How well the sentiment analysis results depend basically on how accurately the sentiment dictionary is constructed [64].

Unlike sentiment, emotions can be classified into various categories at fine-grained levels. Most current emotional dictionary are constructed by adjectives [65]. When the text does not contain adjectives or other keywords, dictionary-based methods may perform poorly in the emotion recognition tasks.

WordNet is a database describing the connections between English words [66]. Badaro et al.

expanded the emotional dictionary with WordNet and treat the dictionary as prior knowledge for emotion recognition [67]. The expanded dictionary performed better when used for emotion recognition tasks.

Most of the datasets involve a specific domain rather than an open domain, so there is also some work to make up for the limitations of the datasets by introducing background knowledge or prior information [68]. In addition to being used to expand dictionaries, databases like WordNet are often used for knowledge representation [69]. Seol et al. used information from a third-party knowledge database directly to recognize emotions, rather than first constructing an emotional dictionary [70]. Implicit knowledge constructed with the third-party information can help when the model is unable to infer the emotion by the keywords.

Knowledge bases can be widely used in industries such as medicine, finance, and education [71]. We find that the information in these knowledge bases is objective, and direct application to a subjective task like emotion recognition may be a bit difficult. Ghosal et al. used commonsense knowledge to assist utterance-level emotion recognition, and achieved good results [72]. The authors proved that the causal relations of events or the states of the characters inferred from common sense contributed to emotion detection.

Classical learning-based methods refer to using traditional machine learning methods such as support vector machines or logistic regression. These methods rely on extracting and selecting features with the most information gain, and then outputting the optimal hyperplane [73]. Artificial feature engineering is more interpretable compared to deep learning.

Experiments by Anusha et al. showed that the important part of the sentence is essential for improving the results of emotion classification [74]. So we believe that similar texts may have similar emotional expressions. Topic models can retrieve information or extract features from unstructured documents [75]. We try to cluster the similar texts by a topic model.

Lin et al. used an LDA-based topic model to analyze sentiment [76]. The authors found that the order of the keywords of the topics might affect the results of the classification. In view of this, we try to use the keywords of the topics as symbols to assist in detecting emotions while retaining the original text.

Deep learning methods have the advantage of automatic feature engineering and a large amount of data information compared to classical learning-based methods [77]. Commonly used deep learning frameworks include convolutional neural networks, recurrent neural networks,

transformers, and so on. Majumder et al. used multiple gate recurrent units (a kind of recurrent neural networks) to model speaker state, global state and emotion representation to detect emotions in conversations [78].

The previously mentioned Ghosal et al. used commonsense knowledge information based on the construction of multiple-gate recurrent units to assist in detecting emotions in conversations [72]. Inspired by them, we attempt to use commonsense knowledge inference to assist in detecting the emotions from the text.

Recently, PTM based on transformers has been shown to be effective in capturing contextual information [79]. Such language models can be classified into two categories depending on the different masking strategies: autoregressive models and autoencoder models. The autoregressive models, represented by Generative Pre-Training 2 [80], use unidirectional transformers to predict words. The autoencoder models, represented by bidirectional encoder representations from transformers (BERT) [22] use bidirectional transformers to integrate contextual information.

The PTM acquires prior information and applies it to downstream tasks by unsupervised learning on large amounts of data in advance. The autoencoder models are more suitable for classification tasks due to the information leakage. Kim et al. achieved good results in the conversation emotion recognition task by using a robustly optimized BERT pretraining approach [81]. We try to find how symbolic approaches affect emotion detection based on the BERT model in the experimental chapter.

Hybrid methods inherit the advantages and disadvantages of the other three kinds of methods. Shaheen et al. constructed annotations of sentences by rules and used the rule-based approach or k-nearest neighbors algorithm to classify according to the semantic similarity between the annotations and the sentences [82]. Different from them, we attempt to merge neural features with symbolic features and explore whether neural networks can achieve better results with the assistance of the emotional dictionary, commonsense knowledge inference and topic model clustering approaches than with the approach alone in this article.

2.2.2 Other Strategies

Contrastive learning used a self-supervised approach to avoid the cost of annotating large-scale datasets in the field of computer vision [83]. Contrastive learning constructs positive and negative samples by means of data augmentation and adjusts the distribution of features in space. Later,

contrastive learning was also gradually used in the field of natural language processing. The most commonly used methods of text data augmentation are replacement, insertion, deletion and swap [84]. Gao et al. used the property of dropout to implicitly construct positive samples and achieved state-of-the-art results on several semantic text similarity tasks [43]. Liang et al. added a regularization strategy to improve the robustness of the model based on the inconsistency of dropout output [85]. Inspired by the above work, we use dropout and construct two large-scale pre-trained models based on fine-tuning to explore the influence of contrastive strategies on emotion recognition.

Consistency training is often used for label-independent semi-supervised learning [86]. Miyato et al. added a regularization term to make the model more robust to input perturbations by an adversarial approach [46]. Gao et al. constructed positive samples implicitly in contrastive learning and achieve good results with the consistency training [43]. Liang et al. used consistency training to alleviate the output instability caused by dropout in the model [87].

Both prompt ensembling and dropout in the language model may perturb the probability distribution to be predicted. Inspired by the above work, we attempt to introduce consistency training into our supervised learning process, to improve the robustness of the model.

Unlike traditional supervised learning, prompt-based learning learns the predicted text probabilities directly with the help of language models [44]. According to the shape of the prompt, prompting methods can be classified into two categories: cloze prompts and prefix prompts. The cloze prompts fill in the blanks of the string by the language model [88]. The prefix prompts continues to generate the next strings [89]. The cloze prompts are more appropriate for the classification tasks. The method of creating a prompt can be automatic [90] or manual [91]. To save computational overhead, we try to create pormpt manually.

Since the semantics of the text includes more than just emotional information, it is not easy to separate multiple underlying emotions from the semantics extracted by the pre-trained language model. Experiments by Zhao et al. showed that prompt-based learning can improve the performance of emotion recognition models [92]. The authors used a language model to predict the most likely emotion. This method is not applicable to multi-label emotion recognition. In this paper, we attempt to manually design a new prompt for directly modeling all possible emotions that are present.

Lots of articles have demonstrated that the multi-prompt learning strategies can further im-

prove the efficacy of prompting methods. Depending on the different construction methods, there are four strategies: ensembling, augmentation, composition and decomposition [44]. Ensembling refers to using multiple similar unanswered prompts to make predictions [93]. Augmentation refers to demonstrating to the model how to provide answers with several additional answered prompts [94]. Composition is more suitable for composable tasks such as relation extraction [95]. Decomposition decomposes the prompt into multiple sub-prompts and is generally used for named entity recognition tasks [96]. We believe that the most suitable strategy for multi-label classification tasks is ensembling.

Chapter 3 Emotional Text Generation

3.1 Task Definition

The emotional response generation task can be formulated as follows:

Given a post $P_i = p_{i0}, p_{i1}, \cdots, p_{ik}$ and a kind of emotion E_i, $E_i \in \{$ "anger" , "disgust", "happiness" , "like" , "sadness" $\}$. The goal is to predict a response $R_i = r_{i0}, r_{i1}, \cdots, r_{in}$, $(r_{i0}, r_{i1}, \cdots, r_{in} \in C)$. C is the character vocabulary of the texts.

We propose a model called CERG. As is illustrated in Figure 1.1, the core of this model is 12 transformer blocks. We take the emotion E_i and the post P_i as the input. After initializing the parameters θ of the model f randomly, we concatenate the emotion E_i, the post P_i and the response R_i replaced by the "[MASK]" label in sequence. The sequence turns into the features after passing three embedding layers. The features are calculated by the transformer blocks and then turn into the hidden states. We try to train the model to minimize the cross-entropy loss function $l_{(\theta)} = -\sum_{r_j \in w} r_j \log f(e_0, p_0, \cdots, p_{L-1}, r_0, \cdots, r_{L-1}; \theta)$. The process of backpropagation $\theta = \theta - \eta(\partial l_{(\theta)}/\partial\theta$ makes θ approach the optimal value. When predicting, we adopt the hidden state where the first mask is located $h_{r_0}(\theta)$ to predict the first character of the response r_1. Then replace the first mask with the first character r_1 and continue to predict the second character r_2. Repeat the above process until the end symbol is predicted or the length of the response reaches the maximum length we set.

3.2 Methodology

3.2.1 Model for Text Generation

As is shown in Figure1.1, we put the emotion label in the first position, then concatenate it with the post and response. Unlike the baseline, emotion and text share the same embedding layers. The embedding layers consist of three parts. Token embedding is used to represent each character; position embedding is used to append the position of the character to the sentence; segment embedding is used to distinguish between post and response. In the input text, we adopt the "[SEP]" label to separate the post and the response. We adopt the "[MASK]" label instead of the current predicted position and the position after it to prevent information leakage.

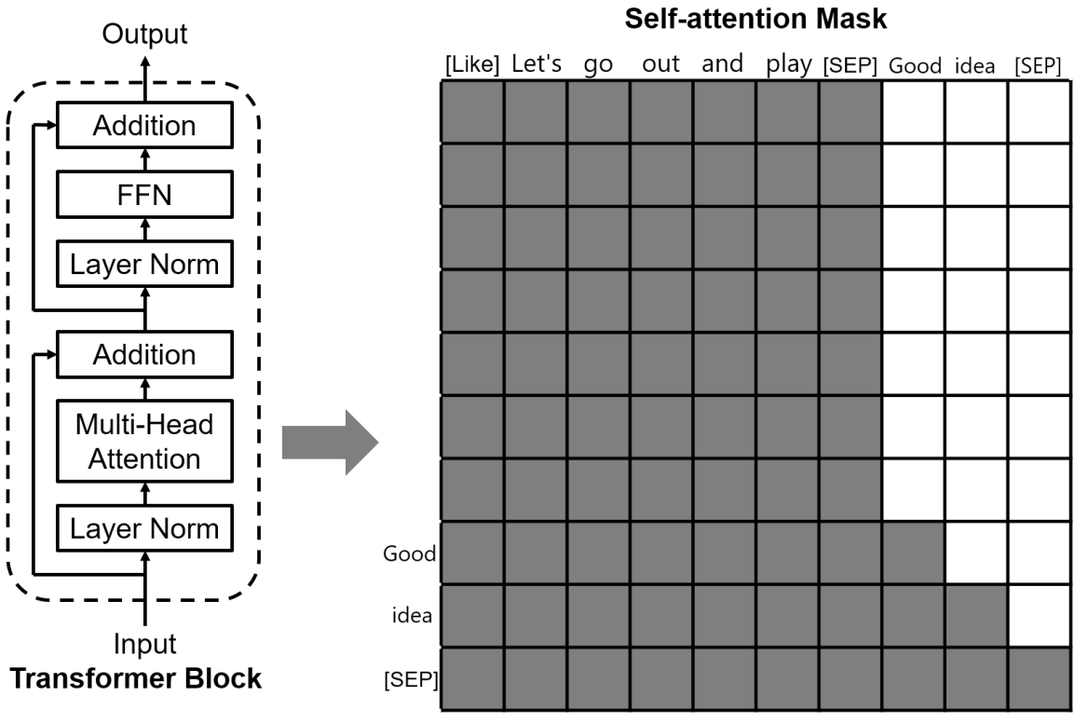

Figure 3.1 The left diagram is the structure of the transformer block, the right matrix is an example of the self-attention mask.

A transformer is a framework in which attention structure replaces loop structure. The traditional Transformer block consists of a multi-head attention layer and a feedforward neural network (FFN) as the core. The left side of Figure 3.1 shows that the layer normalization in each block is placed before the self-attention layer and the feed-forward layer. Xiong et al. pointed out that placing layer normalization in this way can reduce the dependence of the model on the warm-up optimizer during training [97].

The attention matrix is shown on the right side of Figure 3.1. Unlike traditional transformers, we have to prevent the input response from leaking information to the output response. We employ teacher-forcing technology to expand an n-character response into n responses. During training, the output of the current character position will be the next character.

We also try to add some regularization methods to recover overcorrection without disrupting parallel computing. Before training, we adopt the language model BERT to predict replacement characters at random positions in the input text [22]. The replacement augmentation method

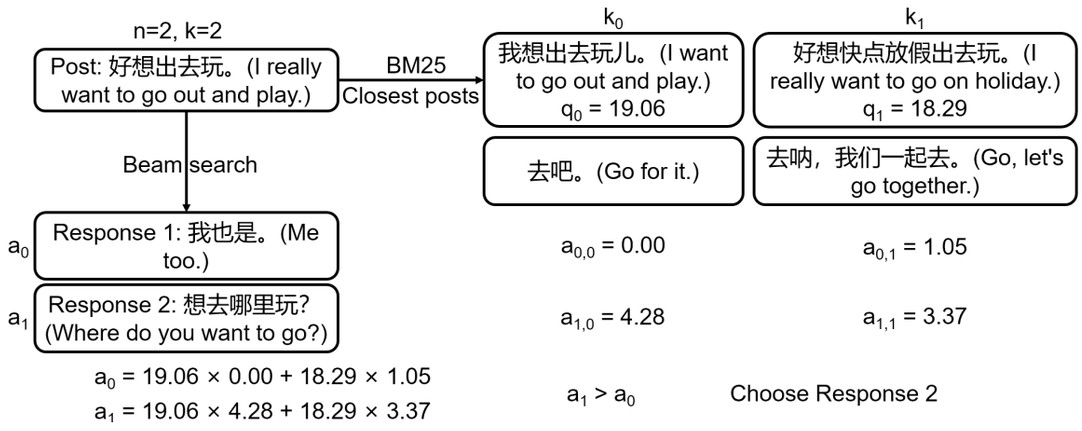

Figure 3.2 An example of using the retrieval method to select a better response in inference. The model predicts 2 candidate responses by beam search method. The response with a lower beam score has a higher retrieval score.

can help to improve the robustness of the model [98]. In case the model is difficult to converge due to the use of regularization methods at the beginning of training, we sample the replacement characters with decay from the ground truth characters.

3.2.2 Retrieval Strategy in Reranking

The retrieval method is applied in the inference process. We employ the beam search method to predict n responses. Then we adopt the BM25 method to find k posts that are closest to the input post in the training set, and calculate the similarity score $q_0, q_1, \cdots, q_{k-1}$. Next, we calculate the similarity score $a_{0,0}, a_{0,1}, \cdots, a_{0,k-1}$ between the first predicted response and the corresponding responses of the k posts. The weighted score of the first response is $a_0 = a_{0,0} \times q_0 + a_{0,1} \times q_1 + \cdots + a_{0,k-1} \times q_{k-1}$. Similarly, the weighted score of the nth sentence is $a_{n-1} = a_{n-1,0} \times q_0 + a_{n-1,1} \times q_1 + \cdots + a_{n-1,k-1} \times q_{k-1}$. Finally, we take the response with the highest weighted score as the output response. Experiments show that the general safe response cannot get high weighted scores here. This method can find out the responses that are more in line with the context of the posts, and increase the diversity of the responses.

For example in Figure 3.2, we employ beam search (beam size = 2) to predict two responses on the left. We adopt the BM25 method to retrieve the two nearest posts from the training set. Then, we compare the similarity between the predicted responses and the corresponding

responses of the retrieved posts. It can be seen from the comparison that, response 2 with a lower score in beam search obtains a higher weighted score. We choose response 2 as the final result.

3.3 Experiment Setups

3.3.1 Datasets

The data set we adopt in this article comes from the NTCIR-14 STC-3 CECG task, which contains more than 1.7 million Chinese Weibo post-response pairs. The data set has already been tokenized. Because the size of the vocabulary is too large for the model training, we re-tokenize the texts into characters. According to our statistics, there are about 0.3% of the texts exceeds 32 characters in length. Considering the training efficiency and possible information loss, we set the length of the training texts to 32 characters.

Besides, we preprocess the texts. We check the data and find that there are some sentences without Chinese characters. We do not use these sentences for training. We also remove the extra duplicate characters and retain 3 times at most.

There are 6 kinds of emotions in this dataset, including “anger”, “disgust”, “happiness”, “like”, “sadness” and “other”. The emotion labels are classified on the real replies of Chinese Weibo by a classifier with an accuracy of about 64%, which are for reference only. We regard the imbalance in the number of categories as the noise of the data set. As can be seen from the pie chart in Figure 3.3, the “anger” category has the least amount of data. This may be one of the reasons for the worst performance of the “anger” category. The “other” item can help the model to generate smooth sentences during the training process, but this emotion is excluded during the inference process.

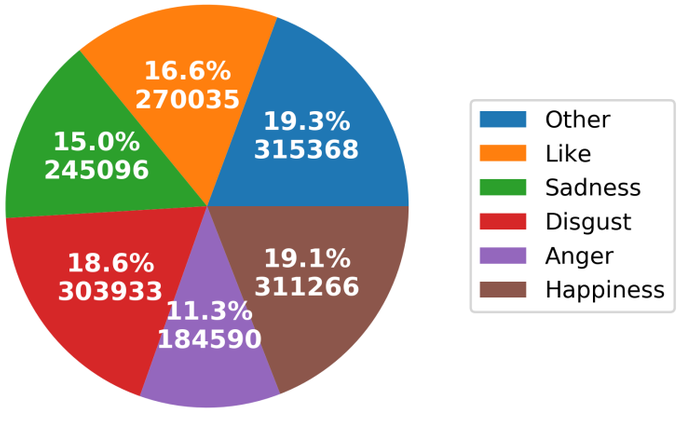

Figure 3.3 The distribution of different emotions in the data set.

3.3.2 Evaluation Metrics

Consistent with the NTCIR-14 STC-3 task, we adopt 200 posts and 5 emotions to predict 1000 responses. Existing generation task automatic evaluation metrics such as BLEU are not suitable for dialogue systems [99]. For example, here is a post: "Someone injured". According to the different contexts, "It is too pitiful" and "Who did it" are both reasonable responses. However, most of the automatic evaluation metrics calculate the similarity between the predicted sentence and the reference sentence through semantic or co-occurrence. We can find that not all reasonable responses can achieve high scores.

Therefore, the NTCIR-14 STC-3 task employs a manual evaluation method. If the predicted sentence is coherent and fluent, it can get the first point. On this basis, if the emotion of the sentence is consistent, it can get the second point. In this article, we adopt a similar but different scoring method. The deep learning generative models tend to predict safe and commonplace responses. In the experiment, we find that the reply using only emoji, "what's going on" and "me too", are 3 main types of responses with a large number and often context-free. These 3 types of responses will not be scored in our evaluation process. Table 3.1 is an example of our manual evaluation method.

Hard voting is a commonly used ensemble method [100]. We choose this method in manual evaluation. In addition, to verify the effectiveness of the retrieval method in our model, we made statistics and comparisons of the safe and commonplace responses.

Table 3.1 An example of manual evaluation.

Post:	Bless me to pass.	Emotion:	disgust
Response 1:	Me too.	Label 0	Not coherent or not fluent or a safe response
Response 2:	What exam?	Label 1	Coherent and fluent
Response 3:	You will fail!	Label 2	Coherent, fluent and emotion consistent

3.3.3 Baselines

We adopt the P&E2R model as the baseline in this article. There are a character embedding layer and an emotion embedding layer in this model. The posts and the responses share the weight through the character embedding layer. We encode the posts and responses separately by using two kinds of recurrent neural networks. The responses here are the predicted responses up to the last moment. The embedded emotions are concatenated with the hidden states of posts and responses. The decoder is another recurrent neural network. The decoding process is to predict the probability distribution of the next character based on the concatenated hidden states. This model achieved ranking second in manual evaluation.

3.3.4 Experimental Details

To balance efficiency and information loss, we set the maximum length of the posts and responses to 32. The size of the vocabulary is set to 13590. We set the embedding size and hidden size of the model to 768, which is consistent with the BERT-base model. We adopt 12 transformer blocks. The training experiment shows that the larger the ratio of augmentation methods, the more difficult it is for the model to converge, and the time cost will also become larger. As the training epoch increases, we gradually increase the augmentation rate to 5%. We use NVIDIA 2080ti GPU training with batch size = 128. It takes about 2.3 hours to train an epoch. The inference experiment shows that with the growth of the beam size and the retrieval k, the computational overhead becomes larger, but the improvement is not significant. The autoresponder needs to be timeliness. So we set these two parameters to 2.

3.4 Experimental Results

We compare the CERG model without the retrieval method and the full version of the CERG model with the baseline. The baseline results are taken from the responses we submitted to NTCIR-14. Tables 3.2 to 3.6 are the comparison results and the statistics about commonplace responses. The reason why the baseline gets lower scores than those published in NTCIR-14 is that we set all the safe and commonplace responses to label 0.

Table 3.2 The evaluation result and the safe-response statistic of like emotion.

Like	Label 0	Label 1	Label 2	Average	What' s going on?	Me too.	Only emoji	Proportion of safe responses
Baseline	142	53	5	0.315	0	10	95	0.525
No retrieval CERG	108	69	23	0.575	0	14	24	0.190
CERG	85	61	54	0.845	0	2	6	0.040

Table 3.2 shows the scores of the three models with the like emotion. The weighted average score of the model we proposed is 0.845, far exceeding the score of baseline. After we removed the retrieval method, our model also achieves a score of 0.575. Table 3.2 also shows the number of commonplace responses and their proportion in all responses. Nearly half of the responses generated by the baseline are emoji only. The emoji may express the respondents' emotions, but has little to do with the context. The responses of the "me too" class in the no retrieval CERG model are more than those of the baseline. However, the proportions' of commonplace responses drop significantly in the complete CERG model.

Table 3.3 is about the sadness emotion. The increase in label 2 more likely comes from label 1, which is different from that of like emotion. The proportion of commonplace responses is also less

Table 3.3 The evaluation result and the safe-response statistic of sadness emotion.

Sadness	Label 0	Label 1	Label 2	Average	What' s going on?	Me too.	Only emoji	Proportion of safe responses
Baseline	119	77	4	0.425	17	53	20	0.450
No retrieval CERG	104	83	13	0.545	4	27	56	0.435
CERG	99	55	46	0.735	1	9	7	0.085

than that of like emotion. The changes in the framework do not improve the result much, and the number of commonplace responses is similar. However, the use of retrieval method increase the weighted average score to 0.735, and the number of commonplace responses decreases greatly as well.

Table 3.4 The evaluation result and the safe-response statistic of disgust emotion.

Disgust	Label 0	Label 1	Label 2	Average	What' s going on?	Me too.	Only emoji	Proportion of safe responses
Baseline	132	67	1	0.345	54	0	19	0.365
No retrieval CERG	101	97	2	0.505	40	0	9	0.245
CERG	103	60	37	0.670	13	0	4	0.085

Table 3.4 shows the experimental results of the disgust emotion. We can see in the table, the generated responses are more coherent after we replace the framework. The emotional relevance of the responses also improves by using the retrieval method. Similar to the foregoing, the CERG model can reduce the proportion of commonplace responses in all the responses.

The experimental results of the anger emotion are shown in Table 3.5. The amount of training data of anger is the smallest. It might be the reason why the weighted average score of anger responses is lower than other emotions. Our model improves the average score to 0.625. There is no emoji in anger responses, and there are not many other commonplace responses. The CERG model still replaces most of these responses with more semantic and emotional responses.

Table 3.6 shows that there is a lot of emoji flood in happiness responses. We set responses containing the only emoji to label 0, so the score looks very low. Despite that, our CERG model raises the weighted average score to 0.755 and reduces the proportion of commonplace responses to 0.275.

Table 3.5 The evaluation result and the safe-response statistic of anger emotion.

Anger	Label 0	Label 1	Label 2	Average	What' s going on?	Me too.	Only emoji	Proportion of safe responses
Baseline	181	18	1	0.100	65	13	0	0.390
No retrieval CERG	135	59	6	0.355	16	37	0	0.265
CERG	92	91	17	0.625	1	4	0	0.025

Table 3.6 The evaluation result and the safe-response statistic of happiness emotion.

Happiness	Label 0	Label 1	Label 2	Average	What' s going on?	Me too.	Only emoji	Proportion of safe responses
Baseline	178	19	3	0.125	16	3	155	0.870
No retrieval CERG	140	49	11	0.335	10	11	108	0.645
CERG	85	79	36	0.755	6	2	47	0.275

3.5 Discussion

From the experimental results, we can conclude that the CERG model we proposed not only improves the speed of generating responses, but also improves the textual representation ability, making the responses more coherent and fluent. On the basis of that, we also add the retrieval method to further improve the semantic relevance and emotional relevance of the responses. From our statistics on commonplace responses, the retrieval method can increase the diversity of responses and avoid context-free responses.

The CERG model maintains the parallelism of calculation while reducing the impact of exposure bias and overcorrection. During the experiment, using the retrieval method at the beginning would make the model difficult to converge. Besides, when the proportion of character replacement increases, the loss value decreases slowly. Therefore, we adopt the teacher forcing method firstly, and gradually replace part of the characters with the augmentation method. This can improve the robustness of the model.

Due to the training efficiency, the retrieval method we employ only focuses on a single character, rather than focusing on the whole word. We will improve this retrieval method in the next step, like optimizing the collocation between the current word and the previous word.

The anger emotion takes up the least proportion in the training data, which may be the reason why the evaluation score is not as high as other emotions. From the commonplace responses analysis table, it can be seen that the response characteristics of each emotion are distinct. For example, the like emotion does not have “what’s going on” responses, and the disgust emotion does not have “me too” responses. This may be related to the preference of the training data. It also shows that if we put the emotion label in the first item of text for input, the model can effectively distinguish different emotions.

There are more than these types of commonplace responses. We do not list other categories that are not typical. As can be seen from Figure 3.4, the keywords in posts rarely appear in commonplace responses. Therefore, we can easily reduce the weight of this type of response by using retrieval methods, and sort more relevant responses before the commonplace response.

3.6 Summary

The dialogue system has always been one of the important topics in the domain of artificial intelligence. So far, most of the mature dialogue systems are task-oriented based, while non-task-oriented dialogue systems still have a lot of room for improvement. We propose a data-driven non-task-oriented dialogue generator "CERG" based on neural networks. This model has the emotion recognition capability and can generate corresponding responses. The data set we adopt comes from the NTCIR-14 STC-3 CECG subtask, which contains more than 1.7 million Chinese Weibo post-response pairs and 6 emotion categories. We try to concatenate the post and the response with the emotion, then mask the response part of the input text character by character to emulate the encoder-decoder framework. We use the improved transformer blocks as the core to build the model and add regularization methods to alleviate the problems of overcorrection and exposure bias. We introduce the retrieval method to the inference process to improve the semantic relevance of generated responses. The results of the manual evaluation show that our proposed model can make different responses to different emotions to improve the human-computer interaction experience. This model can be applied to lots of domains, such as automatic reply robots of social application.

The emotional dialogue system has user-friendly human-computer interaction capabilities and can be applied to many domains such as psychotherapy. In this work, we propose the CERG model for Chinese Weibo emotional response generation. We combine the retrieval method with this generative model to improve the contextual relevance and diversity of generated responses.

The data we adopt comes from the NTCIR-14 STC-3 CECG subtask. The data set contains 6 emotion categories and the corresponding 1.7 million Chinese Weibo post-response pairs. After concatenating emotion, post and response, we employ three embedding layers including token, position and segment embedding layers and 12 transformer blocks for representation. To train the model with the conventional optimizer, we adjust the position of the layer normalization in the transformer blocks.

In the training process, we mask the response part of the input text character by character to emulate the encoder-decoder framework to prevent the leakage of information during inference. We replace the characters with the BERT model predicted characters at random positions of

the input text, which will improve the robustness of the model without disrupting the training parallelism. We introduce retrieval methods in the inference process. We calculate the weight scores of similar posts and responses together with beam search, which can make the predicted responses more in line with the context.

We adopt a hard voting manual metric to evaluate the generative ability of our model. The coherence, fluency, and emotional relevance scores of our model in the manual evaluation are higher than the model without the retrieval method and the baseline model. The proportion of safe and commonplace responses has also been greatly reduced. These results show the effectiveness of our model. The model can be applied to social applications like Chinese Weibo automatic reply robots.

In the next step, we will pay more attention to the combination of retrieval methods and word collocations to further reduce exposure bias due to the replacement we used. The code of the CERG model is available on https://github.com/youngzhou97qz/Beam-Search-Retrieval.

Chapter 4 Multi-label Textual Emotion Detection

4.1 Task Definition

Before elaborating the details of the model, we first present some necessary notations. Since the text may contain several emotions, we denote the set of emotions as E:

$$E = (e_1, e_2, \cdots, e_i, \cdots, e_n) \tag{4.1}$$

where n is the number of emotion categories, and e_i denotes the state of the ith emotion. The emotional state is generally represented by continuous intensity values in the dataset. In this paper, we use the binary 1/0 labels to indicate the presence/absence of the emotions.

Similar to other natural language processing tasks, the process of textual emotion detection can be described as follows: given a piece of text X and the corresponding emotion labels E, the goal is to train a detection model f such that the distribution of the emotion labels mapped by the model is as close to E as possible:

$$Y = f(X) \tag{4.2}$$

where Y is the mapped multivariate Bernoulli distribution:

$$Y = (y_1, y_2, \cdots, y_i, \cdots, y_n) \tag{4.3}$$

4.2 Methodology

4.2.1 Encoder and Classifier for Detection

We construct a framework consisting of an encoder and a classifier.

For encoding, we use a large-scale pre-trained model, load the weights, and fine-tune them. In this paper, we use BERT as the encoder f_e. To be unified with BERT, each sentence is prefixed with a $[cls]$ token at first before it is fed into the model. BERT is designed with two pre-training tasks. The text involves the masked language modeling task, while the $[cls]$ token involves the next sentence prediction task. Thus, the first token of the output contains the sentence semantics to some extent. We use this token for classification.

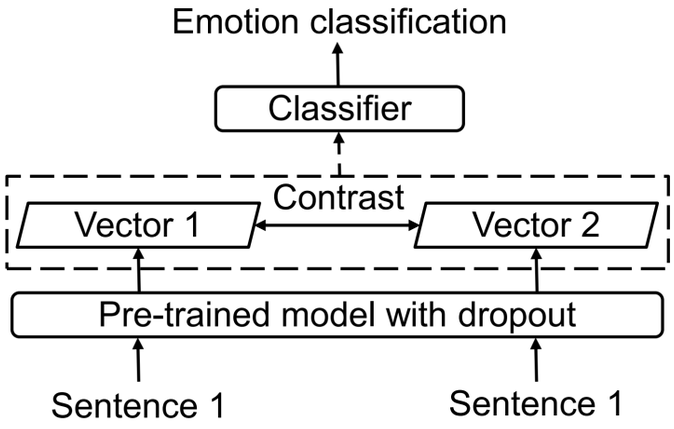

Figure 4.1 Overall framework of the model with contrastive strategy added to the encoding part.

We take the fully connected network with bias as a classifier f_c. Since the first token does not contain the sequence axis, we can directly map it to the emotion state e.

$$e = (e_1, e_2, ..., e_n) \tag{4.4}$$

in which n is is the number of emotion categories. When the output of emotion e_i exceeds the threshold, we believe that sentence s expresses the emotion e_i.

We introduce the contrastive strategy into the encoding and classification parts separately to get two different recognition models.

4.2.2 Contrastive Strategy in Encoding

Given a sentence s, we use an encoder to map s to a representation s'. Then, we use a classifier to transform s' into the emotion e.

The semantics represented by sentence vectors is the most critical part of emotion recognition. Different sentence vectors represented by [CLS] token in BERT have been confirmed to be very similar and do not represent the semantics well [41]. We try to enhance the semantic representation of sentence vectors with a contrastive strategy.

According to the description above, contrastive learning is a self-supervised training process, the same as what large-scale pre-trained models do. We continue to train the weights of the model with similar samples constructed by dropout on the basis of BERT. We expect that this weight for fine-tuning will improve the accuracy of emotion recognition.

As shown in Figure 4.1, we fine-tune the encoder before classification. The encoder predicts

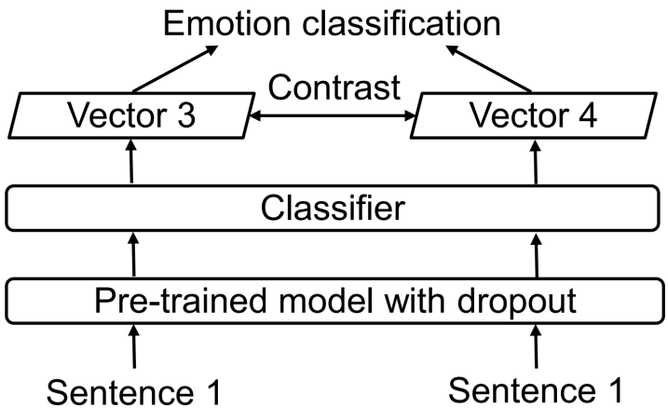

Figure 4.2 Overall framework of the model with contrastive strategy added to the classification part.

two similar representations Vector1 and Vector2 by dropout from the same input Sentence 1. Using unsupervised contrastive learning, the parameters of the encoder can be fine-tuned so that the distribution of the encoded results in the feature space is more favorable for emotion classification. Based on this, we add the classifier at the top for supervised training.

Contrastive learning adjusts the representation of features by decreasing the distance between similar samples and increasing the distance between different samples. Similar samples are generally transformed from the original sample, and we use the dropout method to transform in this article.

Dropout is a method commonly used to prevent overfitting of neural networks [101]. By dropping some neurons randomly during the training process, the model can be prevented from over-relying on some neurons and dependencies. Since dropout has randomness, the model will produce similar but not the same output each time with the same input. According to the property of dropout, it is easy to construct two similar samples.

In addition to self-supervised training, we can use the contrastive strategy for supervised training. Dropout only works at training time, i.e., the model is not consistent at training and prediction. To reduce the gap of the model between training and prediction, we close the distance between the outputs of the model after randomly dropping neurons.

Specifically, we take input during training twice through the model and add a Kullback-Leibler (KL) divergence between the two outputs. KL divergence is a metric used to estimate the

distance between distributions [102]. When calculating the loss function, adding KL divergence ensures that the model outputs are closer after each random drop of neurons to reduce the effect of dropout on the results.

As shown in Figure 4.2, we supervised train the classifier and the encoder at the same time. Due to dropout, the same input Sentence 1 has similar results Vector 3 and Vector 4 when prediction. For the stability of results, we introduce the KL loss function during training to make Vector 3 and Vector 4 as close as possible.

4.2.3 Prompt Engineering in Encoding

To detect the emotions in the text, our first thought is to create an emotion description prompt and concatenate it with the input text X. For example, if X is "You did a good job.", the emotion description prompt can be "I feel [MASK].". [MASK] is the token to be predicted in the masked-language model. Although this is a commonly used prompt in multi-class classification tasks, the emotion representation will be very difficult when a single [MASK] token is used for predicting multiple possible emotions.

Therefore, we use n emotion query prompts to separately predict the states for each emotion. For example, "Are you happy? [MASK]." can be used to predict the state of happiness. Since the subjects of the input texts may not be "you", in order to prevent semantic bias, we create prompts with only an adjective, such as "Happy? [MASK].", instead. Such prompts are short enough to avoid affecting the semantic representation of the input text X too much in the model in case there is are a large number of emotion categories.

In multi-label emotion recognition datasets, the emotion labels can often be used as prompts. Considering that the limited semantic coverage of a single emotion label may lead to biased predictions, we form a new set of labels with the synonyms of the original emotion labels. As shown in the top left of Figure 4.3, "happy" is the synonym of "joyful" in the new set. The prompt ensembling can complement the advantages of different prompts. Introducing the set of synonym labels improves the inclusiveness of the model for representing text samples that are slightly far away from the original emotion cluster center in the semantic space. The ablation experiments partially demonstrate the effectiveness of this approach.

In experiments, we find that if the number of texts containing "fear" in the dataset is small, and we ask the model to predict "fear" at a fixed position, then the model will be tuned to output

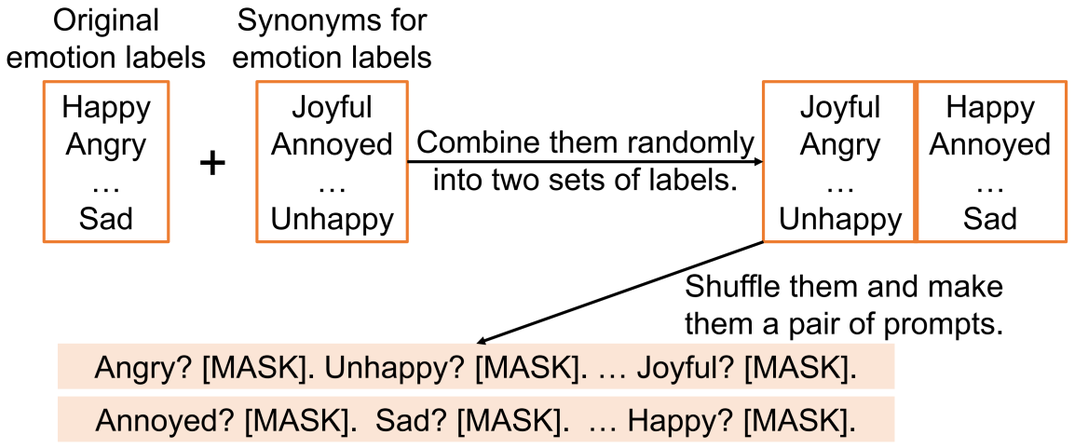

Figure 4.3 The overview of prompt engineering. The original emotion labels are first changed into synonyms, then they are randomly combined into two sets of labels, and finally they are shuffled into a pair of prompts.

0's (i.e., not fear) much more than 1's at this position. Although the prediction accuracy is high during training, the model continues to output 0 when the distribution of "fear" changes in the test set, resulting in a decrease in the accuracy during predicting. A reasonable interpretation is that the model might not really understand the semantics of "fear". To further improve the understanding ability of the model, we randomly combine the two sets of labels into a pair of labels (top right of Figure 4.3) and shuffle their order (bottom of Figure 4.3).

The random combination and shuffling are performed during training, rather than during the pre-processing of the training set. For each epoch during training, two different sets of prompts are concatenated to each input text. This allows the model to learn to understand the semantics of the [MASK] token in different positions and predict the presence of that emotion in different positions for each time the same input text is trained.

We denote the set of prompts for training as P. P can be placed in different positions of the input. For example, if the detection model is based on a unidirectional recurrent neural network, P is generally appended to the input text X at the end to avoid the risk of leaking information by the prompt in advance. Because the detection model we use in this paper is based on a bidirectional transformer, we put P in front of X:

$$Y = f(Concatenate(P, X)) \tag{4.5}$$

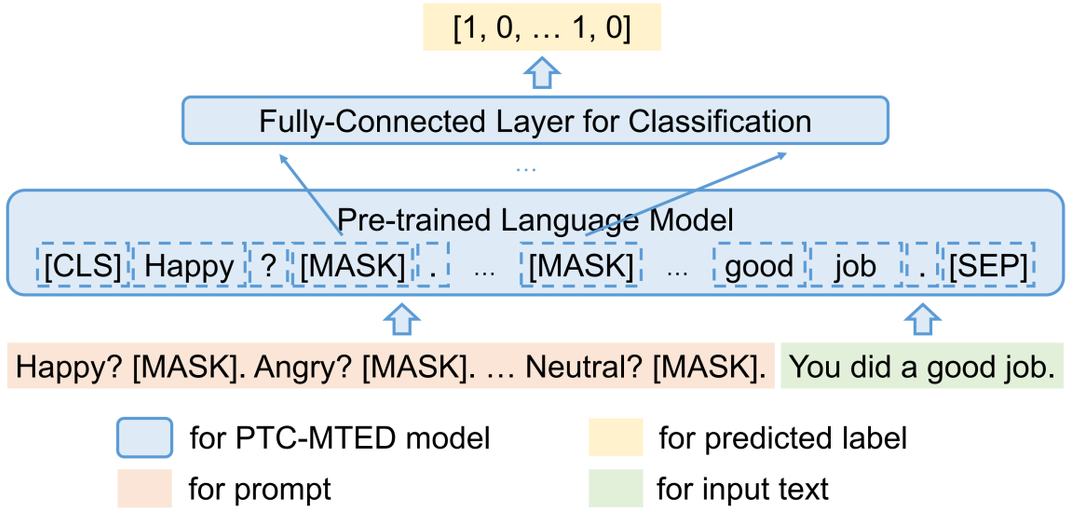

Figure 4.4 The overview framework of the prompt tuning with consistency training model. The input consists of a prompt and a text, and the output is a multi-label classification result.

In this way, there is no need for the model to additionally calculate the relative shift in the positions of [MASK] tokens in P due to the different sequence lengths of X.

4.2.4 Consistency Strategy in Training

The input after prompt is generally fine-tuned for downstream tasks by a pre-trained language model. We adopt an autoencoder pre-trained language model and a single-layer classifier as the emotion detection model.

For example, we use the BERT model as the language model. As shown in Figure 4.4, we add the [CLS] token in front of the prompted input and the [SEP] token at the end. This is the normal operation for BERT model input. The [CLS] token is used for the next sentence prediction task during pre-training, and is often used for sentence semantic representation in classification tasks during fine-tuning. We compare the classification fine-tuning method with [CLS] token to our proposed method in the experimental chapter. [SEP] token is used to distinguish two different sentences during pre-training. The calculation process of the pre-trained language model is not altered. After encoding, the input text with the added prompt is represented as a hidden state.

We take the hidden states of n [MASK] tokens in the prompt for classification. According to the language model, these hidden states can be predicted as “yes” or “no” that represent

positive or negative. We classify these hidden states with a fully-connected layer. Finally, the output is classified into two categories of 1 and 0 by means of a sigmoid activation function, or by setting a threshold. Since we randomly shuffle the emotions in the prompt part, we need to reorder the ground truth labels accordingly for calculating the loss value.

Based on the input of the same and different prompts, we propose two consistency training strategies.

The "same-consistency" training strategy is similar to R-Drop [87]. Two identical inputs with the same set of prompts will have slightly different outputs after passing through the model with dropout. Dropout is a competitive technique for preventing the model from over-relying on certain neurons by dropping the outputs of some neurons randomly [101]. During training, the randomness causes each forward pass to be calculated by a different sub-model, while during testing, all neurons in the model are involved in the computation. Therefore, we adopt the "same-consistency" training strategy to alleviate the inconsistency in the output distribution.

We denote the two distributions computed from the prompt P_1 and the input text X by the pre-trained model as Y_1 and Y_1'. Kullback-Leibler (KL) divergence is a metric used to estimate the distance between distributions [102]. The KL divergence is asymmetric. We use $D_{KL}(Y_1||Y_1')$ to represent the KL distance between the distributions Y_1 and Y_1'. In training, we add a loss function $L_{J(same)}$ to minimize the symmetrised KL divergence (or called Jeffreys divergence [103]):

$$L_{J(same)} = D_{KL}(Y_1||Y_1') + D_{KL}(Y_1'||Y_1) \tag{4.6}$$

There is another set of prompt P_2 for the same input text X. We denote the two distributions of P_2 and X as Y_2 and Y_2', and apply the "same-consistency" training strategy.

The "different-consistency" training strategy is used between the original labels and the synonyms. As mentioned in chapter 4.1.4, we employ the prompts with synonymous words as emotion labels to improve the generalization ability of the model. Although the two sets of prompts are different, we expect the model outputs to be the same. Therefore, we also add a loss function $L_{J(diff)}$ between the two different sets of prompts. For example, between Y1 and Y2:

$$L_{J(diff)} = D_{KL}(Y_1||Y_2) + D_{KL}(Y_2||Y_1) \tag{4.7}$$

As shown in Figure 4.5, for an input text X, we predict four results with a pair of prompts.

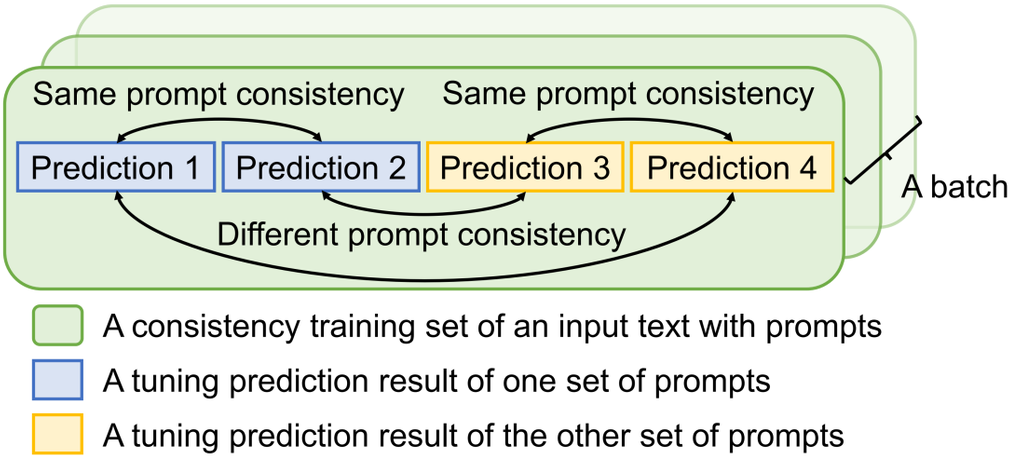

Figure 4.5 The overview of the consistency training strategies. A training data point yields four predictions for a pair of prompts represented by the blue box and the yellow box, respectively.

The loss function of consistency training can be summarized as:

$$\begin{aligned} L_J = & D_{KL}(Y_1||Y_1') + D_{KL}(Y_1'||Y_1) + \\ & D_{KL}(Y_2||Y_2') + D_{KL}(Y_2'||Y_2) + \\ & D_{KL}(Y_1||Y_2) + D_{KL}(Y_2||Y_1) + \\ & D_{KL}(Y_1'||Y_2') + D_{KL}(Y_2'||Y_1') \end{aligned} \tag{4.8}$$

4.2.5 Emotional Dictionary Matching in Preprocessing

Emotional keywords are very important factors in textual emotion detection, so our first thought is to find the emotional keywords in a text by retrieving them from the emotional dictionary. Since our experimental data are Chinese texts, we have to use a Chinese emotional dictionary. The Chinese Emotional Dictionary of Dalian University of Technology (DLUTE) contains seven kinds of emotions: "liked", "happy", "sad", "angry", "fearful", "disgusted", and "surprised" [104]. Compared with other open-source Chinese emotional dictionaries, this dictionary has more emotion categories and is more suitable for the dataset we selected.

Figure 4.6 shows the number of words for each emotion in the dictionary. The counts of "Liked" and "disgusted" are significantly larger than the remaining emotions. The authors explain that many nuanced categories are included in these categories. For example, words expressing respect, trust, praise, etc. are classified as "liked" and words expressing jealousy, suspicion, criticism, etc. are classified as "disgusted".

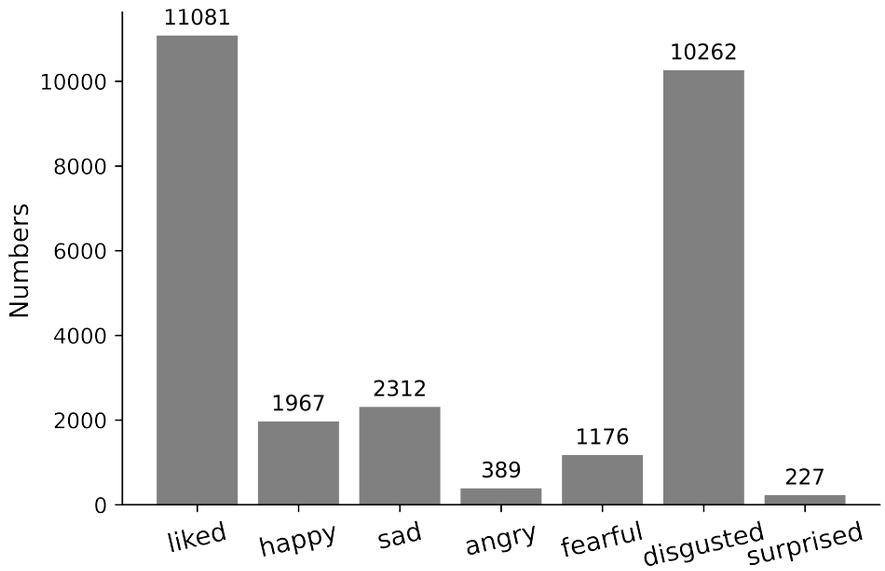

Figure 4.6 Histogram of the number of words for different emotions in the emotional dictionary.

Unlike English, we need to segment the words before processing Chinese text. We use the "Jieba" word segmentation tool here [*]. In the sentence "I love this movie", "love" is the word that belongs to the "liked" category. The model may predict this sentence as "liked". We believe that emphasizing the emotional keyword "love" can help the detection model, so we add it to the input text.

There may be more than one kind of emotion in the text, and there may be more than one keyword for each emotion. We list all the emotional keywords in front of the input text and separate them by commas. As shown in Figure 4.7, we match the words in the input text with the emotional dictionary and find four keywords. We concatenate these four keywords with the text and then detect the emotions in them:

$$Y = f(cat(D_w, X)) \tag{4.9}$$

where D_w is the emotional keywords and cat refers to concatenation. Since BERT is an autoencoder model, it does not matter where the emotional keywords are placed.

In addition, we believe that the number of different categories of emotional keywords in the text may also be related to the emotions. For example, the more keywords belonging to the

[*] https://github.com/fxsjy/jieba

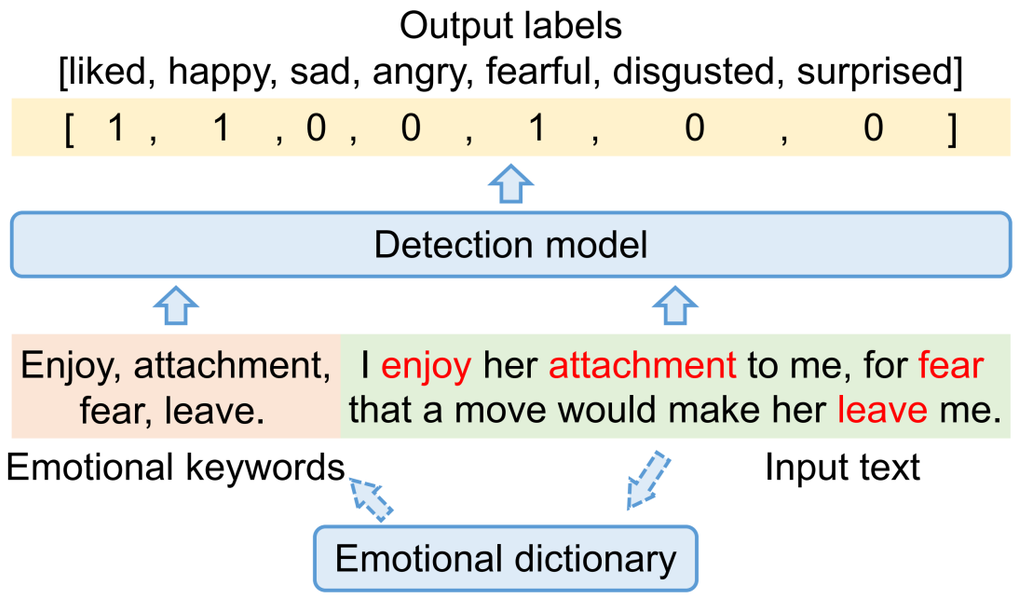

Figure 4.7 An example of matching keywords with an emotional dictionary and aiding emotion detection.

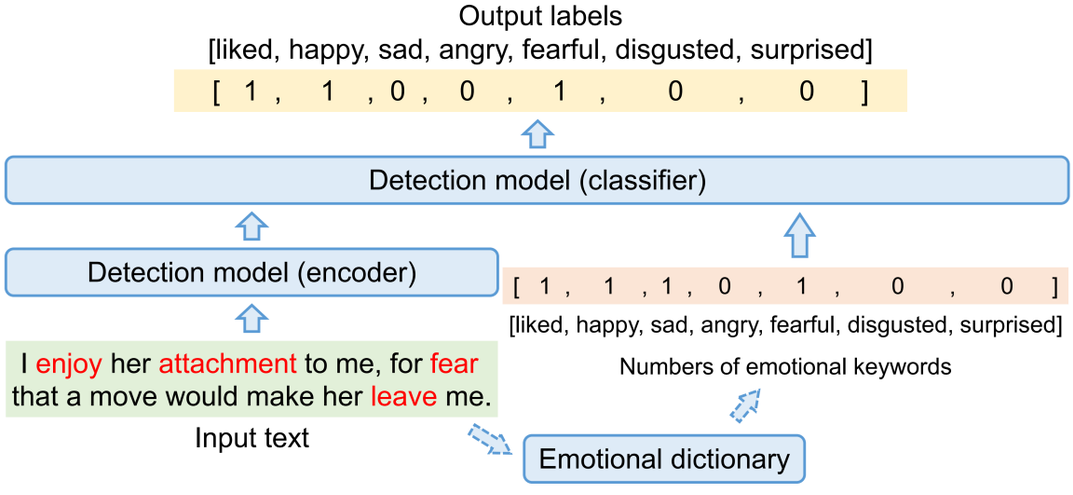

Figure 4.8 An example of converting the number of emotional words into a vector to aid emotion detection.

"liked" category in a text, the more possible that the text contains the emotion of "liked". We use a $1 \times n$ vector D_n to represent the numbers of all emotional keywords. As shown in Figure 4.8, "enjoy", "attachment", "fear" and "leave" in the input text correspond to the four emotions of "happy", "liked", "fearful" and "sad" in the emotional dictionary, respectively. We set these four emotions in the vector to 1 and other emotions to 0, and use the vector to assist in emotion detection. It is worth mentioning that n here refers to the number of emotion categories in the dataset.

However, different datasets have different emotion categories. The emotion categories of the dataset may not correspond to those of the emotional dictionary. For example, if some datasets have "expect" emotion but no such kind of emotional words in the dictionary, we set the number of the corresponding position in the vector to 0. If there is not any emotional keyword in the whole text, we set the "neutral" position (if available) in the vector to 1 and the number of all the other emotional words to 0.

Considering that adding several numbers to the text may affect the semantics of the input text, we concatenate the vector D_n with the hidden state of the encoder before classification.

$$Y = f_c(cat(f_e(X), D_n)) \tag{4.10}$$

where f_c is the classifier and f_e is the BERT encoder.

4.2.6 Commonsense Knowledge Inference in Preprocessing

We attempt to assist emotion detection by the knowledge inference approach. Commonsense transformers for automatic knowledge graph construction (Comet) is an open-source knowledge base built automatically based on commonsense knowledge graphs [105]. This knowledge base can infer the state and the behavior of the person in an event based on commonsense. For example, given the event "I want to see a movie.", the Comet can infer my reaction as "entertained". Although the textual descriptions do not have an obvious emotion, if we infer that "I see the movie" for "entertained" from commonsense, we can find that the emotion implied in the event may be "happy". This suggests that the predictions of Comet help detect emotions.

Comet has nine different types of predictions, including six predictions for himself/herself/themselves and three predictions for others. During pre-processing on the experimental datasets, we find that not every type of prediction has a meaningful outcome for

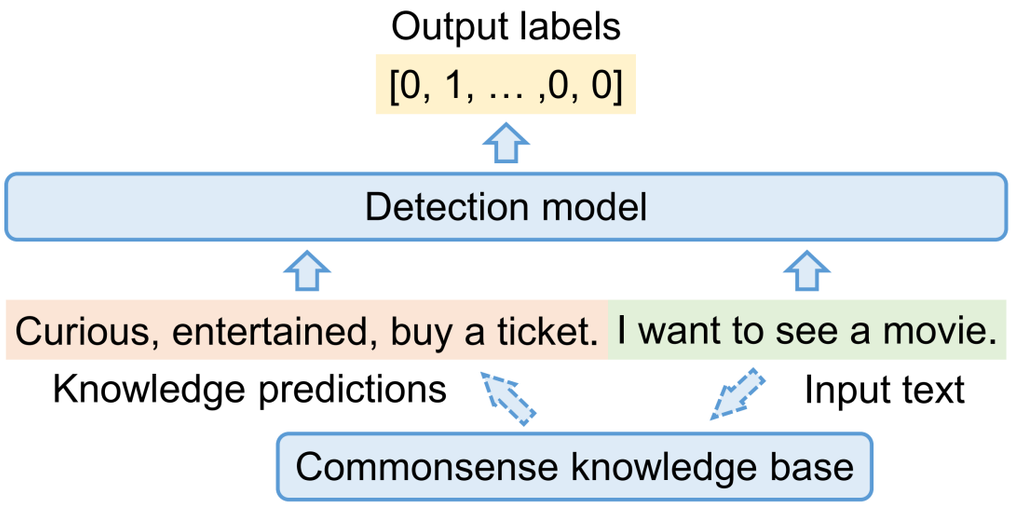

Figure 4.9 An example of using commonsense knowledge inference to aid emotion detection.

an event. In the previous example of seeing a movie, the Comet infers that others feel "nothing". The description of the subject, the reaction of the subject, and what the subject will do next, which have a high percentage of meaningful predictions, are selected for the experiments.

We put the inferred outcome before the input text and separate them with a period. If the Comet does not infer a meaningful outcome, we replace the outcome with a "none" word. When we use multiple predictions, the outcomes are separated by commas. As shown in Figure 4.9, the commonsense knowledge base describes my state as "curious", predicts my reaction as "entertained", and predicts that I will "buy a ticket" after the event "I want to see a movie." We concatenate these predictions together with the input text:

$$Y = f(cat(K, X)) \tag{4.11}$$

where K is the outcome inferred from the commonsense knowledge base Comet.

4.2.7 Topic Model Clustering in Preprocessing

We believe that semantically close texts may imply similar emotions. Therefore, we try to cluster the texts in the dataset by a topic model and add the topic categories to the emotion detection process. Topic to vector (Top2vec) is a model based on joint document and word semantic embedding to find topic vectors [106]. This model can represent semantic similarity based on the distance between the topic vectors and the document word vectors. Since clustering

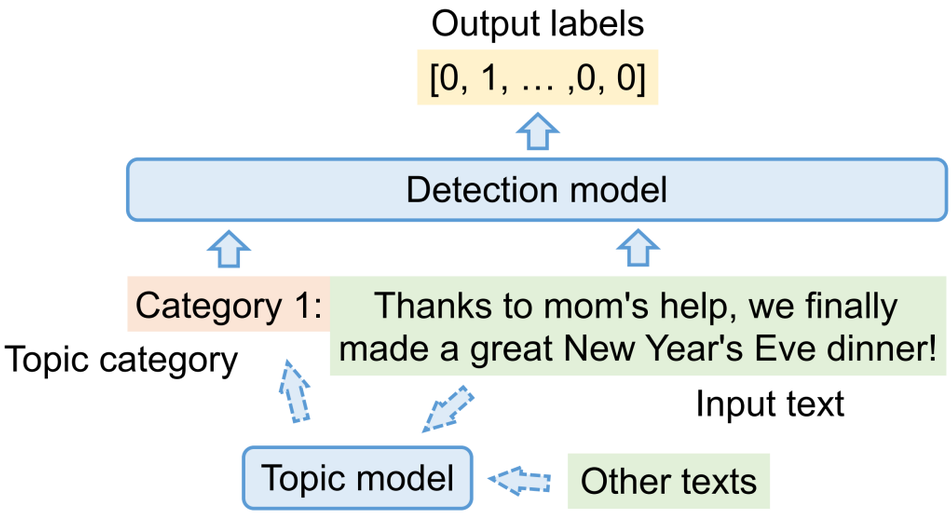

Figure 4.10 An example of using a topic category to aid emotion detection.

is unsupervised learning, this process includes data from the test set.

As can be seen from Figure 4.10, if a text is classified in the first category, then we prefix it with "Category 1:" before the text:

$$Y = f(cat(T_c, X)) \tag{4.12}$$

where T_c is the topic category number.

Considering that the category number is too abstract for neural networks, we try to replace the category number with the topic keywords of each category. More than half of the keywords extracted from English text are nones like "dollar" by the Top2vec model, while a significant proportion of the keywords extracted from Chinese text are adverbs like "actually". These adverbs have no obvious emotional meaning and are not beneficial to emotion detection. In contrast, adjectives and nouns may be more helpful for emotion detection.

We use the "Jieba" tool to tag the part-of-speech of these topic keywords. The adjectives and nouns remain. For each topic category, we select five words in ranking order as topic words. If there are less than five words, we use the co-occurrence graph method to extract keywords for supplement [107]. We separate these words with commas and place them before the input text.

As shown in Figure 4.11, the topic of the input text is related to the "family". We concatenate the clustered topic words with the text:

$$Y = f(cat(T_w, X)) \tag{4.13}$$

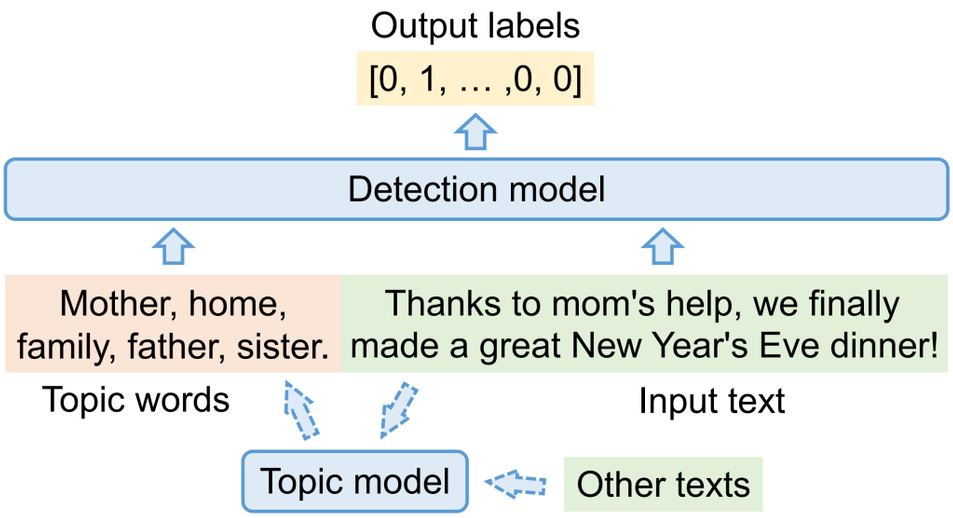

Figure 4.11 An example of using topic words to aid emotion detection.

where T_w are the topic words.

4.2.8 Multi-label Loss Function for Detection

In addition to the loss function for consistency training, we also use a multi-label classification loss function L_{MC}. In multi-label classification tasks, binary cross-entropy (BCE) can be used for each label if the label is independent of each other [108]. According to the statistics in the dataset chapter, the probability of co-occurrence between positive emotions tends to be higher than that between positive and negative emotions. That is, there is some kind of connection between emotion labels, which is not taken into account in the BCE computation.

Su [†]proposed a loss function that involves the computation of scores between labels. When the maximum value of the difference between all non-target class scores and target class scores is less than zero, it is guaranteed that each target class score is not smaller than each non-target class score. For example, if there are three emotions in E, with labels 1 for e_1 and e_2 and 0 for e_3, then when:

$$\max(y_3 - y_1, y_3 - y_2) < 0 \tag{4.14}$$

a threshold or mapping can always be found such that the target class scores of y_1 and y_2 are 1 and the non-target class score of y_3 is 0.

[†] https://www.spaces.ac.cn/archives/7359

We employ the loss function of Su as the loss function for multi-label classification. The experimental chapter demonstrates the effectiveness of this loss function.

4.3 Experiment Setups

4.3.1 Datasets

We select the following two multi-label emotion classification datasets for textual emotion detection.

The Ren-CECps dataset was obtained from the Chinese weblog containing 34,702 sentences including 27,008 training sentences and 7,694 test sentences. The texts in the Ren-CECps have at least 1 and at most 86 Chinese characters, with a median of 30 and a mean of 35.9. There are eight kinds of emotions annotated in the dataset: “love”, “anxiety”, “sorrow”, “joy”, “expect”, “hate”, “anger” and “surprise”. Emotional states are in range [0, 1]. We pre-label the emotional states greater than 0 as 1 and the others as 0. We label the sentences with all 0 emotion scores as “neutral”. The details of this dataset can be found in the article [109].

To better compare the differences between the experimental results and the ground truth, we calculate the correlation coefficient of the emotion of the test set in Figure 4.15. As can be seen from the figure, the correlation coefficient between “Love” and “Joy” is high, and the correlation coefficient between “Love” and “Sorrow” is low. It shows that the possibility of co-occurrence between “Love” and “Joy” is greater than that between “Love” and “Sorrow”. The correlation coefficients between “Surprise” and other emotions are almost close to zero. This indicates that

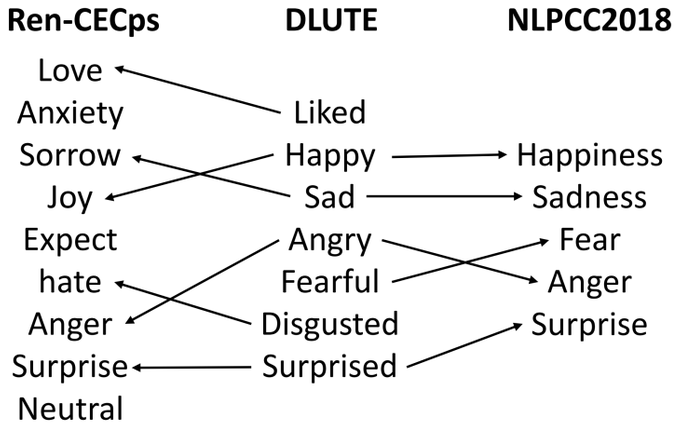

Figure 4.12 The emotion correspondence between DLUTE and the two datasets. The Ren-CECps dataset is on the left and the NLPCC2018 dataset is on the right.

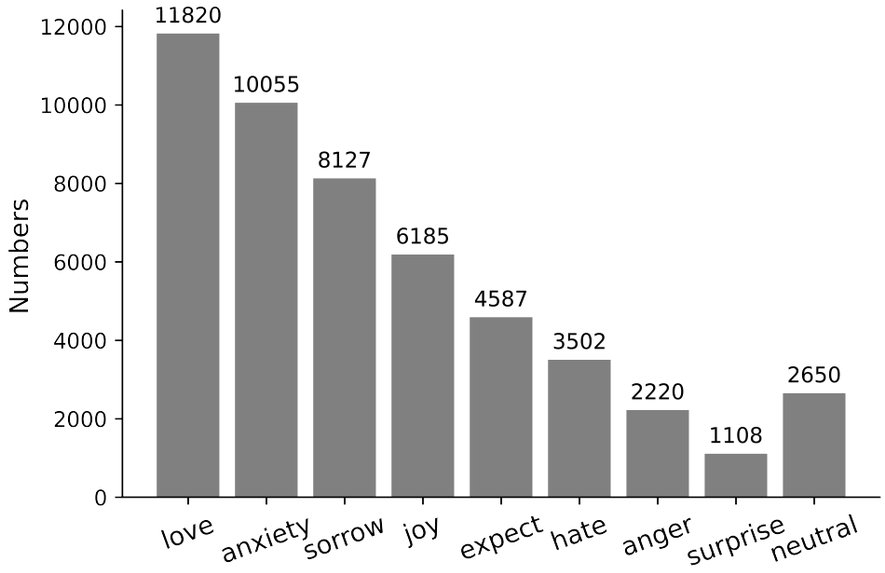

Figure 4.13 Histogram of the number of texts for different emotions in the Ren-CECps dataset.

there is no relevance between "Surprise" and other emotions.

The correspondence between the emotion categories of this dataset and those of the emotional dictionary DLUTE is shown on the left side of Figure 4.12. There is no emotional word of "anxiety" or "expect" in the Ren-CECps that can be matched from the dictionary. According to the correspondence, the "love/liked" has the largest quantity in the Ren-CECps and the dictionary, while the "surprise/surprised" has the smallest quantity. Figure 4.13 shows the number of texts for different emotions in the Ren-CECps dataset. The reason why "hate" has a low number in Ren-CECps but the corresponding "disgusted" has a high number in the DLUTE is that the "disgusted" also includes emotional words with jealous, skeptical and critical words. In the Ren-CECps, the percentage of meaningful predictions of "description", "reaction" and "next step" inferred by the Comet are 99.90%, 97.10% and 99.31%, respectively. The number of topics clustered by the Top2vec for this dataset is 160.

There are 6,728 sentences of training data and 1,200 sentences of test data in the NLPCC2018 dataset. The texts in the NLPCC2018 have at least 9 and at most 213 Chinese characters, with a median of 68 and a mean of 77.6. This dataset contains five kinds of emotions: "happiness", "sadness", "fear", "anger" and "surprise". As with the Ren-CECps dataset, we add the "neutral" to the labels. The details of this dataset can be found in the article [110].

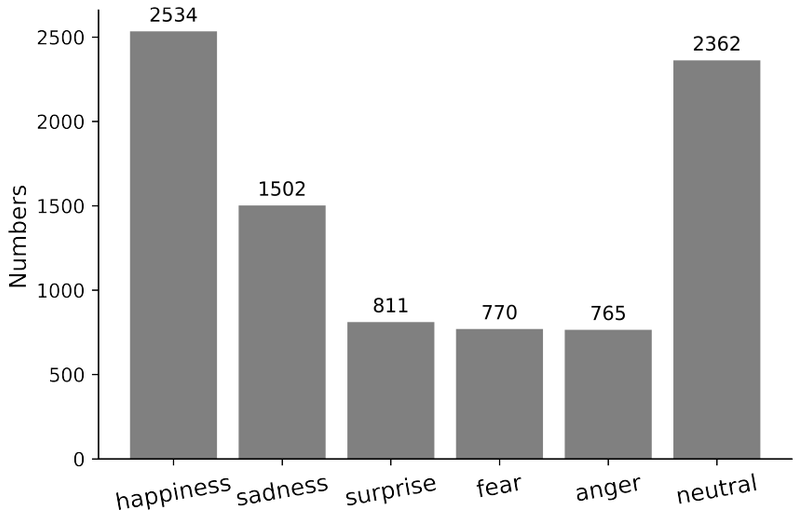

Figure 4.14 Histogram of the number of texts for different emotions in the NLPCC2018 dataset.

The correspondence between the emotion categories of the NLPCC2018 and those of the DLUTE is shown on the right side of Figure 4.12. The two major categories "liked" and "disgusted" in the DLUTE are not matched. Figure 4.14 shows the number of texts for different emotions in the NLPCC2018 dataset. In the NLPCC2018, the percentage of meaningful predictions of "description", "reaction" and "next step" inferred by the Comet are 96.49%, 92.72% and 94.37%, respectively. The number of topics clustered by the Top2vec for this dataset is 47.

4.3.2 Evaluation Metrics

Unlike the evaluation metrics for single-label classification tasks, the commonly-used accuracy does not reflect the effectiveness of multi-label classification models well. We use the following evaluation metrics.

F1 score is a combination of precision and recall. Micro F1 score is calculated for all categories as a whole. When micro F1 scores are used for imbalanced datasets, the error may be large. Macro F1 score is calculated separately for different categories with the same weight. This score is susceptible to extreme precision and extreme recall. Average precision (AP) is the average score of precision for recall from 0 to 1. For the above three metrics, higher values indicate better model performance.

Coverage error (CE) indicates the average number of labels that can cover the ground truth

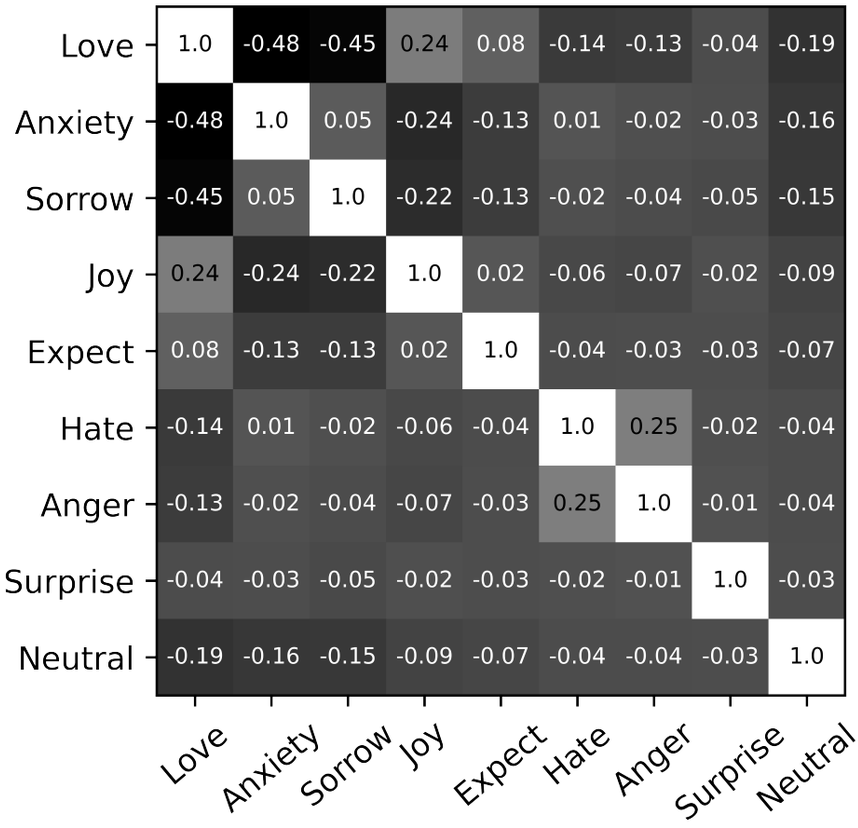

Figure 4.15 The emotional correlation coefficient matrix of the Ren-CECps test set.

labels. Ranking loss (RL) is the proportion of misclassifications after weighting by the number of relevant labels. For the above two metrics, lower values indicate better model performance.

4.3.3 Baselines

The multi-label emotion detection architecture from sentence (MEDA-FS) is a hierarchical model [111]. This model captures the underlying emotion-specified features through a multi-channel feature extractor and then predicts emotion sequences through an emotion correlation learner. MEDA-FS achieves the best-published results on the two datasets mentioned above. Other baseline models commonly used for multi-label classification, such as BP-MLL [112], are also compared in the article of MEDA-FS. We do not list the results in this paper.

In addition, we also try to compare with the classification fine-tuning method with [CLS] token. The [CLS] token of large-scale pre-trained models is often used to fine-tune downstream

tasks. But our proposed method does not use the [CLS] token as the semantic representation of the input text. It is worth mentioning that we also use the loss function of Su in the fine-tuning method with [CLS] token experiment since we do not make any modification to this loss function. More experiments related to [CLS] token or other semantic representations can be found in Appendix.

4.3.4 Experimental Details

In multi-label classification tasks, binary cross-entropy can be used for each label if the label is independent of each other [108]. However, the probability of co-occurrence between positive emotions tends to be higher than that between positive and negative emotions. That is, the connection between emotion labels is not taken into account in the binary cross-entropy computation.

We employ a loss function that involves the computation of scores between labels proposed by Su for multi-label classification [‡] When the maximum value of the difference between all non-target class scores and target class scores is less than zero, it is guaranteed that each target class score is not smaller than each non-target class score. For example, if there are three emotions in E, with labels 1 for e_1 and e_2 and 0 for e_3, then when:

$$\max(y_3 - y_1, y_3 - y_2) < 0 \tag{4.15}$$

a threshold or mapping can always be found such that the target class scores of y_1 and y_2 are 1 and the non-target class score of y_3 is 0.

We used the hugging face open source large-scale pre-training model BERT (chinese-bert-wwm-ext) as our backbone [113] [114]. When predicting, if the output result of an emotion is greater than the boundary threshold, we assume that the emotion is expressed.

For the model with contrastive strategy added to the encoding part, 0.1M news titles are used for contrastive learning to adjust the semantic expressiveness of sentence embedding. For the model with contrastive strategy added to the classification part, we add KL loss between the replicated data points in a batch.

The length of the input text is truncated at 100 except for the prompt part. The dropout rate is the default value of 0.1. We use the AdamW optimizer to train the model [115]. The learning

[‡] https://www.spaces.ac.cn/archives/7359

rate of the pre-trained language model is 1e-5. The learning rate of the classifier is 1e-4. We use layer normalization between the pre-trained language model and the classifier.

4.4 Experimental Results

Table 4.1 shows the results of the Ren-CECps dataset. MEDA-FS uses the feature extraction approach for downstream tasks. BERT FT refers to fine-tuning approach with BERT on the dataset. The BERT FT improves from 60.76/48.31 to 65.42/53.57 in f1 scores compared to MEDA-FS. CE model means that the weights are adjusted with the self-supervised contrastive learning approach before fine-tuning. This method makes the evaluation metrics worse, especially the macro f1 score of 45.80 which is even lower than that of the MEDA-FS. CC model refers to the addition of supervised contrastive training to fine-tuning. This method improves the f1 score from 65.42/53.57 to 66.16/53.75 compared to the BERT FT. CC model also performs better in AP, CE and RL metrics.

Table 4.2 shows the results of the NLPCC2018 dataset. The results are similar to those in Table 1.1. In terms of evaluation metrics, the fine-tuning approach outperforms the feature extraction approach to a certain extent. BERT FT achieves scores of 85.13 on the AP metrics, which is the best results in the comparison. Compared to BERT FT, the macro f1 score of the CE model decrease from 49.94 to 43.40. Compared to BERT FT, the macro f1 scores of the CC model improve to 50.57. CC model also achieves scores of 1.2775 and 0.1090 on the CE and RL metrics. It proves that the model with a contrastive strategy added to the classification part is more effective in the textual emotion recognition task than to the encoding part.

Table 4.1 The experimental results of contrastive strategies on the Ren-CECps dataset. In this paper, for the metrics of Micro F1, Macro F1 and AP, higher values indicate better performance; For the metrics of CE and RL, lower values indicate better performance.

	Micro F1	Macro F1	AP	CE	RL
MEDA-FS	60.76	48.31	76.51	2.2226	0.1062
BERT FT	65.42	53.57	81.99	1.9046	0.0871
CE model	61.78	45.80	79.58	2.0187	0.1008
CC model	**66.16**	**53.75**	**82.00**	**1.9021**	**0.0868**

Table 4.2 The experimental results of contrastive strategies on the NLPCC2018 dataset.

	Micro F1	Macro F1	AP	CE	RL
MEDA-FS	**63.02**	49.42	77.12	1.7288	0.1681
BERT FT	59.05	49.94	**85.13**	1.2883	0.1114
CE model	60.67	43.40	84.74	1.2958	0.1137
CC model	58.68	**50.57**	85.09	**1.2775**	**0.1090**

Table 4.3 The ablation experimental results of prompt consistency on the Ren-CECps dataset.

	Micro F1	Macro F1	AP	CE	RL
MEDA-FS	0.6076	0.4831	0.7651	2.2226	0.1062
CLS-FT	0.6542	0.5357	0.8199	1.9046	0.0871
PTC-MTED	**0.6627**	**0.5432**	**0.8233**	**1.8851**	0.0850
-shuffle	0.6509	0.5209	0.8187	1.9056	0.0869
-random	0.6574	0.5314	0.8221	1.8858	**0.0847**
-synonym	0.6554	0.5313	0.8159	1.9240	0.0899
-$L_{J(diff)}$	0.6561	0.5384	0.8189	1.8990	0.0865
-$L_{J(same)}$	0.6549	0.5314	0.8196	1.9062	0.0871
-L_{MC}	0.6550	0.5269	0.8161	1.8963	0.0861

Table 4.3 shows the experimental results of each evaluation metric on the Ren-CECps dataset. MEDA-FS is a BERT-based feature extraction model, and the experimental results are directly taken from the original paper. The [CLS] token fine-tuned baseline model (CLS-FT) refers to the result of the fine-tuning on the [CLS] token by the same pre-trained language model as PTC-MTED with the loss function of Su. This model substantially outperforms the results of the MEDA-FS model in the evaluation of all five metrics. Our proposed PTC-MTED model

achieves scores of 0.6627/0.5432/0.8233/1.8851/0.0850 on Micro F1/Macro F1/AP/CE/RL metrics, respectively. The scores of the PTC-MTED model on these metrics are further improved compared to the CLS-FT model, which demonstrates the effectiveness of our method.

The last six rows of Table 4.3 show the ablation experiments for each component of the PTC-MTED model. "-shuffle" means that the order of the emotions in the prompt set is not shuffled. "-shuffle" is lower than PTC-MTED on Micro F1 score by 0.0118. "-random" means that the original emotions and synonyms in a pair of prompts are not randomly exchanged. "-random" surpasses PTC-MTED in RL scores, and only slightly trails in other metrics. "-synonym" means that only original emotion labels are used as the prompt. Since there is no pair of prompt sets, the "different-consistency" training strategy cannot be adopted. "-synonym" is the worst performing model in ablation experiments on AP/CE/RL metrics. "-$L_{J(diff)}$" and "-$L_{J(same)}$" refer to training without different and same consistency strategies, respectively. "-$L_{J(diff)}$" and "-$L_{J(same)}$" are lower than PTC-MTED in Macro F1 score by 0.0048 and 0.0118. "-L_{MC}" refers to replacing Su's loss function with BCE. "-L_{MC}" is lower than PTC-MTED on AP score by 0.0072. The ablation experiments prove that each of the proposed components contributes to the final performance of the PTC-MTED model.

Table 4.4 shows the experimental results of each evaluation metric in the NLPCC2018 dataset. The models to which the names refer are the same as in Table 4.3. CLS-FT performs better than MEDA-FS in AP/CE/RL metrics, but does not have an advantage in F1 scores. Our PTC-MTED achieves scores of 0.6401/0.5269/0.8755/1.2275/0.0959 on Micro F1/Macro F1/AP/CE/RL metrics, respectively. Compared with the models above, our proposed model shows significant improvements in all metrics, indicating that our method is more effective.

In the ablation experiments in Table 4.4, "-shuffle" is 0.0194 lower than PTC-MTED in Micro F1 score. "-random" is still close to PTC-MTED in F1 metrics. "-synonym" is 0.0156 lower than PTC-MTED in AP score. "-$L_{J(diff)}$" and "-$L_{J(same)}$" are 0.0063 and 0.0145 lower than PTC-MTED in Macro F1 score. "-L_{MC}" is the almost worst performing model in ablation experiments on five metrics. Each component in the ablation experiment affects the performance of the model more or less.

Table 4.5 and Table 4.6 show the experimental results for each evaluation metric on the RenCECps dataset and the NLPCC2018 dataset, respectively. BERT-FT refers to the results of fine-tuning the [CLS] token based on the BERT model. We regard it as the baseline model for

Table 4.4 The ablation experimental results of prompt consistency on the NLPCC2018 dataset.

	Micro F1	Macro F1	AP	CE	RL
MEDA-FS	0.6302	0.4942	0.7712	1.7288	0.1681
CLS-FT	0.5905	0.4994	0.8513	1.2883	0.1114
PTC-MTED	**0.6401**	**0.5269**	**0.8755**	**1.2275**	**0.0959**
-shuffle	0.6207	0.5233	0.8639	1.2525	0.1024
-random	0.6245	0.5241	0.8631	1.2600	0.1040
-synonym	0.5958	0.5225	0.8599	1.2717	0.1069
-$L_{J(diff)}$	0.6197	0.5206	0.8600	1.2583	0.1042
-$L_{J(same)}$	0.5948	0.5124	0.8544	1.2875	0.1110
-L_{MC}	0.5812	0.4883	0.8441	1.3367	0.1233

comparison experiments here.

BERT+D_w refers to the results of detecting emotions with the aid of emotional words matched in DLUTE. BERT+D_n refers to the results of detecting emotions with the aid of the vector with the number of emotional words. BERT+D_{wn} refers to the results of using both of the approaches. The micro F1 scores of BERT+D_{wn} on both datasets are 0.0069 and 0.0432 higher than the baseline model respectively, indicating that the emotional dictionary approach helps the BERT model to detect emotions.

BERT+K_d refers to the results of using Comet inferred descriptions to aid in emotion detection. BERT+K_r refers to the results of using Comet inferred reactions to aid in emotion detection. BERT+K_n refers to the results of using Comet inferred next behaviors to aid in emotion detection. Multiple subscripts refer to experimental results where multiple inferences are used simultaneously. The AP scores of BERT+K_{drn} on both datasets, which are 0.0024 and 0.0117 higher than the baseline model, indicate that the commonsense knowledge inference approach helps the BERT model to detection emotions.

Table 4.5 The experimental results of neuro-symbolic approaches on the Ren-CECps dataset.

	Micro F1	Macro F1	AP	CE	RL
BERT-FT	0.6542	0.5357	0.8199	1.9046	0.0871
BERT+D_w	0.6615	0.5359	0.8194	1.8868	0.0851
BERT+D_n	0.6537	0.5280	0.8190	1.8949	0.0858
BERT+D_{wn}	0.6611	0.5304	0.8191	1.8901	0.0852
BERT+K_d	0.6624	0.5394	0.8215	1.8846	0.0847
BERT+K_r	0.6595	0.5340	0.8201	1.8845	0.0845
BERT+K_n	0.6591	0.5363	0.8203	1.8938	0.0858
BERT+K_{dr}	0.6594	0.5303	0.8225	1.8860	0.0844
BERT+K_{dn}	0.6612	**0.5425**	0.8212	1.8785	0.0838
BERT+K_{rn}	0.6618	0.5296	0.8257	1.8730	0.0831
BERT+K_{drn}	0.6622	0.5346	0.8223	1.8799	0.0835
BERT+T_c	0.6559	0.5371	0.8159	1.9010	0.0871
BERT+T_w	0.6609	0.5364	0.8206	1.8972	0.0862
BERT+T_{cw}	0.6620	0.5378	0.8228	1.8858	0.0845
B$D_{wn}K_{drn}$	**0.6652**	0.5418	0.8255	1.8735	0.0830
B$D_{wn}T_{cw}$	0.6591	0.5309	0.8211	1.8822	0.0842
B$K_{drn}T_{cw}$	0.6603	0.5323	0.8216	1.8853	0.0843
B$D_{wn}K_{drn}T_{cw}$	0.6633	0.5407	**0.8260**	**1.8706**	**0.0824**
all-LQ	0.3444	0.0699	-	-	-
w/o-text	0.5584	0.4269	-	-	-

Table 4.6 The experimental results of neuro-symbolic approaches on the NLPCC2018 dataset.

	Micro F1	Macro F1	AP	CE	RL
BERT-FT	0.5905	0.4994	0.8513	1.2883	0.1114
BERT+D_w	0.6108	0.5273	0.8605	1.2633	0.1055
BERT+D_n	0.6138	0.5207	0.8609	1.2583	0.1044
BERT+D_{wn}	0.6337	0.5118	**0.8699**	1.2500	0.1019
BERT+K_d	0.6295	0.5330	0.8636	1.2600	0.1044
BERT+K_r	0.6165	0.5236	0.8629	1.2425	0.0996
BERT+K_n	0.6114	0.5132	0.8597	1.2592	0.1042
BERT+K_{dr}	0.6191	0.5216	0.8654	1.2400	0.0993
BERT+K_{dn}	0.6043	0.5219	0.8555	1.2650	0.1061
BERT+K_{rn}	0.6202	0.5270	0.8648	1.2450	0.1009
BERT+K_{drn}	0.6276	0.5295	0.8630	**1.2385**	**0.0990**
BERT+T_c	0.6115	0.5310	0.8528	1.2783	0.1090
BERT+T_w	0.6108	0.5265	0.8530	1.2642	0.1066
BERT+T_{cw}	**0.6378**	0.5338	0.8602	1.2503	0.1020
B$D_{wn}K_{drn}$	0.6242	0.5231	0.8633	1.2492	0.1021
B$D_{wn}T_{cw}$	0.6306	0.5261	0.8635	1.2527	0.1025
B$K_{drn}T_{cw}$	0.6137	**0.5339**	0.8603	1.2500	0.1018
B$D_{wn}K_{drn}T_{cw}$	0.6140	0.5266	0.8610	1.2392	0.0998
all-LQ	0.4450	0.1160	-	-	-
w/o-text	0.5136	0.3642	-	-	-

BERT+T_c refers to the results of using clustered topic categories to aid in emotion detection. BERT+T_w refers to the results of using clustered topic words to aid in emotion detection. BERT+T_{cw} refers to the results of using both of the above approaches. The micro F1 scores of BERT+T_{cw} on both datasets are 0.0078 and 0.0473 higher than the baseline model respectively, indicating that the topic model clustering approach helps the BERT model to detect emotions.

We mix the above three approaches and combine them with the BERT model. $BD_{wn}K_{drn}$ achieves the best result with the micro F1 score on the Ren-CECps dataset. $BK_{drn}T_{cw}$ achieves the best result with the macro F1 score on the NLPCC2018 dataset. $BD_{wn}K_{drn}T_{cw}$ achieves the best results with AP, CE and RL scores on the Ren-CECps dataset.

All-LQ refers to the results of setting all predictions to the emotion that accounts for the largest quantity of the dataset ("love" in Ren-CECps and "happiness" in NLPCC2018). W/o-text refers to removing text from $BD_{wn}K_{drn}T_{cw}$ and only using symbolic features to detect emotions. W/o-text has much higher experimental results than all-love/all-happiness but is slightly inferior to BERT-FT, indicating that the symbolic features contain emotional information.

4.5 Discussion

4.5.1 About Contrastive Strategy

Compared to feature extraction, fine-tuning has better performance in downstream tasks. The large-scale pre-trained model utilizes a large corpus for self-supervised learning, and this corpus may deviate a little from the corpus used for the downstream task. The fine-tuning approach adaptively adjusts the model weights to make it more accurate in feature representation for downstream tasks. This is reflected in both Table 4.1 and Table 4.2. However, fine-tuning requires supervised learning, so the computational overhead is greater.

The self-supervised contrastive learning is adopted before fine-tuning, but the results become worse. We intended to represent the sentence semantics that was slighted in the pre-training task better by contrastive learning. However, this approach was not effective enough in the textual emotion classification task. The CE model does not perform well in all metrics for both datasets.

We originally thought the possible reason was that emotion classification did not depend entirely on sentence semantics. However, after fine-tuning by replacing the $[cls]$ token with pooled word vectors, the results slightly decreased. This suggests that sentence semantics plays a bigger role in emotion classification than word embeddings. Therefore, the reason for the worse results may be that the semantics compared to contrastive learning is not exactly related to emotion. That is, a better comparison scheme may improve the results.

Another possible reason is the inconsistent training goals of the model at different stages. The CE model focuses on sentence similarity rather than classification when encoding. Compared to the CE model, the CC model has better performance because it adds the comparison of hidden states similarity to the classification part.

We tried fine-tuning the pre-contrastive training with data from the unlabeled training set. We combined the textual parts of Ren-CECps and NLPCC2018 (excluding labels) for unsupervised training. The results were worse than the classification results obtained with the additionally found title data. This indicates that the self-supervised learning corpus affects the training results. A more general, larger corpus facilitates parameter fine-tuning during unsupervised learning.

The contrastive training on the same sample representation is adopted during fine-tuning and the results are improved. This suggests that the contrastive strategy used in models with dropout is beneficial to the textual emotion classification task.

We experimented with the effect of different drop rates on the results. In this kind of task, the classification results get worse as the drop rate increases. We believe that dropout is a regularization method, so the larger the drop rate, the stronger the constraint on the model and the more unstable results predicted by the model. Under the contrastive strategy, stability training becomes more difficult as the drop rate increases. If the amount of data is large enough, a larger drop rate may have a better performance. However, in our experiment, the best result is achieved by the BERT model with a default rate of 0.1.

We also tried both contrastive learnings before and during fine-tuning. The results are better than the CE model but worse than the CC model. We believe that the two contrastive strategies are independent and do not affect each other. Overall, the strategy of contrastive training during fine-tuning is a better choice for the textual emotion classification task.

4.5.2 About Fine-tuning

Figure 4.16 shows the emotional correlation coefficient matrix of the Ren-CECps test set predicted by PTC-MTED. Compared to Figure 4.15, the positive/negative correlations between emotions do not change substantially. In terms of degree, the biggest change is that the correlation between the two negative emotions of "Anxiety" and "Sorrow" increased from +0.05 to +0.57. Taking "Love" as an example, its negative correlation degrees with "Anxiety", "Sorrow" are slightly higher than the ground truth, and its positive correlation degrees with "Joy", "Expect" are also slightly higher than the ground truth, and the correlation degrees with other emotions do not change significantly. The significantly correlated emotions learned by PTC-MTED are "Anxiety" and "Sorrow" (+0.57), "Hate" and "Anger" (+0.40), "Love" and "Joy" (+0.36). These results are consistent with the ground truth, indicating that PTC-MTED has learned the correlation between emotions.

Table 4.7 shows the confusion matrices evaluated on the NLPCC2018 test set predicted by PTC-MTED. "True" denotes the number of ground truth labels, and "Pred" denotes the number of predicted labels. The label numbers for "Anger", "Fear" and "Surprise" count for a very small proportion, which is less than 10% of the total number of emotion labels in the test set. The

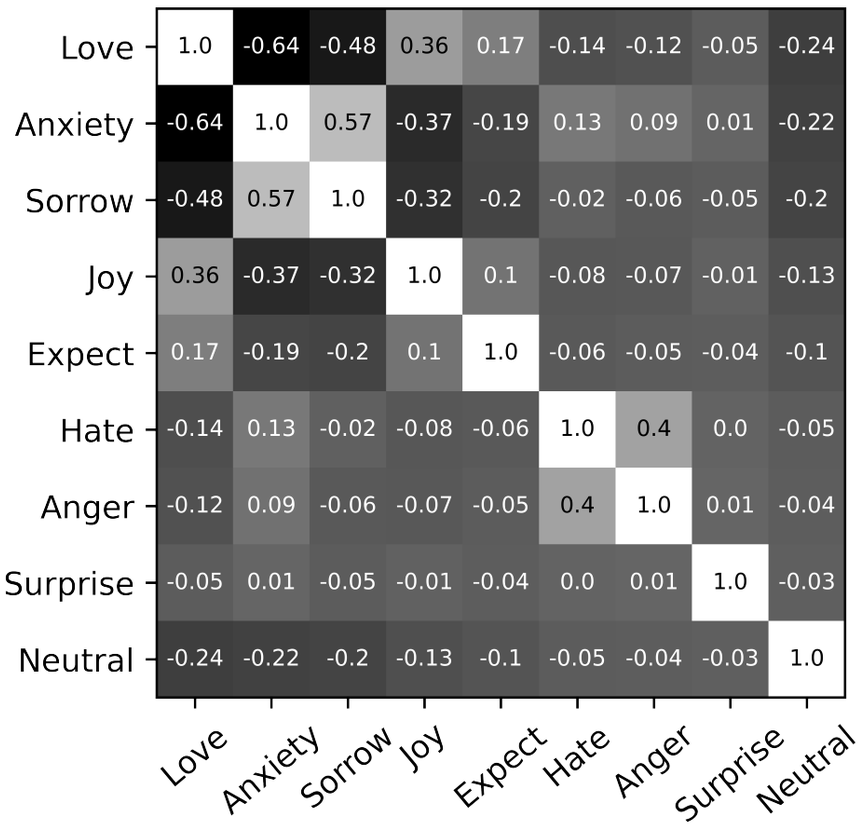

Figure 4.16 The emotional correlation coefficient matrix learned by PTC-MTED.

false negative predictions of PTC-MTED are still high on these minority emotion categories. In contrast, PTC-MTED performs much better on the majority emotion categories, such as "Happiness". This indicates that our proposed model has a certain learning ability when the data is imbalanced.

From the experimental results on both datasets, the BERT-based fine-tuning model outperforms the BERT-based feature extraction model. CLS-FT is only lower than MEDA-FS on the Micro F1 metric of the NLPCC 2018 dataset. This indicates that although the feature extraction method has a small number of parameters, it does not fit the downstream task as well as the fine-tuning method. The language model has been pre-trained on a very huge corpus, while the downstream emotion classification datasets that our model learns on are smaller in size and different in content. The fine-tuning method can adjust the weight of the model according to the corpus of the downstream task, but it also has a larger computational overhead.

Table 4.7 The confusion matrices for six emotions on the NLPCC2018 test set. True represents the ground truth, and Pred represents the predicted result.

	Happiness		Sadness		Anger	
	Pred 0	Pred 1	Pred 0	Pred 1	Pred 0	Pred 1
True 0	630	80	830	74	1098	34
True 1	148	342	137	159	33	35

	Fear		Surprise		Neutral	
	Pred 0	Pred 1	Pred 0	Pred 1	Pred 0	Pred 1
True 0	1064	25	1099	64	683	272
True 1	56	55	26	11	38	207

CLS-FT uses the [CLS] token to semantically refine the input text, while PTC-MTED focuses on the model by giving a prompt to the input text. Our results on both emotion datasets suggest that the prompting method is generally better than the method with [CLS] token. The [CLS] token contains more redundant semantic information than the prompt tokens. In contrast, the goal of the prompting method is more direct, that is, to predict the presence of each emotion label directly. Because prompt increases the overall length of the input, the computational overhead of the prompt-tuning increases slightly.

Although the experiments in this paper use BERT as the pre-trained language model, it is in fact very easy to be replaced with other pre-trained autoencoder language models such as RoBERTa [116].

4.5.3 About Prompt Engineering

The results of "-synonym" are slightly worse than the results of CLS-FT on the Ren-CECps dataset and slightly better than those of CLS-FT on the NLPCC2018 dataset. This indicates that a single emotional word in different datasets cannot completely cover its actual semantics of

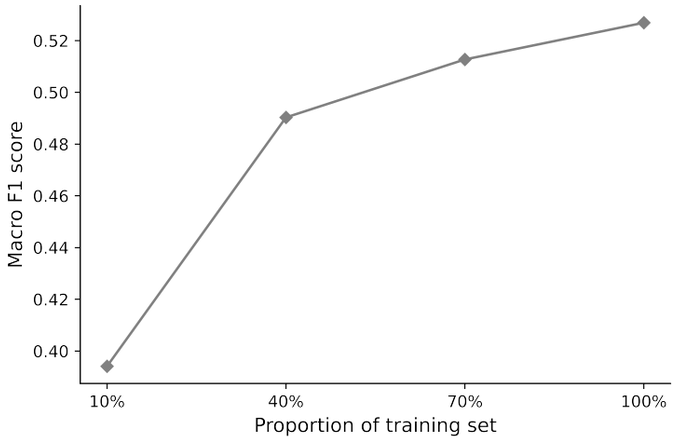

Figure 4.17 The Macro F1 score performance of the PTC-MTED model with different sizes of NLPCC2018 training data.

the corresponding emotion category in the semantic space, resulting in no significant advantage of using only the original emotion label as prompt over traditional tuning. When we extent the prompt with synonyms of the emotional words, the experimental results are improved. This indicates that synonyms can indeed complement the semantic representations of the emotion labels.

The performance of "-random" on both datasets is second only to that of PTC-MTED. This result indicates that the representation of two synonyms in the semantic space is only slightly different from each other. Randomly swapping synonyms within the emotion sets can slightly improve the generalization ability of the model, but the effect is limited.

We believe that fixing the position of the emotion in the prompt will make the model learn to output "no" for the minority emotion categories, which is a general problem caused by the biased training data. The "-shuffle" experiment proves that this does reduce the learning ability of the model. Shuffling the prompts order allows the model to understand which emotion is being predicted, which makes the learning of all emotion labels more purposeful.

The prompting method based on pre-trained language models has been proven to reduce the reliance on large amounts of supervised data [117]. We select the NLPCC2018 dataset with a small amount of data to test the learning efficiency of the PTC-MTED model under different proportions of training data. Figure 4.17 shows the Macro F1 score performance of the model

from random 10% data to full data. Using 40% of the training data (about 2.7k), the model achieves a result close to the CLS-FT method. This indicates that the PTC-MTED model has good learning ability with a small amount of data.

4.5.4 About Multi-label Loss Function

Two synonyms have different cluster centers in the semantic space. The experimental results of "-$L_{J(diff)}$" indicate that the consistent training for the prediction results of the pair of prompts can simultaneously shorten the distance between the data points and the two cluster centers in space. It suggests that the "different-consistency" training strategy is helpful for PTC-MTED.

The experiment results of "-$L_{J(same)}$" indicate that the "same-consistency" training strategy is very effective. Dropout is a regularization method that makes the model different at training and prediction phases. Increasing the drop rate will strengthen the constraint on the model, which will cause the instability of the model when predicting. The scale of the NLPCC2018 dataset is smaller than that of Ren-CECps, so the scores of "-$L_{J(same)}$" on the NLPCC2018 dataset drop more significantly compared to those of Ren-CECps. This observation suggests that smaller datasets are more susceptible to dropout instability. It also suggests that the "same-consistency" training strategy is generally effective to reduce the instability caused by dropout.

"-L_{MC}" performs much worse than PTC-MTED on both datasets. This illustrates that Su's loss function is effectively taking the connection between different labels into the multi-label classification loss calculation, and is more suitable for this task than the BCE loss.

The weight of the loss function for consistency training also affect the performance of the model:

$$L = L_{MC} + \alpha L_J \tag{4.16}$$

We set four reference values according to the R-Drop article [87]. As can be seen from Table 4.8, the model performs the best on the Ren-CECps dataset when α is 1/8, and the model performs the best on the NLPCC2018 dataset when α is 1/2. We apply the best-performing α to the comparison experiments and the ablation experiments.

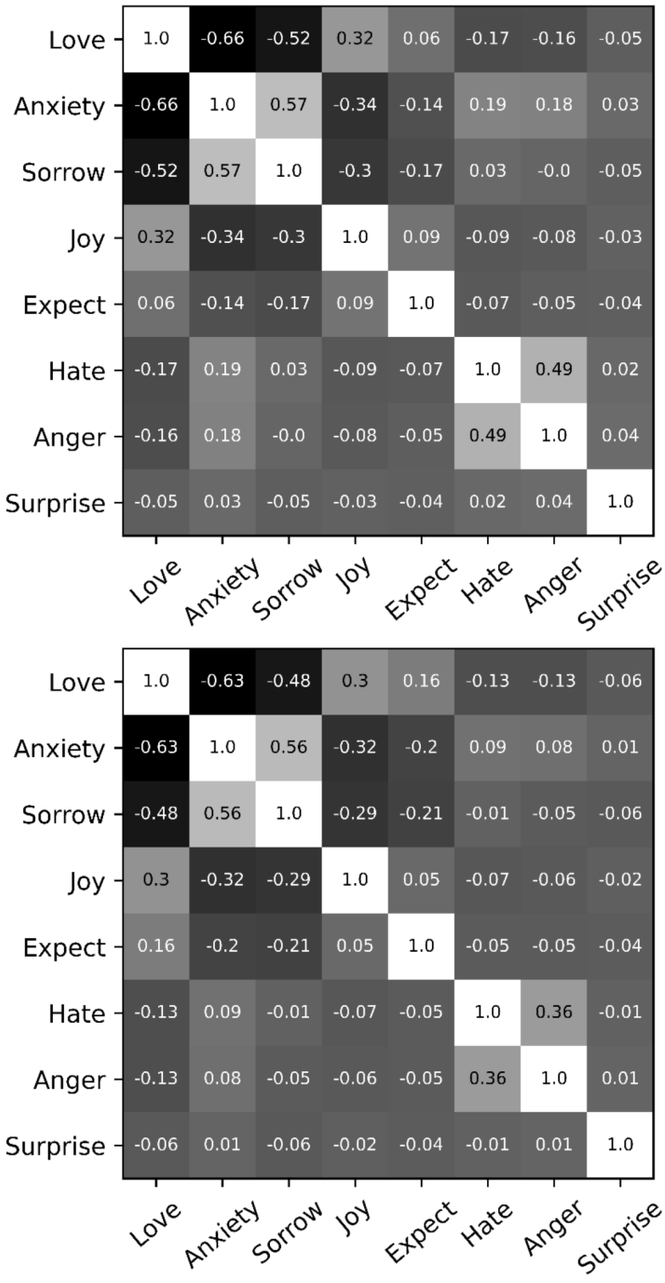

Figure 4.18 The emotional correlation coefficient matrix of the Ren-CECps test set. The top is the matrix of BERT-FT, and the bottom is the matrix of $D_{wn}K_{drn}T_{cw}$.

Table 4.8 The results of experiments with the weight of the consistency training on the two datasets.

Ren-CECps	Micro F1	Macro F1	AP	CE	RL
$\alpha = 1/8$	**0.6627**	**0.5432**	**0.8233**	**1.8851**	**0.0850**
$\alpha = 1/4$	0.6598	0.5368	0.8202	1.8928	0.0856
$\alpha = 1/2$	0.6602	0.5311	0.8229	1.8886	0.0852
$\alpha = 1$	0.6569	0.5327	0.8198	1.9070	0.0867
NLPCC2018	Micro F1	Macro F1	AP	CE	RL
$\alpha = 1/8$	0.6175	0.5193	0.8571	1.2725	0.1074
$\alpha = 1/4$	0.5963	0.5147	0.8526	1.2992	0.1138
$\alpha = 1/2$	**0.6401**	**0.5269**	**0.8755**	**1.2275**	**0.0959**
$\alpha = 1$	0.6251	0.5267	0.8666	1.2600	0.1038

4.5.5 About Neurosymbolic Approaches

Figure 4.18 shows the emotional correlation coefficient matrix of the Ren-CECps test set. Compared to ground truth, BERT-FT is essentially the same in terms of positive/negative correlations of emotions, indicating that the baseline model has learned the correlation between emotions. In terms of degree, the absolute values of correlation coefficients such as "love/anxiety" and "anxiety/sorrow" are higher than those of ground truth, indicating that the baseline model is a little bit overfitting to some extent. In general, the absolute values of some of the correlation coefficients become closer to those of the ground truth after the addition of the symbolic approaches. This indicates that the symbolic approaches can effectively alleviate the overfitting of the baseline model.

The emotional dictionary approach improves more on the NLPCC2018 dataset than on the Ren-CECps dataset, possibly because the emotion categories in the NLPCC2018 dataset all cor-

respond in DLUTE. “Expect” in the Ren-CECps dataset has no corresponding emotion category in DLUTE, which may lead to the correlation coefficients of this emotion with other emotions deviating from the ground truth. In the NLPCC2018 test set, there are 49.67% of the texts that contain the keywords in the dictionary with the emotion and also have this emotion in the ground truth (including the cases where no emotional word is matched and the ground truth is “neutral”). This proportion is 44.29% in the Ren-CECps test set, which is slightly lower than that in the NLPCC2018 test set. This may also be the reason why the emotional dictionary approach performs better on the NLPCC2018 dataset.

The syntactic structure can also affect the role of emotional words in emotion detection [118]. If there is a negation word in a text, it seems that the text may not have a certain emotion. In the sentiment analysis task, the rule of identifying negation words in the text can further increase accuracy since there are only two categories: positive and negative. When there are emotional words and a negation word in a text, the proportion of the ground truth with this emotion is 6.15% in the Ren-CECps test set, and the proportion without this emotion is 11.10%. These proportions are 8.75% and 11.83% in the NLPCC2018 test set. This shows that the rule of identifying negation words is not applicable in the emotion detection task.

The commonsense knowledge inference approach shows a significant improvement in the evaluation metrics on both datasets, which indicates that this approach is effective in helping the baseline model to detect emotions. Taking the descriptions of the subject in the event inferred by Comet as an example, “happy” is inferred 79 times, which is the most frequent description in the NLPCC2018 test set. Of the texts with “happy” description, 78.48% shows “happiness” emotion in the ground truth. In the entire dataset, 31.96% of the texts show “happiness” emotion. This suggests that the commonsense knowledge inference approach is effective in explicitly extracting emotion features. Similar results can be found in Ren-CECps. “Sad” is inferred 387 times in the Ren-CECps test set, in which 264 texts show “sorrow” emotion in the ground truth. Of the 264 texts with the “sorrow” emotion, the BERT with a “sad” description successfully detects 260 texts, while the BERT without a “sad” description only detects 255 texts. This indicates that the description of the subject inferred from commonsense knowledge is beneficial for emotion detection. Using multiple inference outcomes simultaneously seems to further improve the evaluation metrics.

The topic model clustering approach outperforms BERT-FT on both datasets. As can be

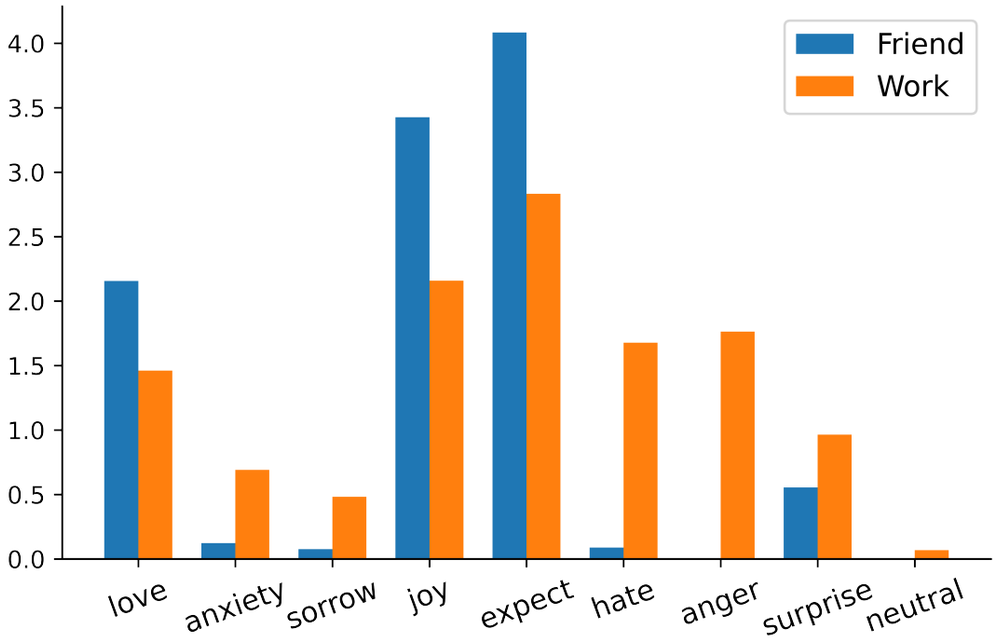

Figure 4.19 Histogram of proportionally normalized emotions in Ren-CECps texts of two topics. The vertical coordinate refers to the proportion of the emotion in the texts of the topic compared to that in the whole dataset.

seen from the results of T_c and T_w, the topic words contain more clustering information and therefore can better help the baseline model to detect emotions compared to only given the number of clustered categories. The texts clustered by the topic model have significantly different distributions in terms of emotions. As shown in Figure 4.19, texts related to "friends" tend to show positive emotions, while texts related to "work" show more negative emotions. Due to the imbalance in the number of emotions, we have normalized the proportion in Figure 4.19. This indicates that the topic model clustering approach can give emotional information to the detection model. Compared with the commonsense knowledge inference approach, the topic model clustering approach does not introduce new information, only emphasizes information to help detect emotions by feature engineering on the existing data. This may be the reason why the topic model clustering approach does not improve as much as the commonsense knowledge inference approach.

From the experimental results, a mixture of the above-mentioned symbolic approaches per-

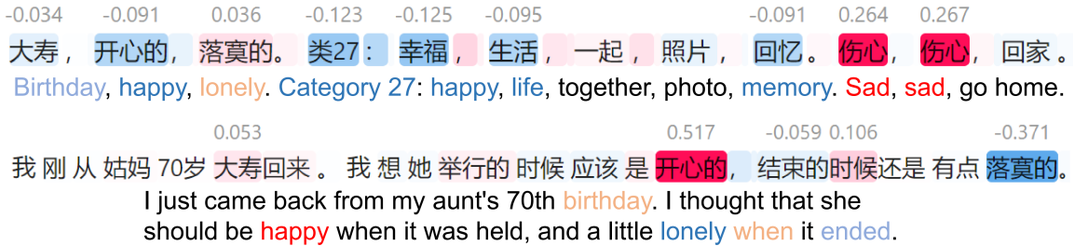

Figure 4.20 A case study of a visualization that explains the neurosymbolic approach by SHAP. The effect of the symbolic information in emotion detection is at the top. The effect of text information in emotion detection is at the bottom. Numbers with absolute values over 0.03 are shown above the words.

forms better on the Ren-CECps dataset. This may be since shorter texts are more likely to be affected by the added symbolic information when representing text semantics with the language model. From the statistics of the datasets chapter, the average length of the texts in the Ren-CECps dataset is smaller than that in the NLPCC2018 dataset. Therefore, the mixture of symbolic approaches is more beneficial for shorter texts.

W/o-text performs significantly better than all-LQ, indicating that the symbolic information can effectively help emotion detection. Comparing the results of w/o-text and BERT-FT, the text information is more advantageous than the symbolic information in emotion detection. SHapley Additive exPlanations (SHAP) is a tool to explain the output of the model in terms of the classic Shapley values from game theory [119]. As shown in Figure 4.20, SHAP scores each token of the input text to measure its contribution in emotion detection (the darker the color, the greater the contribution). “Birthday”, “happy” and “lonely” are the emotional words. “Happy”, “life”, “together”, “photo” and “memory” are the topic words. “Sad”, “sad” and “go home” are the outcomes of knowledge inference. The text in the case study implies “happiness” and “sadness” emotions. When the symbolic approaches are not used, the “happy” and “lonely” in the text contribute more than other words (“happiness” in red here). When using the symbolic approaches, most of the words contribute significantly to the emotion detection process (“sadness” in red here). This illustrates that the symbolic approaches assist the neural networks by explicit features in detecting emotions while improving the interpretability of the neural networks.

4.6 Summary

This chapter is devoted to investigating how contrastive strategies affect the accuracy of multi-label emotion recognition from text. Using unsupervised contrastive learning to adjust parameters before fine-tuning a large-scale pre-trained language model has no significant benefit for emotion recognition. In contrast, adding a supervised contrastive learning approach to fine-tune the large-scale pre-trained language model is beneficial for accuracy improvement. We experiment on two multi-label emotion classification datasets: Ren-CECps and NLPCC2018.

We also devote to designing a prompting method and two consistency training strategies for the multi-label textual emotion detection task. We experiment on two mentioned multi-label emotion classification datasets. The best-published result is achieved by a feature extraction approach of the language model [111]. Our model achieves over 6% and 3% improvement in Macro F1 scores on two datasets with new state-of-the-art results, respectively. This indicates a clear benefit of using the prompting method and the consistency training strategies in the multi-label textual emotion detection task.

We combine neural networks with symbolic approaches and train new emotion detection models to help the neural networks improve the fitting process. Specifically, we explore the effectiveness of the emotional dictionary, knowledge inference, and topic model clustering approaches on the PTM in this article. We experiment on two multi-label emotion classification datasets mentioned above. With the aid of the symbolic approaches, the proposed approaches get improvements on both datasets. Multiple experimental results demonstrate that the combination of symbolic approaches and neural networks is effective for multi-label textual emotion detection.

Chapter 5 Conclusion and Future work

5.1 Conclusion

Textual emotion recognition is an important part of the human-computer interaction field. Current methods of textual emotion recognition mainly use large-scale pre-trained models fine-tuning. However, these methods are not accurate enough in the semantic representation of sentences. Contrastive learning has been shown to optimize the representation of vectors in the feature space. Therefore, we introduce the contrastive strategies to the textual emotion recognition task. We propose two approaches: using self-supervised contrastive learning before fine-tuning the pre-trained model, and using contrastive training on the same inputs during fine-tuning. We experiment on two multi-label emotion classification datasets: Ren-CECps and NLPCC2018. The experimental results demonstrate that the latter contrastive approach effectively improves the accuracy of emotion recognition.

We introduce the contrastive strategies for multi-label textual emotion classification tasks. Based on the large-scale pre-trained model BERT, we propose two approaches: using self-supervised contrastive learning before fine-tuning the model, and using contrastive training on the same inputs during fine-tuning. We experiment with the effectiveness of the strategies on two multi-label emotion classification datasets: Ren-CECps and NLPCC2018. The experimental results demonstrate that using the contrastive strategy in the classification part is more effective in improving the accuracy of emotion recognition than using the contrastive strategy in the encoding part.

Textual emotion detection is playing an important role in the human-computer interaction domain. The mainstream methods of textual emotion detection are extracting semantic features and fine-tuning by language models. Due to the information redundancy in semantics, it is difficult for these methods to accurately detect all the emotions implied in the text. The prompting method has been shown to make the language models more purposeful in prediction by filling the cloze or prefix prompts defined. Therefore, we design a prompting approach for multi-label classification. To stabilize the output, we design two consistency training strategies. We experiment on two multi-label emotion classification datasets: Ren-CECps and NLPCC2018.

Our proposed prompt tuning with consistency training for multi-label textual emotion detection (PTC-MTED) model achieves Macro F1 scores of 0.5432 and 0.5269, respectively. The experimental results indicate that our proposed method has significant effectiveness in the multi-label textual emotion detection task.

In this paper, a prompt tuning with consistency training model is proposed for the multi-label textual emotion detection task. We change the emotion labels into a pair of prompts and fine-tune the model with two consistency training strategies. Our experimental results on the Ren-CECps dataset and the NLPCC2018 dataset demonstrate the effectiveness of our proposed model.

Neural networks are replacing symbolic approaches as better methods for textual emotion detection due to their powerful feature extraction capabilities. However, neural networks are prone to overfitting during training because of the small amount of emotion detection data. Based on experience or knowledge, symbolic approaches can fit a small amount of data by low-dimensional features and also outperform neural networks in terms of interpretability. We design three models combining symbolic approaches with neural networks for detecting all potential emotions from texts in this article. Due to the importance of emotional words in detection, we retrieve these words from texts by an emotional dictionary approach; we predict the reaction and describe the state of the subject to help detect emotions by a commonsense knowledge inference approach; we cluster texts by a topic model since texts with similar topics may have similar emotions. We supplement a large-scale pre-trained language model with symbolic approaches and experiment on two multi-label emotion classification datasets, which are Ren-CECps and NLPCC2018. The experimental results show that the symbolic approaches improve the fitting process, improve the interpretability and increase the accuracy of neural networks. This indicates that neurosymbolic methods are effective in the multi-label textual emotion detection task.

The neural network approach, while performing well in emotion detection tasks, is prone to overfitting on the dataset. In this article, we employ three symbolic approaches to assist a neural network model in detecting emotions. Our experimental results on the Ren-CECps dataset and the NLPCC2018 dataset demonstrate that the neurosymbolic approach can alleviate overfitting, increase detection accuracy, and improve interpretability. The commonsense knowledge inference approach introduces new information and has a better performance compared to the emotional dictionary approach and the topic model clustering approach. The code of the model

is available at https://github.com/youngzhou97qz/Neurosymbolic-multi-label-textual-emotion-detection and https://github.com/youngzhou97qz/Prompt-consistency-for-multi-label-emotion-detection.

5.2 Future work

In the future, we will investigate the relationship between word embeddings and $[cls]$ token to minimize the loss of emotional information contained in sentences during training. The emotion labels are treated as 0 or 1 in this paper, which may cause the information loss of the emotional intensity. Therefore, we will focus on handling the continuous labels through prompt tuning. In addition, our future work will focus on other kinds of symbolic approaches and effective methods for merging neural features with symbolic features.

Acknowledgment

I am honored to complete my doctor's degree at Tokushima University. I am very grateful that in the past three years, Tokushima University has provided me with a strong academic atmosphere, rich campus life, and brought me many friendships from all over the world. The platform provided by Tokushima University has broadened my international horizons. I will never forget the study life here.

I am honored to be able to carry out my academic work in the A1 group. In particular, Professor Kenji Terada and Professor Fuji Ren, always give me a lot of help in professional knowledge and in life. I also want to thank another teacher in the A1 group, Xin Kang, who gives me a lot of great research opinions at the group meetings. In addition, I would like to thank Professor Fuketa and Professor Shishibori, participating in the publication of my thesis giving valuable suggestions. I would also like to thank the classmates who fought with me in the A1 group and brought me happiness during my research process. I am very happy to be able to meet these teachers and classmates.

I am honored to have family members who have always supported me. My wife, Lipei Liu, studied at Tokushima University with me. Without her encouragement, I will not finish my graduation thesis so smoothly. I also want to thank my parents for continuing to give me the care and love, so that I don't feel lonely when I encountering difficulties.

Finally, I would like to express my heartfelt thanks to all those who have helped me.

References

[1] Rosalind W Picard. *Affective computing*. MIT press, 2000.

[2] Rosalind W Picard. Toward machines with emotional intelligence. 2008.

[3] Amy Reeves. Emotional intelligence: recognizing and regulating emotions. *Aaohn Journal*, 53(4):172–176, 2005.

[4] Alan M Turing. Computing machinery and intelligence. parsing the turing test, 2009.

[5] Jens Edlund, Joakim Gustafson, Mattias Heldner, and Anna Hjalmarsson. Towards human-like spoken dialogue systems. *Speech communication*, 50(8-9):630–645, 2008.

[6] Fuji Ren and Yanwei Bao. A review on human-computer interaction and intelligent robots. *International Journal of Information Technology & Decision Making*, 19(01):5–47, 2020.

[7] Tsung-Hsien Wen, David Vandyke, Nikola Mrksic, Milica Gasic, Lina M Rojas-Barahona, Pei-Hao Su, Stefan Ultes, and Steve Young. A network-based end-to-end trainable task-oriented dialogue system. *arXiv preprint arXiv:1604.04562*, 2016.

[8] Bingjie Liu and S Shyam Sundar. Should machines express sympathy and empathy? experiments with a health advice chatbot. *Cyberpsychology, Behavior, and Social Networking*, 21(10):625–636, 2018.

[9] Grégoire Mesnil, Yann Dauphin, Kaisheng Yao, Yoshua Bengio, Li Deng, Dilek Hakkani-Tur, Xiaodong He, Larry Heck, Gokhan Tur, Dong Yu, et al. Using recurrent neural networks for slot filling in spoken language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):530–539, 2014.

[10] Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. A survey on dialogue systems: Recent advances and new frontiers. *Acm Sigkdd Explorations Newsletter*, 19(2):25–35, 2017.

[11] Marina Krakovsky. Artificial (emotional) intelligence. *Communications of the ACM*, 61(4):18–19, 2018.

[12] Xutong Li and Rongheng Lin. Speech emotion recognition for power customer service. In *2021 7th International Conference on Computer and Communications (ICCC)*, pages 514–518. IEEE, 2021.

[13] Danang Satrio, Sony Heru Priyanto, and Albert KNA Nugraha. Viral marketing for cul-

tural product: The role of emotion and cultural awareness to influence purchasing intention. *Montenegrin Journal of Economics*, 16(2):77–91, 2020.

[14] Kangning Yang, Chaofan Wang, Yue Gu, Zhanna Sarsenbayeva, Benjamin Tag, Tilman Dingler, Greg Wadley, and Jorge Goncalves. Behavioral and physiological signals-based deep multimodal approach for mobile emotion recognition. *IEEE Transactions on Affective Computing*, 2021.

[15] Lei Zhang, Shuai Wang, and Bing Liu. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253, 2018.

[16] Jun Li, Yuemei Xu, Hao Xiong, and Yan Wang. Chinese text emotion classification based on emotion dictionary. In *2010 IEEE 2nd Symposium on Web Society*, pages 170–174. IEEE, 2010.

[17] Navonil Majumder, Soujanya Poria, Alexander Gelbukh, and Erik Cambria. Deep learning-based document modeling for personality detection from text. *IEEE Intelligent Systems*, 32(2):74–79, 2017.

[18] Bing Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.

[19] Changqin Quan and Fuji Ren. An exploration of features for recognizing word emotion. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 922–930, 2010.

[20] Mohiuddin Qudar and Vijay Mago. A survey on language models. 2020.

[21] Goutam Majumder, Partha Pakray, Alexander Gelbukh, and David Pinto. Semantic textual similarity methods, tools, and applications: A survey. *Computación y Sistemas*, 20(4):647–665, 2016.

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

[23] Xipeng Qiu, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897, 2020.

[24] Nourah Alswaidan and Mohamed El Bachir Menai. A survey of state-of-the-art approaches for emotion recognition in text. *Knowledge and Information Systems*, 62(8):2937–2987,

2020.

[25] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[26] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[27] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[28] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

[29] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*, 2015.

[30] Jianhua Tao and Tieniu Tan. Affective computing: A review. In *International Conference on Affective computing and intelligent interaction*, pages 981–995. Springer, 2005.

[31] Fuji Ren. Affective information processing and recognizing human emotion. *Electronic notes in theoretical computer science*, 225:39–50, 2009.

[32] Xiao Sun, Xiaoqi Peng, and Shuai Ding. Emotional human-machine conversation generation based on long short-term memory. *Cognitive Computation*, 10(3):389–397, 2018.

[33] Fuji Ren and Kazuyuki Matsumoto. Semi-automatic creation of youth slang corpus and its application to affective computing. *IEEE Transactions on Affective Computing*, 7(2):176–189, 2015.

[34] Yaoqin Zhang and Minlie Huang. Overview of the ntcir-14 short text generation subtask: emotion generation challenge. In *Proceedings of the 14th NTCIR Conference*, 2019.

[35] Hao Zhou, Minlie Huang, Tianyang Zhang, Xiaoyan Zhu, and Bing Liu. Emotional chatting machine: Emotional conversation generation with internal and external memory. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[36] Yangyang Zhou, Zheng Liu, Xin Kang, Yunong Wu, and Fuji Ren. Tua1 at the ntcir-14 stc-3 task. In *Proceedings of the 14th NTCIR Conference on Evaluation of Information Access Technologies*, 2019.

[37] Ronald J Williams and David Zipser. A learning algorithm for continually running fully

recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

[38] Anil Bandhakavi, Nirmalie Wiratunga, Deepak Padmanabhan, and Stewart Massie. Lexicon based feature extraction for emotion text classification. *Pattern recognition letters*, 93:133–142, 2017.

[39] Shuhua Monica Liu and Jiun-Hung Chen. A multi-label classification based approach for sentiment classification. *Expert Systems with Applications*, 42(3):1083–1093, 2015.

[40] Dhivya Chandrasekaran and Vijay Mago. Evolution of semantic similarity—a survey. *ACM Computing Surveys (CSUR)*, 54(2):1–37, 2021.

[41] Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. On the sentence embeddings from pre-trained language models. *arXiv preprint arXiv:2011.05864*, 2020.

[42] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020.

[43] Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*, 2021.

[44] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*, 2021.

[45] Tong Wang and Graeme Hirst. Near-synonym lexical choice in latent semantic space. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1182–1190, 2010.

[46] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.

[47] Yassine Ouali, Céline Hudelot, and Myriam Tami. Semi-supervised semantic segmentation with cross-consistency training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12674–12684, 2020.

[48] Jiawen Deng and Fuji Ren. A survey of textual emotion recognition and its challenges. *IEEE Transactions on Affective Computing*, 2021.

[49] Zhang Wei, Chunyu Wu, and Lei Chen. Symbolism and connectionism of artificial in-

telligence. In *IEEE APCCAS 2000. 2000 IEEE Asia-Pacific Conference on Circuits and Systems. Electronic Communication Systems.(Cat. No. 00EX394)*, pages 364–366. IEEE, 2000.

[50] David Landy, Colin Allen, and Carlos Zednik. A perceptual account of symbolic reasoning. *Frontiers in psychology*, 5:275, 2014.

[51] Shu-Hsien Liao. Expert system methodologies and applications—a decade review from 1995 to 2004. *Expert systems with applications*, 28(1):93–103, 2005.

[52] Marvin Minsky. *The emotion machine: Commonsense thinking, artificial intelligence, and the future of the human mind*. Simon and Schuster, 2007.

[53] Lamyaa Sadouk, Taoufiq Gadi, and El Hassan Essoufi. Robust loss function for deep learning regression with outliers. In *Embedded Systems and Artificial Intelligence*, pages 359–368. Springer, 2020.

[54] Xiaowei Huang, Daniel Kroening, Wenjie Ruan, James Sharp, Youcheng Sun, Emese Thamo, Min Wu, and Xinping Yi. A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability. *Computer Science Review*, 37:100270, 2020.

[55] Artur d'Avila Garcez, Marco Gori, Luis C Lamb, Luciano Serafini, Michael Spranger, and Son N Tran. Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning. *arXiv preprint arXiv:1905.06088*, 2019.

[56] Hongxu Chen, Hongzhi Yin, Xue Li, Meng Wang, Weitong Chen, and Tong Chen. People opinion topic model: opinion based user clustering in social networks. In *Proceedings of the 26th International Conference on World Wide Web Companion*, pages 1353–1359, 2017.

[57] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. *Advances in Neural Information Processing Systems*, 32, 2019.

[58] Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. Bridging the gap between training and inference for neural machine translation. *arXiv preprint arXiv:1906.02448*, 2019.

[59] Peng Si Ow and Thomas E Morton. Filtered beam search in scheduling. *The International Journal Of Production Research*, 26(1):35–62, 1988.

[60] Stephen E Robertson, Steve Walker, Susan Jones, Micheline M Hancock-Beaulieu, Mike Gatford, et al. Okapi at trec-3. *Nist Special Publication Sp*, 109:109, 1995.

[61] Long Mai and Bac Le. Joint sentence and aspect-level sentiment analysis of product comments. *Annals of Operations research*, 300(2):493–513, 2021.

[62] Nicu Sebe, Ira Cohen, Theo Gevers, and Thomas S Huang. Multimodal approaches for emotion recognition: a survey. In *Internet Imaging VI*, volume 5670, pages 56–67. SPIE, 2005.

[63] Sophia Yat Mei Lee, Ying Chen, and Chu-Ren Huang. A text-driven rule-based system for emotion cause detection. In *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, pages 45–53, 2010.

[64] Tanvi Hardeniya and Dilipkumar A Borikar. Dictionary based approach to sentiment analysis-a review. *International Journal of Advanced Engineering, Management and Science*, 2(5):239438, 2016.

[65] Vo Ngoc Phu, Vo Thi Ngoc Chau, Vo Thi Ngoc Tran, and Nguyen Duy Dat. A vietnamese adjective emotion dictionary based on exploitation of vietnamese language characteristics. *Artificial Intelligence Review*, 50(1):93–159, 2018.

[66] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[67] Gilbert Badaro, Hussein Jundi, Hazem Hajj, and Wassim El-Hajj. Emowordnet: Automatic expansion of emotion lexicon using english wordnet. In *Proceedings of the seventh joint conference on lexical and computational semantics*, pages 86–93, 2018.

[68] Fuji Ren and Jiawen Deng. Background knowledge based multi-stream neural network for text classification. *Applied Sciences*, 8(12):2472, 2018.

[69] Yankai Lin, Xu Han, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. Knowledge representation learning: A quantitative review. *arXiv preprint arXiv:1812.10901*, 2018.

[70] Yong-Soo Seol, Dong-Joo Kim, and Han-Woo Kim. Emotion recognition from text using knowledge-based ann. In *ITC-CSCC: International Technical Conference on Circuits Systems, Computers and Communications*, pages 1569–1572, 2008.

[71] Jorge Martinez-Gil. Automated knowledge base management: A survey. *Computer Science Review*, 18:1–9, 2015.

[72] Deepanway Ghosal, Navonil Majumder, Alexander Gelbukh, Rada Mihalcea, and Soujanya

Poria. Cosmic: Commonsense knowledge for emotion identification in conversations. *arXiv preprint arXiv:2010.02795*, 2020.

[73] Samina Khalid, Tehmina Khalil, and Shamila Nasreen. A survey of feature selection and feature extraction techniques in machine learning. In *2014 science and information conference*, pages 372–378. IEEE, 2014.

[74] Vajrapu Anusha and Banda Sandhya. A learning based emotion classifier with semantic text processing. In *Advances in intelligent informatics*, pages 371–382. Springer, 2015.

[75] R Sandhiya, AM Boopika, M Akshatha, SV Swetha, and NM Hariharan. A review of topic modeling and its application. *Handbook of Intelligent Computing and Optimization for Sustainable Development*, pages 305–322, 2022.

[76] Chenghua Lin and Yulan He. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 375–384, 2009.

[77] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[78] Navonil Majumder, Soujanya Poria, Devamanyu Hazarika, Rada Mihalcea, Alexander Gelbukh, and Erik Cambria. Dialoguernn: An attentive rnn for emotion detection in conversations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6818–6825, 2019.

[79] Xu Han, Zhengyan Zhang, Ning Ding, Yuxian Gu, Xiao Liu, Yuqi Huo, Jiezhong Qiu, Liang Zhang, Wentao Han, Minlie Huang, et al. Pre-trained models: Past, present and future. *AI Open*, 2021.

[80] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[81] Taewoon Kim and Piek Vossen. Emoberta: Speaker-aware emotion recognition in conversation with roberta. *arXiv preprint arXiv:2108.12009*, 2021.

[82] Shadi Shaheen, Wassim El-Hajj, Hazem Hajj, and Shady Elbassuoni. Emotion recognition from text based on automatically generated rules. In *2014 IEEE International Conference on Data Mining Workshop*, pages 383–392. IEEE, 2014.

[83] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2021.

[84] Connor Shorten, Taghi M Khoshgoftaar, and Borko Furht. Text data augmentation for deep learning. *Journal of big Data*, 8(1):1–34, 2021.

[85] Xiaobo Liang, Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, and Tie-Yan Liu. R-drop: Regularized dropout for neural networks. *arXiv preprint arXiv:2106.14448*, 2021.

[86] Vikas Verma, Kenji Kawaguchi, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. *arXiv preprint arXiv:1903.03825*, 2019.

[87] Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, Tie-Yan Liu, et al. R-drop: regularized dropout for neural networks. *Advances in Neural Information Processing Systems*, 34, 2021.

[88] Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*, 2019.

[89] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

[90] Zhengbao Jiang, Frank F Xu, Jun Araki, and Graham Neubig. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020.

[91] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[92] Jinming Zhao, Ruichen Li, Qin Jin, Xinchao Wang, and Haizhou Li. Memobert: Pre-training model with prompt-based learning for multimodal emotion recognition. *arXiv preprint arXiv:2111.00865*, 2021.

[93] Timo Schick and Hinrich Schütze. Exploiting cloze questions for few shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*, 2020.

[94] Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*, 2020.

[95] Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. Ptr: Prompt tuning

with rules for text classification. *arXiv preprint arXiv:2105.11259*, 2021.

[96] Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. Template-based named entity recognition using bart. *arXiv preprint arXiv:2106.01760*, 2021.

[97] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pages 10524–10533. PMLR, 2020.

[98] Marzieh Fadaee, Arianna Bisazza, and Christof Monz. Data augmentation for low-resource neural machine translation. *arXiv preprint arXiv:1705.00440*, 2017.

[99] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[100] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[101] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[102] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[103] Harold Jeffreys. *Theory of probability*. Oxford: Clarendon Press, 2nd edition, 1948.

[104] Xudong Deng. cnsenti: An open-source python library for chinese text sentiment analysis. `https://github.com/hiDaDeng/cnsenti`, 2019.

[105] Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. Comet: Commonsense transformers for automatic knowledge graph construction. *arXiv preprint arXiv:1906.05317*, 2019.

[106] Dimo Angelov. Top2vec: Distributed representations of topics. *arXiv preprint arXiv:2008.09470*, 2020.

[107] Yutaka Matsuo and Mitsuru Ishizuka. Keyword extraction from a single document using

word co-occurrence statistical information. *International Journal on Artificial Intelligence Tools*, 13(01):157–169, 2004.

[108] Oscar Luaces, Jorge Díez, José Barranquero, Juan José del Coz, and Antonio Bahamonde. Binary relevance efficacy for multilabel classification. *Progress in Artificial Intelligence*, 1(4):303–313, 2012.

[109] Ji Li and Fuji Ren. Creating a chinese emotion lexicon based on corpus ren-cecps. In *2011 IEEE International Conference on Cloud Computing and Intelligence Systems*, pages 80–84. IEEE, 2011.

[110] Zhongqing Wang, Shoushan Li, Fan Wu, Qingying Sun, and Guodong Zhou. Overview of nlpcc 2018 shared task 1: Emotion detection in code-switching text. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 429–433. Springer, 2018.

[111] Jiawen Deng and Fuji Ren. Multi-label emotion detection via emotion-specified feature extraction and emotion correlation learning. *IEEE Transactions on Affective Computing*, 2020.

[112] Min-Ling Zhang and Zhi-Hua Zhou. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE transactions on Knowledge and Data Engineering*, 18(10):1338–1351, 2006.

[113] Thomas Wolf, Julien Chaumond, Lysandre Debut, Victor Sanh, Clement Delangue, Anthony Moi, Pierric Cistac, Morgan Funtowicz, Joe Davison, Sam Shleifer, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, 2020.

[114] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[115] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[116] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert

pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[117] Robert L Logan IV, Ivana Balažević, Eric Wallace, Fabio Petroni, Sameer Singh, and Sebastian Riedel. Cutting down on prompts and parameters: Simple few-shot learning with language models. *arXiv preprint arXiv:2106.13353*, 2021.

[118] Keliang Jia and Zhinuo Li. Chinese micro-blog sentiment classification based on emotion dictionary and semantic rules. In *2020 International Conference on Computer Information and Big Data Applications (CIBDA)*, pages 309–312. IEEE, 2020.

[119] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.