university of groningen

campus fryslân

Parameter-Efficient Fine-Tuning for Sarcasm Detection in Speech Using the Self-Supervised Pre-Trained Model WavLM

Weixi Lai

university of groningen

campus fryslân

University of Groningen - Campus Fryslân

Parameter Efficient Fine-Tuning for Sarcasm Detection in Speech Using the Self-Supervised Pre-Trained Model WavLM

Master's Thesis

To fulfill the requirements for the degree of
Master of Science in Voice Technology
at University of Groningen under the supervision of
X. Gao (Voice Technology, University of Groningen)
with the second reader being
Assoc. Prof. M. Coler (Voice Technology, University of Groningen)

Weixi Lai (S5500079)

July 12, 2024

Acknowledgements

Firstly, I want to express my heartfelt gratitude to my supervisor, Xiyuan Gao, for her unwavering professional support and understanding throughout this journey. Every meeting with her has been a delightful experience; she always listens with the most beautiful smile and the keenest attention, offering feedback in the gentlest voice and with the highest level of expertise. Her email responses have always been prompt and timely. In my 25 years, I have never met a teacher as professional, patient, and kind as she is. I sincerely wish her all the best in her work and personal life. I am also grateful for her expert advice on my thesis, even though I couldn't implement all of it due to my own time constraints.

I would like to thank our department head, Matt Coler, as well as all my professors during my master's program: Phat, Shekhar, Vass, and Joshua. They opened the door to the world of artificial intelligence and speech technology for me. Although I am uncertain if I will pursue a career in this field, their teachings have significantly influenced my career development and life trajectory.

Special thanks to my student advisor, Hieke. Thank you for understanding my illness, listening to my concerns, and providing invaluable advice. You witnessed the few tears I shed in the Netherlands and comforted my fears.

I am grateful to my family and friends for their care and understanding during my illness.

I acknowledge the Center for Information Technology at the University of Groningen for their technical support and access to the Hábrok high-performance computing cluster. Thanks to GP for their help and medical treatment, which allowed me to pursue my studies relatively stably. I also appreciate the Netherlands, especially its weather, for making me stronger and helping me realize my desire to live in a warmer place in the future.

I also want to thank myself. In those challenging moments, I found solace in love and positivity, as well as in facing my own weaknesses—procrastination, self-doubt, and forgetfulness—that motivated me to move forward. Despite disappointments in life and in myself, I always remind myself that I'm on my side, no matter what happens. As long as I'm alive, there's hope. Life doesn't always balance good and evil perfectly, but I hope to live long enough to see how it unfolds. I embrace imperfect family members, flawed systems, surprises, ups and downs, and my own decisions. I accept this world with its complexities, knowing it may not notice someone as insignificant as me.

Lastly, I extend my heartfelt thanks to everyone who has contributed, no matter how small, to shaping my academic journey and helping me successfully complete this thesis.

Abstract

As an integral part of human language and culture, sarcasm has naturally attracted great interest from researchers across various fields, including artificial intelligence. While much attention has been devoted to sarcasm detection in textual data, the realm of speech has remained relatively unexplored. Leveraging recent advancements in self-supervised learning (SSL) in speech proessing, I aim to explore Low-rank Adaptation (LoRA), one of the parameter-efficient fine-tuning (PEFT) technique with the self-supervised pre-trained model WavLM. To my knowledge, this study represents a pioneering effort in utilizing PEFT and WavLM for this specific task. By leveraging recent advancements in WavLM, the effectiveness of LoRA is rigorously evaluated through extensive analysis and comparison with the traditional fine-tuning method and other PEFT approaches. The results demonstrate LoRA's superiority in F1 score, recall, and precision metrics for sarcasm speech detection, while also highlighting its capability to significantly reduce parameter requirements. These findings provide valuable insights into the potential and challenges of employing LoRA with SSL in sarcasm speech detection, offering critical guidance for future research in advancing natural language understanding and enhancing human-computer interaction.

Contents

1 Introduction

Sarcasm is a complex linguistic phenomenon where speakers intentionally use expressions opposite to their literal meanings to convey mockery, sarcasm, or disdain (Potamias, Siolas, & Stafylopatis, 2020). Sarcasm expressions often carry an underlying negative sentiment while maintaining a positive surface sentiment.

Characterized by its interplay of linguistic cues and contextual understanding, sarcasm presents a challenge in the realm of natural language understanding, particularly in speech. Unlike its textual counterpart, which relies on linguistic cues and contextual understanding (Aboobaker & Ilavarasan, 2020), sarcasm in speech involves prosodic phenomena, making its detection very complex. For example, a sentence like "The weather in the Netherlands is really great" may be sarcastic or not, depending not only on the context (the weather in the Netherlands is typically overcast) but also on the way in which the sentence is produced. That is, if the word "really" was emphasized in a particular way, a sarcastic meaning could be conveyed.

As an integral part of human language and culture, sarcasm has naturally garnered great interest from researchers from varied fields of study, including artificial intelligence. Though automatic sarcasm detection has become an increasingly popular topic in the past decade, previous work has mainly focused on sarcasm detection in text (Joshi, Bhattacharyya, & Carman, 2017), utilizing indicators like special characters, emoticons (Carvalho, Sarmento, Silva, & De Oliveira, 2009), and distinctive organizational patterns identifiable through both syntactic and semantic features (Suhaimin, Hijazi, Alfred, & Coenen, 2017). However, sarcasm detection in speech has not received as much scholarly attention comparatively, despite the fair amount of work done on automatically detecting emotion in human speech (El Ayadi, Kamel, & Karray, 2011), which is also a prosody-weighted classification task. Tepperman, Traum, and Narayanan (2006) stands out as one of the pioneering publications in sarcasm detection using speech.

As voice technology becomes increasingly prevalent in our daily activities and development of human-computer interaction, sarcasm detection in speech drives more and more attention in recent years. Traditional approaches to sarcasm detection in speech have relied largely on the identification of prosodic cues such as intonation and stress (Bryant, 2010; Woodland & Voyer, 2011) and machine learning techniques such as tree-based methods and Support Vector Machines (SVM) (Castro et al., 2019; Rakov & Rosenberg, 2013; Tepperman et al., 2006). Then a few attempts such as relying on Deep Convolutional Neural Networks (DCNNs) based transfer learning (Gao, Nayak, & Coler, 2022) prove that Neural Networks (NNs) based transfer learning can also be a very effective tool for sarcasm speech detection, however, when it comes to the choice of pre-trained models in transfer learning domain, the self-supervised learning (SSL) has not been fully explored in sarcasm speech detection.

In the field of speech processing, self-supervised learning (SSL) pre-trained models like Wave2vec 2.0 (Baevski, Zhou, Mohamed, & Auli, 2020) and HuBERT (Hsu et al., 2021) have gained popularity. Vastly diverse unlabeled data leveraged in SSL models can lead to better generalization when learning higher-level speech representations compared to traditional supervised learning models. Among various SSL pre-trained speech models, the state-of-the-art model WavLM stands out as a versatile solution applicable to full stack downstream speech tasks (S. Chen et al., 2022). Extending the HuBERT framework to masked speech prediction and denoising modeling, wavLM enables the pre-trained models to perform well on automatic speech recognition, including speaker verification, speech recognition, and speech classification, etc. Notably, its effectiveness extends to speech emotion recognition (Atmaja & Sasou, 2022), highlighting its proficiency in prosody-weighted classification tasks. This capability suggests that WavLM also holds significant potential for recognizing sarcasm in speech.

Recently, Pre-trained Language Models (PLMs) and pre-training-fine-tuning approach have become the paradigm for Natural Language Processing (NLP) tasks (Ding et al., 2023). Though emerging research evidence consistently demonstrates that larger models often yield better performance, fine-tuning the entire large-scale PLMs can be computationally prohibitive and inefficient due to the vast number of parameters involved. For instance, the WavLM Base and WavLM Base+ have 12 Transformer encoder layers, 768-dimensional hidden states, and 8 attention heads, resulting in 94.70M parameters. To address these limitations, parameter-efficient fine-tuning (PEFT) techniques have been proposed recently as a prevalent methodology to adapt PLMs to downstream tasks, providing strong performances on many popular NLP benchmarks without modifying the pretrained architecture (Houlsby et al., 2019). PEFT has shown effectiveness for various speech tasks including ASR (Thomas, Kessler, & Karout, 2022) and speech emotional recognition (Atmaja & Sasou, 2022). The dominant PEFT methods includes adapter (Houlsby et al., 2019), Prefix-Tuning (Li & Liang, 2021), Prompt Tuning (Lester, Al-Rfou, & Constant, 2021) and LoRA (Hu et al., 2021). Compared to other PEFT methods, LoRA overcomes the inference latency brought by the adapter, as well as the challenges of prefix-tuning which reduces the model's usable sequence length, successfully matching the fine-tuning baselines without compromising efficiency and model quality. In addition, when it comes to prosody-weighted tasks, LoRA has already showed better performance on speech emotion recognitions compared to other PEFT methods like adpter and Prompt Tuning (Feng & Narayanan, 2023).

Since the utilization of SSL pre-trained models and PEFT methods has not been explored in the research of sarcasm detection in speech, this thesis aims to fill this research gap. Specifically, leveraging the advantages of WavLM and LoRA in prosody-weighted classification tasks, I will use WavLM as the base model and combine it with loRA as the PEFT method. I will investigate how to more accurately identify sarcasm in speech and compare its performance with traditional fine-tuning methods, offering insights that could propel further innovation.

In light of the preceding discussion, the research question at the core of this study can be formulated as follows:

> **Can the parmater-efficient fine-tuning method LoRA adapts the Self-supervised pre-trained speech model WavLM for speech sarcasm detection, and how effective is it?**

My hypothesis are:

- **hypothesis 1** As fine-tuning generally improves the performance of pre-trained models on specific tasks,the WavLM model with LoRA-based PEFT method is expected to outperform directly extracting embedding features from the pre-trained WavLM for sarcasm speech detection.

- **hypothesis 2** Compared to the traditional fine-tuning method of freezing the pre-trained encoder and only fine-tuning the downstream model, the application of LoRA-based PEFT on the WavLM model is expected to improve performance in sarcasm detection.

- **hypothesis 3** Compared with other PEFT methods applied to the WavLM model, such as adapter tuning and embedding prompt tuning, the LoRA technique is expected to demonstrate superior performance in terms of F1 score, recall, and precision metrics in sarcasm speech detection, providing a basis for selecting the most effective PEFT approach.

This forms the basis for the research questions and hypotheses, laying the groundwork for new explorations in this field.

Now that a brief motivation for this research has been presented, the structure of the thesis is the following: Section 2 provides an extensive literature review that frames the research question and hypothesis in the state-of-the-art. In section 3, the methodology is covered and the underlying models used are explained. Then, section 4 describes the experimental setup developed to answer the research questions and validate the hypothesis. Section 5 describes the results obtained. In section 6, I discuss the previously-mentioned results in detail and some limitations with recommended future work. Lastly, section 7 summarizes the thesis and presents the conclusions drawn.

2 Literature Review

This section is dedicated to providing a comprehensive review of the existing research on sarcasm detection, with a specific focus on the development of speech sarcasm detection. The review outlines the trajectory of sarcasm detection research, transitioning from text-based methods to new explorations in speech. Despite progress, the application of SSL pre-trained models in speech sarcasm detection remains less explored. Meanwhile, the recent combination of PEFT Methods and large-scale pre-trained models in the deep learning field has emerged as a new paradigm, reducing computational resources while maintaining accuracy in downstream tasks. Various SSL models and different PEFT methods have been explored in speech emotion recognition tasks, which, like sarcasm detection, are also prosody-based. Among them, research indicates that WavLM and LoRA exhibit advantages in this domain. The literature review is organized as follows: section 2.1 provides an overview of sarcasm detection, covering textual detection in subsection 2.1.1 and speech-based detection in subsection 2.1.2. It elucidates the challenges inherent in sarcasm detection, particularly in speech, and highlights the evolving research landscape in this domain. Section 2.2 explores advancements in SSL for speech, and examine notable SSL pre-trained speech models such as Wav2vec2.0 and HuBERT in subsection 2.2.1 and WavLM in subsection 2.2.2, elucidating the latter's better performance when it comes to prosody-weighted classification tasks like speech emotion recognition. Section 2.3 delves into PEFT methods, exploring their significance for large-scale pre-trained models and various downstream tasks. Subsequently, the chapter delves into the adapter-based approach in Section 2.3.1, followed by an examination of the Prefix-tuning approach in subsection 2.3.2. Furthermore, Subsection 2.3.3 explores the prompt tuning technique, which can be seen as a simplification of prefix-tuning. Finally, Section 2.3.4 presents the LoRA method, which addresses the challenges posed by the adapter and prefix-tuning approaches.

By structuring the literature review in this way, I provide a comprehensive analysis of existing research and set the foundation for my research questions and hypothesis.

2.1 Sarcasm Detection

Sarcasm detection is a growing field i in the field of nature learning processing, playing a crucial role in sentiment analysis and facilitating better understanding for effective communication between machines and humans. For example, identifying sarcastic behavior on online social networks such as Facebook, Twitter, Instagram, and surveys has become essential, as it significantly impacts social and personal relationships (Ashwitha et al., 2021).

The remarkable aspect of sarcasm is its ability to create a substantial divergence between its literal and intended meanings (Ashwitha et al., 2021). This metaphorical essence of sarcasm presents a considerable challenge in sentiment analysis and poses obstacles for sarcasm detection within the realm of natural language processing. While sarcasm detection has been extensively explored in textual analysis, efforts in the domain of speech remain relatively limited.

2.1.1 Sarcasm Detection in Text

Investigations into sarcasm primarily focus on identifying sarcasm within textual content, framing sarcasm recognition as a challenge in text classification. Sarcasm detection involves predicting sarcasm in text, which is a crucial step in sentiment analysis given the prevalence and challenges of sarcasm in emotionally charged text.

In the text domain, prior to 2020, most papers were focused on utilizing Twitter datasets. However, recent studies have shown an increasing interest in multiple data sources, including Reddit, news articles, books, and even YouTube. A trend that is emerging is the utilization of diverse data sources for constructing sarcasm datasets (Băroiu & Trăușan-Matu, 2022). These datasets are typically annotated using two primary strategies: manual annotation and distant supervision via labeled hashtags. Other research endeavors to leverage contextual information to gather shared knowledge between speakers and audiences. Various contextual features have been explored, such as the background and behavior of speakers on online platforms, embeddings of expressed sentiment and speaker personality traits, learning user-specific representations, user-community features, as well as stylistic and discourse features.

The main methods used for text-based sarcasm detection are rule-based, lexicon-based, machine learning-based, and deep learning-based approaches (Aboobaker & Ilavarasan, 2020). The rule-based methods identify sarcasm indicators or patterns based on lexical, syntactic, semantic, and pragmatic properties of the text. Lexicon-based methods utilize opinion word lexicons and sentiment analysis. Rule-based methods identify sarcasm indicators or patterns based on lexical, syntactic, semantic, and pragmatic properties of the text. Lexicon-based methods utilize opinion word lexicons and sentiment analysis. Machine learning approaches such as Support Vector Machines (SVM), naive Bayes, and the K-Nearest Neighbors algorithm were initially used with engineered features. More recently, deep learning models such as Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) networks have been applied.

2.1.2 Sarcasm Detection in Speech

In contrast to the extensive attention given to sarcasm detection in text, scholarly attention to sarcasm detection in speech is relatively low. However, the importance of detecting sarcasm in speech is vital and highly useful in products and spoken dialog systems.

Research in sarcasm speech detection primarily focuses on utilizing prosodic cues, which refer to acoustic changes that vary with the speaker's attitude regardless of phrase type. The most consistently observed prosodic correlate of sarcasm is the decrease in mean fundamental frequency (F0). Additionally, reductions in F0 standard deviation and changes in the harmonics-to-noise ratio (HNR) and voice quality are commonly associated with sarcasm (Cheang & Pell, 2008). Rockwell (2000) was among the pioneers in studying the vocal tonalities of sarcastic speech, identifying slower speech rates and greater intensity as potential markers of sarcasm. However, despite sarcasm being viewed as a prosody-weighted classification task, Tepperman et al. (2006) concluded that prosody on its own is not sufficient for reliably detecting sarcasm after testing the accuracy of detection of these cues on both an individual and combined basis in the phrase 'yeah, right'. They found that a combination of contextual and spectral cues distinguishes sarcasm from sincerity more accurately. Prosodic features such as intonation and stress are considered significant indicators of sarcasm (Bryant, 2010; Woodland & Voyer, 2011), and certain pitch and intensity contours are also predictive of sarcastic speech (Rakov & Rosenberg, 2013).

Prosthetic cues have been applied in sarcasm research in previous research in the field of speech technology. However, compared to similarly prosody-cued classification tasks like automatic emotion recognition, which have been extensively studied, research on sarcasm detection has been limited. Traditional statistical models, such as SVMs and decision trees, have been commonly used for sarcasm detection (Castro et al., 2019; Rakov & Rosenberg, 2013; Tepperman et al., 2006). For instance, Rakov and Rosenberg (2013) leveraged a SimpleLogistic (LogitBoost) classifier to predict sarcasm with 81.57% accuracy. In recent years, with the rapid development of deep learning in speech processing, several sarcasm detection methods based on deep neural networks have emerged. For example, Gao et al. (2022) proposed a transfer learning method based on DCNNs. Their results indicate that selected DCNN models, Xception and VGGish, improved sarcasm detection by 5% and 7%, respectively, compared to the SVM baseline model.

2.2 Self-supervised Learning in Speech

Over the past decade, deep learning methods have undergone a significant leap in performance, revolutionizing speech processing and enabling various practical applications. Supervised learning with deep neural networks has been pivotal in driving this transformation, achieving remarkable advancements in scenarios abundant with labeled data (Bourlard & Morgan, 2012; Hinton et al., 2012; LeCun, Bengio, & Hinton, 2015). However, the heavy reliance on supervised learning has simultaneously constrained progress in languages and domains lacking equivalent levels of labeling investment. Supervised learning not only hinges on costly annotations but also faces challenges such as generalization errors, spurious correlations, and adversarial attacks(X. Liu et al., 2021).

The emergence of self-supervised pre-training methods has partially mitigated these issues. This approach involves initially pre-training models on large-scale unlabeled data and then fine-tuning them on small-scale annotated data. self-supervised pre-trained models exhibit excellent generalization capabilities and often perform well on downstream tasks. Recently, self-supervised pre-training has demonstrated significant improvements across various machine learning domains, including speech and audio processing(Mohamed et al., 2022).

Based on the different pre-training objectives, self-supervised pre-training methods can be categorized into generative and contrastive approaches (X. Liu et al., 2021). Generative methods reconstruct original speech features using continuous or discrete latent variables; for example, autoencoders can predict future time frames or masked speech features. On the other hand, contrastive methods pre-train models through contrastive learning or predicting discrete indices; for instance, wav2vec 2.0 (Baevski et al., 2020), which utilizes convolutional neural networks(CNN) to predict masked audio; HuBERT (Hsu et al., 2021), which predicts masked hidden units on Transformer (Vaswani et al., 2017) encoder.

However, unlike self-supervised learning methods in computer vision and natural language processing fields, where a pre-trained model can adapt to various downstream tasks, wav2vec2.0 and HuBERT, despite significant advancements, have only been validated in ASR tasks. They are limited to single-speaker tasks and perform poorly in multi-speaker tasks such as speaker separation. Additionally, since these models are trained on the LibriLight (Kahn et al., 2020) dataset, their performance on out-of-domain downstream tasks was suboptim until the emergence of wavLM (S. Chen et al., 2022), which adopts Transformer encoder-decoder architecture to predict masked time frames and frequency bands information.

2.2.1 Wav2vec2.0 and HuBERT

After pre-training on 60,000 hours of data, both wav2vec2.0 and HuBERT achieved state-of-the-art (SOTA) performance on the speech recognition dataset Librispeech (Panayotov, Chen, Povey, &

Khudanpur, 2015). These methods use waveform as the model input and downsample it through a CNN module. The downsampled features are randomly masked and fed into a Transformer encoder.

Wav2vec2.0 employs contrastive learning for model training, discretizing the unmasked CNN outputs using a vector quantizer and calculating the InfoNCE loss on the masked positions' Transformer output representations, with positive samples from the discretized vectors at those positions and negative samples from other positions in the speech sequence.

On the other hand, HuBERT adopts the loss function of the masked language model from BERT (Devlin, Chang, Lee, & Toutanova, 2018) and utilizes a Transformer to predict the discrete ids of the masked positions for model training. HuBERT iteratively generates training targets, i.e., discrete ids for each frame. Researchers at Microsoft Research Asia (MSRA) initially applied k-means clustering to MFCC features of speech to generate discrete ids for training the first-generation HuBERT model. Subsequently, they clustered the output representations of the previously trained model and generated new ids for the next round of learning.

2.2.2 WavLM

WavLM is a recent self-supervised large pre-trained model, proposed by MSRA, built with Transformer modules, and trained on the largest-scale English speech (S. Chen et al., 2022). It not only learns speech recognition-related information through masked prediction tasks on speech but also enhances the potential for non-ASR tasks through speech denoising. Atmaja and Sasou (2022) evaluated self-supervised speech models for speech emotion recognition and found that compared to other pre-trained models, WavLM exhibits superior performance on emotion recognition tasks, which are also prosody-weighted classification tasks and can provide some insights into sarcasm speech detection.

Following the natural language pre-training Transformer model architecture pioneered by MSRA, researchers have proposed a pre-training scheme called Denoising Masked Speech Modeling.

The WavLM model comprises a convolutional encoder and a Transformer encoder. The convolutional encoder consists of 7 layers, each containing a temporal convolutional layer, a layer normalization layer, and a GELU (Hendrycks & Gimpel, 2016) activation function layer. In the Transformer encoder, researchers have introduced gated relative position encoding to incorporate relative positions into the computation of the attention network, thus better modeling local information. During training, WavLM randomly transforms input wavs, such as mixing two wavs or adding background noise. Subsequently, approximately 50% of the audio signal is randomly masked, and the model predicts the labels corresponding to the masked positions at the output. WavLM follows the idea proposed by HuBERT, using the K-means method to transform continuous signals into discrete labels and modeling them as targets. Specifically, given input speech X, its labels Y are first extracted. Then, X is noised and masked to generate , and the Transformer model predicts the labels Y of the masked positions based on the input .

WavLM was pre-trained on 94,000 hours of English speech, making it the largest-scale training data used by current open-source English models. Large-scale unsupervised speech data from various domains contributes to WavLM's enhanced robustness. Previous studies mostly relied on the LibriSpeech or LibriLight datasets for pre-training, which, being sourced from audiobooks, limited the model's generalization ability. Moreover, the speech environment in audiobooks differs from real-world scenarios, often characterized by more noise. Therefore, researchers utilized two additional datasets to augment the training data: (1) 10,000 hours of GigaSpeech (G. Chen et al.,

2021) data, collected from audiobooks, podcasts, and YouTube, covering various topics such as arts, science, and sports; (2) VoxPopuli (Wang et al., 2021) data. This is a large-scale, multilingual unlabeled audio dataset consisting of over 40,000 hours of audio in 23 languages, collected from European Parliament (EP) recordings from 2009-2020. Researchers only utilized 24,000 hours of English data from VoxPopuli for pre-training.

The Speech processing Universal PERformance Benchmark (SUPERB) (Yang et al., 2021) is an evaluation dataset jointly proposed by National Taiwan University, Massachusetts Institute of Technology (MIT), Carnegie Mellon University, and Meta, Inc. It includes 13 speech understanding tasks designed to assess the performance of pre-trained models. These tasks include Speaker Identification, Automatic Speaker Verification, Speaker Diarization, Phoneme Recognition, Automatic Speech Recognition, Keyword Spotting, Query by Example Spoken Term Detection (QbE), Intent Classification, Slot Filling, Emotion Recognition, Speech Separation, Speech Enhancement, and Speech Translation. Evaluation results on SUPERB demonstrate that WavLM surpasses previous pre-trained models and even outperforms the previous best-performing HuBERT large model with fewer parameters when fine tuning.

2.3 Parameter-Efficient Fine-Tuning Methods

Large pre-trained language models (PLMs) represent groundbreaking advancements across multiple application domains, achieving remarkable success in various tasks. However, their unprecedented scale brings significant computational costs. These models typically consist of billions of parameters, requiring substantial computational resources to function. Particularly when customizing them for specific downstream tasks, especially on hardware platforms with limited computational capacity, the expanded scale and computational demands pose considerable challenges.

Parameter-Efficient Fine-Tuning (PEFT) has emerged as the predominant methodology for adapting PLMs to downstream tasks. It consistently delivers robust performance across numerous popular NLP benchmarks without necessitating modifications to the pretrained architecture (Houlsby et al., 2019). Unlike conventional methods that involve fine-tuning the entire pre-trained model, PEFT offers the advantage of avoiding the need to store separate copies of model parameters for individual downstream tasks. This significantly reduces the computational resources required, particularly considering that modern pre-trained language models often contain hundreds of millions or even hundreds of billions of parameters (J. He, Zhou, Ma, Berg-Kirkpatrick, & Neubig, 2021).

2.3.1 Adpater

Houlsby et al. (2019) first proposed a adapter approach for BERT.They pointed out that when facing a specific downstream task, it is too inefficient to perform full-fintuning (all parameters in the pre-trained model are fine-tuned), while it is difficult to achieve better results if certain layers of the pre-trained model are fixed and only those layers close to the downstream task are fine-tuned. So they designed the adapter structure as shown in the figure 1.

The design idea of the adapter method is to insert a low-rank feed-forward neural network (FFN) module serially between the Transformer layers. This low-rank FFN module is the adapter, whose structure contains a layer of dimensionality reduction FFN, a layer of nonlinear transformation, a layer of ascending FFN, and the dimensionality of the dimensionality reduction has been compressed to a very small size, so that the introduction of the amount of adpaters'parameters is also very small.

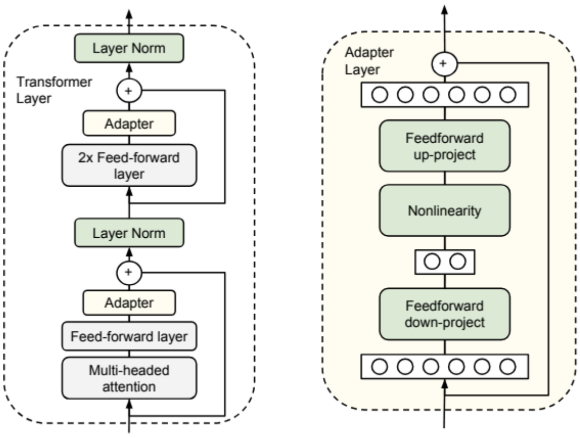

Figure 1: Two Adapter Modules Inserted In a Transformer Layer

2.3.2 Prefix-Tuning

Prefix-tuning is a lightweight alternative to fine-tuning for natural language generation tasks. Prefix-tuning keeps the language model parameters frozen and optimizes a small continuous task-specific vector which is called the prefix.The core idea of prefix-tuning is to add learnable prompts at the input layer of the model, which can be fixed-length vector sequences. During training, only these prompt vectors are updated, while the rest of the model remains unchanged. In this way, the model can adapt to different generative tasks by adjusting the prompt vectors without altering its original structure and parameters.

Li and Liang (2021) observed that by learning only 0.1% of the parameters in their experiment, prefix-tuning obtains comparable performance in the full data setting, outperforms fine-tuning in low-data settings, and extrapolates better to examples with topics unseen during training.

2.3.3 Prompt Tuning

Prompt Tuning can be seen as a simplification of prefix-tuning proposed by Li and Liang (2021). Instead of altering the deep structural weights of the model, prompt tuning refines the prompts that guide the model's responses. This method introduces soft prompts, which are tunable parameters added at the beginning of the input sequence.

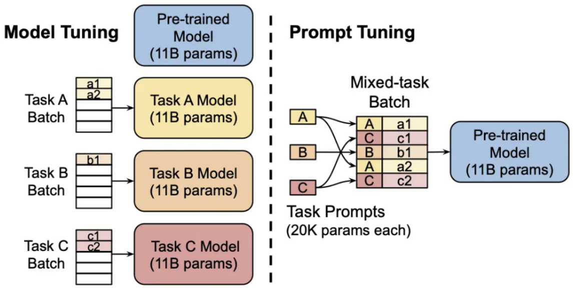
Model Tuning
Pre-trained Model (11B params)
Task A Batch
a1
a2
Task A Model (11B params)
Task B Batch
b1
Task B Model (11B params)
Task C Batch
c1
c2
Task C Model (11B params)
Prompt Tuning
Mixed-task Batch
A
B
C
A a1
C c1
B b1
A a2
C c2
Pre-trained Model (11B params)
Task Prompts (20K params each)

Figure 2: Model Tuning vs Prompt Tuning

Figure 2 contrasts model tuning with prompt tuning. In model tuning, each task necessitates a distinct model. In contrast, prompt tuning leverages the same foundational model across multiple tasks by refining task-specific prompts.

2.3.4 Low-rank Adaptation(LoRA)

Although some progress in the fine-tuning field, the existing adapter and prefix-tuning methods still have some problems:

1. **Adapters introduce inference latency**:The Adapter method uses a serial structure, the inserted adapter module can easily become a computational bottleneck, especially when the degree of parallelism is low (smaller batch, shorter length) for the model's computational efficiency has a greater impact.

2. **Prefix-tuning is hard**: The Prefix-tuning method uses a parallel structure, but the introduction of prefix tokens will take up the available input length of the model, resulting in poor scalability of Prefix-tuning, increasing the number of parameters will inevitably increase the number of prefix tokens, so that the available input length of the model will be even more serious crowding.

3. **Trade-off between efficiency and accuracy**: Efficiency and quality often involve a trade-off, and sacrificing one may lead to inferior results compared to full fine-tuning.

A neural network comprises numerous dense layers that execute matrix multiplication. The weight matrices in these layers generally possess full rank. When adapting to a specific task, Aghajanyan, Zettlemoyer, and Gupta (2020) shows that the pre-trained language models have a low "instrisic dimension" and can still learn efficiently despite a random projection to a smaller subspace.

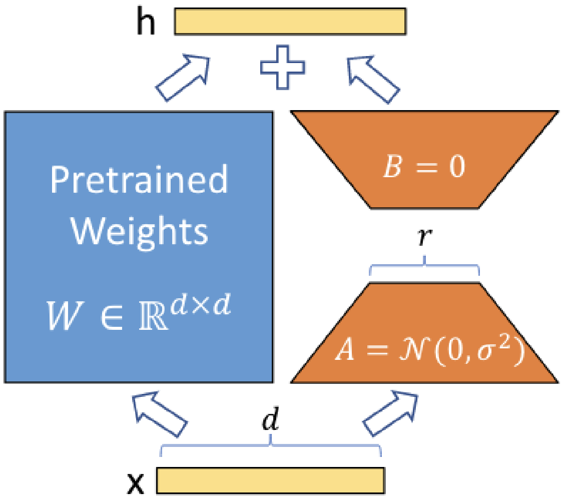
h
Pretrained Weights
$W \in \mathbb{R}^{d \times d}$
$B = 0$
r
$A = \mathcal{N}(0, \sigma^2)$
d
x

Figure 3: the Representation of LoRA

Hu et al. (2021) was inspired by this view and proposed Low-Rank Adaption (LoRA), which is designed as the figure 5, in the module involving matrix multiplication, A and B are introduced. In the module involving matrix multiplication, two low-rank matrix modules such as A and B are introduced to simulate the process of Full-finetuning, which is equivalent to updating only the low-rank instrisic dimension that plays a key role in the language model.

When the pre-trained weight matrix $W_0 \in R^{d\times k}$ where d represents the dimensionality of the input, and k represents the dimensionality of the output, the update is constrained by representing it with a low-rank decomposition $W_0 + \Delta W = W_0 + BA$, where $B \in R^{d\times r}$, $A \in R^{r\times k}$, and the rank $r \leq \min(d,k)$. During training, W_0 is frozen and does not receive gradient updates, while A and B contain trainable parameters. It's important to note that both W_0 and $\Delta W = BA$ are multiplied with the same input, and their respective output vectors are summed coordinate-wise. For $h = W_0 x$, the modified forward pass yields:

$$h = W_0 x + \Delta W x = W_0 x + BAx$$

A is initialized using random Gaussian initialization and B using zero initialization, so $\Delta W = BA$ is zero at the beginning of training. Then, $\Delta W x$ is scaled by αr, where α is a constant related to r. When optimizing with Adam, tuning α is roughly equivalent to tuning the learning rate. Therefore, α is set to the first r tried and not adjusted further. This scaling helps reduce the need to retune hyperparameters when r varies.

In the experiments(Hu et al., 2021), the researchers combined this LoRA module with the attention module of Transformer on several large models, namely RoBERTa (Y. Liu et al., 2019), DeBERTa (P. He, Liu, Gao, & Chen, 2020), GPT-2 (Radford et al., 2019)and GPT-3 (**?**), and the experimental results also fully proved the effectiveness of the method. In prosody-weighted speech emotional recognition field, loRA is shown to achieve better results compared to other PEFT methods (Feng & Narayanan, 2023).

3 Methodology

This section outlines the methodological approach adopted in this study for sarcasm speech detection. Subsection 3.1 introduces the pre-trained self-supervised model, WavLM Base+, which serves as the core architecture. Subsection 3.2 describes the downstream modeling classifier, a deep neural network-based binary classifier for downstream sarcasm detection task. Finally, subsection 3.3 focuses on the PEFT method employed in this study, specifically LoRA. The workflow is depicted in Figure 4.

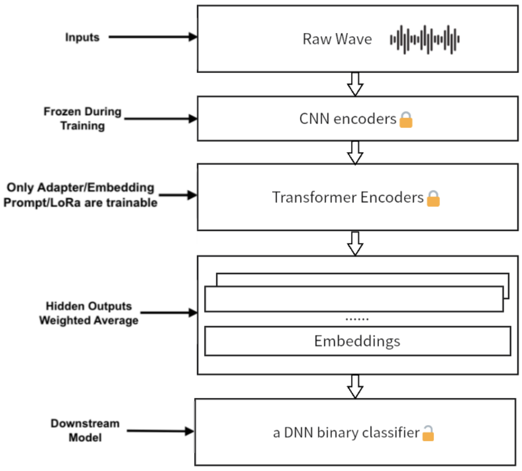

Figure 4: Modeling framework used in this work. The pre-trained model shown in the diagram is WavLM Base+

3.1 Pre-trained Self-supervised Model

There are varied WavLM models. Among them, WavLM Base+ is utilized as the core architecture of the current study for the following reasons:

Parameter number and computational efficiency: WavLM Large has 24 Transformer encoder layers, each with 1024-dimensional hidden states and 12 attention heads, totaling 316.62 million parameters. In contrast, WavLM Base+ has only 12 Transformer encoder layers, with each layer containing 768-dimensional hidden states and 8 attention heads, totaling 94.70 million parameters. The fewer parameters of WavLM Base+ enable higher computational efficiency and lower resource requirements when fine-tuning for sarcastic speech detection compared to WavLM Large.

Pre-training data and performance: WavLM Base and WavLM Base+ have the same structure and parameters, but WavLM Base+ was pre-trained for 1 million steps on a larger and more diverse dataset, while WavLM Base was only pre-trained for 400k steps on 960 hours of LibriSpeech audio. More pre-training data and steps help the model learn more features, enabling it to outperform WavLM Base overall and even surpass wav2vec 2.0 Large and HuBERT Large in overall scores. It performs exceptionally well on certain specific test sets (such as ASV, OOD-ASR, IC, SF, and ER). This indicates that WavLM Base+ excels in handling diverse audio tasks, making it more suitable for sarcastic speech detection than WavLM Base.

Given these two factors, and consider the data amount of theutilized dataset which is comparatively small, WavLM Base+ provides a good balance between performance and computational efficiency, making it a suitable choice compared to WavLM Large and WavLM Base.

3.2 Downstream Modeling classifier

The classification is conducted by using the downstream model which is a DNN-based binary classifier designed to process input feature vector sequences for classification predictions. It includes a feature extraction step where the mean of the input feature vector sequences is computed to obtain a feature representation for each sample. These features are then passed through a FNN consisting of two linear layers and a dropout layer. Finally, the model outputs a softmax-activated probability distribution over sarcastic class labels.

3.3 Parameter-efficient Fine-tuning Method - LoRA

After choosing LoRA as PEFT method for sarcasm speech detection, I used multiple LoRA ranks (1, 2, 4, 8, 16) in the experiment. The selection of these different ranks impacts both the performance and efficiency of the model.

A rank of 1 means that the model's weight matrix can be represented as an outer product of single vectors, significantly reducing the number of parameters. This is suitable for resource-constrained environments or scenarios demanding high computational efficiency but may limit the model's expressive capacity. Ranks 2 or 4 in LoRA maintain relative efficiency while preserving a certain level of model expressiveness, suitable for moderate-sized model compression needs. Higher ranks such as 8 or 16 provide greater model expressiveness by retaining more weight information, but they also increase the number of parameters and computational costs.

By experimenting and evaluating across different ranks, this study aims to identify the optimal LoRA configuration for sarcasm detection in speech, achieving the goal of PEFT, which is the trade-off between performance and efficiency.

4 Experimental Setup

In this section, I detail the experimental setup used to explore sarcasm speech detection. First, subsection 4.1 introduced the dataset used in this study. Then subsection 4.2 introduces the data processing which aims to prepare the data for the model input. Subsection 4.3 illustrates the training process, in which the evaluation method training hyperparameters are elaborated in details. Finally, subsection 4.4 will discuss the baseline methods and other two PEFT methods: adapter and embedding prompt tuning methods employed for comparison with the proposed LoRA method. Finally, Subsection 4.5 introduces the evaluation metrics used, namely recall, precision, and F-score, which are suitable for assessing the performance of binary classification tasks such as sarcasm speech detection.

4.1 Dataset - MUStARD++

The dataset I utilized for training, validating and testing the models is MUStARD++ (Aghajanyan et al., 2020). It is a multimodal dataset where each utterance is presented with visual, audio and textual modalities. It consists of 1202 instances, evenly split between sarcastic and non-sarcastic examples, making it suitable for evaluating sarcasm detection models.

- **Data Source** Videos from situational comedy TV shows like Friends, The Big Bang Theory, The Golden Girls, Burnistoun, and The Silicon Valley.
- **Annotations** Annotations in the MUStARD++ dataset include labels for sarcasm presence/absence, emotion categories (such as anger, excitement, fear, sadness, surprise, frustration, happiness, neutral, disgust, and ridicule), valence and arousal annotations to quantify emotion intensity, and sarcasm type labels (propositional, illocutionary, like-prefixed, embedded).
- **Additional Information** Additional information provided includes contextual details such as preceding utterances or sentences, as well as speaker information. These elements collectively enrich the dataset, enhancing its utility for research in multimodal sarcasm detection and emotion analysis.

Compared to the first multimodal sarcasm dataset MUStARD (Castro et al., 2019), MUStARD++ not only increases in data scale but also enriches and corrects annotation information, providing a more comprehensive resource for sarcasm detection. In conclusion, MUStARD++ serves as a benchmark multimodal dataset for sarcasm detection, advancing research in multimodal sarcasm detection.

4.2 Data Processing

I extract audios from videos in MUStARD++ and convert it into single channel and 16000 khz in order to be processed by wavLM base +.

Since a masked speech denoising and prediction framework were utilized to enhance model robustness for more complex acoustic environments such in the pre-trained model, (S. Chen et al., 2022), I haven't use any speech augmentation or denoising processing techniques, just use the raw waveform as the input for training, though the audios contain background noise, especially laughter from comedy shows.

Additionally, after inspecting the dataset's content, I decided to limit the maximum audio duration to 20 seconds to improve processing efficiency.

4.3 Training Process

The experiments were conducted using the PyTorch framework, leveraging its flexibility and powerful capabilities in deep learning tasks. Training was performed on the high-performance computing server Harbok provided by the University of Groningen, equipped with an A100 GPU. This powerful hardware infrastructure accelerated computation speed and model training.

Data Splitting of Subsets During the experiment, I first conducted a five-fold cross-validation. Considering that the MUStARD++ dataset contains only around 1200 data points, this approach aimed to comprehensively evaluate the model's performance and robustness with limited data. Through the five-fold cross-validation, I ensured multiple trainings, validations, and tests on different subsets of the data to obtain more reliable performance metrics. This method helps reduce the randomness introduced by a single split and enhances the credibility of the experimental results.

Hyper-parameter Setting and Tuning During training, I initially set the learning rate to 0.0005. This lower learning rate typically helps the model converge more steadily, reducing drastic fluctuations or oscillations during training. It also helps prevent early overfitting by limiting the magnitude of parameter updates per iteration, encouraging the model to learn general features of the data rather than specific sample details. I capped the maximum number of training epochs at 50 to ensure the model has sufficient time to learn without increasing computational cost and time consumption. For dynamic hyper-parameter tuning, I employed a learning rate scheduler and Adam optimizer.The scheduler adjusts the learning rate based on the validation loss at the end of each epoch, facilitating better convergence. The Adam optimizer, with settings for learning rate (lr) and weight decay (L2 regularization), manages model complexity to mitigate overfitting. For the batch size, given the size of MUStARD++, I opted for a batch size of 32 to balance computational efficiency while maintaining training speed and the model's generalization capability.

Transfer Learning To leverage the advantages of transfer learning, I used checkpoints from the Hugging Face model hub of the pre-trained model WavLM base+. These checkpoints served as a robust starting point, offering valuable knowledge learned from large-scale datasets.

Early Stopping The training will be ended early if the model shows no improvement on the validation set for multiple consecutive epochs (10 epochs), to prevent overfitting.

4.4 Comparison with Baselines and other PEFT methods

To thoroughly explore the effectiveness of LoRA in sarcasm speech detection, I conducted comprehensive experiments using different ranks of LoRA: 1, 2, 4, 8, and 16.

Baseline 1: Direct Embedding Extraction For this baseline method, I directly extracted embedding features from the pre-trained model WavLM and utilized them for training downstream models. This approach is straightforward and involves extracting representations directly from the pre-trained model without further modification. By using this method, I aimed to establish a baseline performance level for downstream tasks using raw embeddings.

Baseline 2: Fine-tuning Downstream Models with Frozen Pre-trained Encoders In this method, I kept the pre-trained encoders frozen and only fine-tuned the downstream models to adapt to sarcasm detection. This traditional fine-tuning approach involves freezing the parameters of the pre-trained encoders to preserve their learned representations while allowing the downstream models to adapt to the specific task. By fine-tuning only the downstream layers, I aimed to leverage the knowledge captured by the pre-trained model while tailoring the model to the sarcasm detection task.

Other Two PEFT Methods: Adapter Tuning and Embedding Prompt Tuning Additionally, in order to investigated the effectiveness of different PEFT methods in sarcasm detection, I comparing LoRA with two other PEFT approaches: adapter tuning and embedding prompt tuning. Notably, there are several ways to integrate adapters and inn this study, I attach the adapter to the output of the feed-forward layer.

4.5 Evaluation Metrics - Recall, Precision and F-score

As sarcasm speech detection is a binary prosody-weighted classification task, I utilize recall, precision, and F-score to comprehensively evaluate model performance.

Recall Recall refers to the proportion of true positives that are correctly identified by the model among all actual positives. In other words, recall measures the model's ability to recognize positives, indicating how well the model captures all true positives.

Precision Precision denotes the fraction of correctly predicted positive samples among all samples predicted as positive by the model. In simple terms, precision gauges the accuracy of the model's positive predictions, representing how many predicted positives are indeed correct.

F-Score F-score is the harmonic mean of recall and precision, providing a balanced assessment of a model's performance. The F-score is calculated as:

$$F = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

The F-score is commonly used to evaluate the overall performance of classification models as it simultaneously considers a model's recall and precision. It is robust against imbalanced datasets.

5 Results

In this section, I evaluated the performance of the WavLM Base+ pre-trained model using LoRA for sarcasm detection in speech. Specifically, I thoroughly explored the effectiveness of LoRA with different ranks (1, 2, 4, 8, 16). In addition to comparing Baseline 1 (direct feature extraction from the pretrained WavLM Base+) and Baseline 2 (fine-tuning only the downstream model), I also supplemented the comparison with two other PEFT methods: adapter tuning and embedding prompt tuning, to assess the effectiveness of different PEFT methods in sarcasm speech detection. The following subsections detail the model performance comparison in subsection 5.1, stability analysis in subsection 5.2, parameter requirements 5.3, and visual representations through loss curves 5.4 and confusion matrix 5.5.

5.1 Model Performance Comparison

Method	**Pre-trained Architecture**	**Precision**	**Recall**	**F-score**
Baseline 1	WavLM Base+	0.6085	0.6087	0.6069
Baseline 2	WavLM Base+	0.7233	0.7283	0.7164
LoRA (rank=1)	WavLM Base+	0.7077	0.7040	0.6993
LoRA (rank=2)	WavLM Base+	0.7227	0.7193	0.7145
LoRA (rank=4)	WavLM Base+	0.7247	0.7219	0.7191
LoRA (rank=8)	WavLM Base+	0.7351	0.7346	0.7328
LoRA (rank=16)	WavLM Base+	**0.7372**	**0.7360**	**0.7331**
Adapter tuning	WavLM Base+	0.7315	0.7349	0.7315
Embedding prompt tuning	WavLM Base+	0.7222	0.7199	0.7173

Table 1: Performance Comparison of Different PEFT Methods

Table 1 shows the average performance in 3 metrics: precision, recall and F-score. From rank 1 to 16, LoRA showed significant improvements over Baseline 1 in all three metrics. Although the performance at rank 1 was slightly lower compared to Baseline 2 (Precision: 0.7077, Recall: 0.7040, F-score: 0.6993), it matched Baseline 2 at ranks 2 (Precision: 0.7227, Recall: 0.7193, F-score: 0.7145) and 4 (Precision: 0.7247, Recall: 0.7219, F-score: 0.7191), and surpassed it at rank 8 (Precision: 0.7351, Recall: 0.7346, F-score: 0.7328) and rank 16 (Precision: 0.7372, Recall: 0.7360, F-score: 0.7331). Among these, rank 16 (Precision: 0.7372, Recall: 0.7360, F-score: 0.7331) achieved the optimal performance.

Among the two supplementary PEFT methods, adapter tuning (Precision: 0.7315, Recall:0.7349, F-score:0.7315) performed nearly on par with LoRA (rank=8), while embedding prompt tuning (Precision: 0.7222, Recall:0.7199, F-score:0.7173) did not achieve the performance levels of Baseline 2 across metrics.

Method	**Pre-trained Architecture**	**Precision Std**	**Recall Std**	**F-score Std**
LoRA (rank=1)	WavLM Base+	0.0153	0.0140	0.0174
LoRA (rank=2)	WavLM Base+	**0.0149**	**0.0117**	0.0140
LoRA (rank=4)	WavLM Base+	0.0256	0.0239	0.0255
LoRA (rank=8)	WavLM Base+	0.0179	0.0182	0.0195
LoRA (rank=16)	WavLM Base+	0.0157	0.0129	**0.0104**
Adapter tuning	WavLM Base+	0.0228	0.0232	0.0226
Embedding prompt tuning	WavLM Base+	0.0210	0.0196	0.0188

Table 2: Performance Standard Deviation of Different PEFT Methods

5.2 Model Performance Stability Comparison

Table 2 further illustrates the performance variability.At rank 2, LoRA exhibited moderate standard deviation at Precision (0.0149) and Recall (0.0117). Performance variability increased notably at rank 4 (Precision Std = 0.0256, Recall Std = 0.0239, F-score Std = 0.0255). Ranks 8 (Precision Std = 0.0179, Recall Std = 0.0182, F-score Std = 0.0195) and 16 (Precision Std = 0.0157, Recall Std = 0.0129, F-score Std = 0.0104) showed moderate variability. Notably, at rank 16, the F-score had the lowest standard deviation.

In addition, adapter tuning (Precision Std = 0.0210, Recall Std = 0.0232, F-score Std = 0.0226)exhibited higher performance variability across different datasets.

5.3 Parameter Requirements Comparison

Method	**Pre-trained Architecture**	**Number of Tuned Parameters (M)**
LoRA (rank=1)	WavLM Base+	**0.29**
LoRA (rank=2)	WavLM Base+	0.38
LoRA (rank=4)	WavLM Base+	0.57
LoRA (rank=8)	WavLM Base+	0.93
LoRA (rank=16)	WavLM Base+	1.67
Adapter tuning	WavLM Base+	0.24
Embedding prompt tuning	WavLM Base+	2.57

Table 3: Comparison of the Number of Tuned Parameters for Different PEFT Methods (M denotes Million)

The number of tuned parameters refers to the count of model parameters that are adjusted or optimized during the fine-tuning process of a pretrained model. Table 3 illustrates the parameter requirements when it comes to fine tune. As rank increased, the parameter requirements for LoRA also linearly increased. At rank=1, the parameter requirement was minimal at 0.29M. It was moderately increased to 0.93M at rank=8, and significantly higher at 1.67M at rank=16, indicating reduced efficiency.

Although adapter required lower parameters requirements at 0.24M when tuning, embedding prompt tuning required the highest parameters (2.57M) among the PEFT methods.

5.4 Five-Fold Cross-Validation Loss Curve

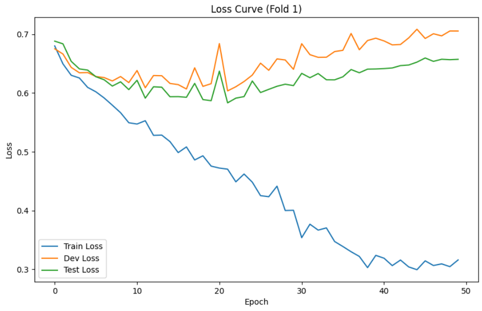

Figure 5: Loss Curve Fold 1

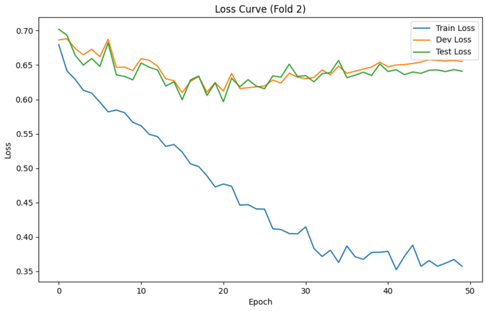

Figure 6: Loss Curve Fold 2

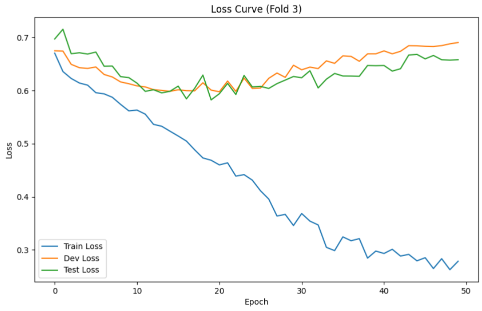

Figure 7: Loss Curve Fold 3

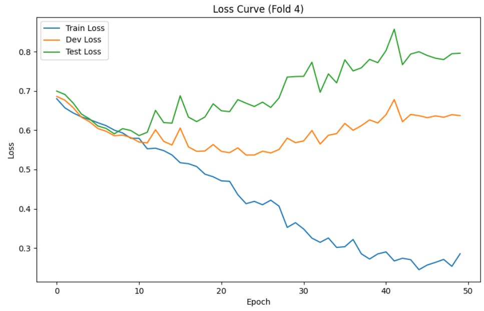

Figure 8: Loss Curve Fold 4

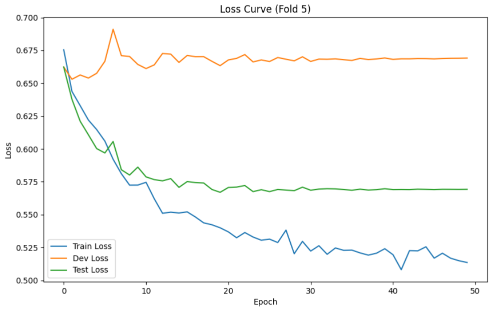

Figure 9: Loss Curve Fold 5

These five graphs depict the loss curves for each fold in the five-fold cross-validation using LoRA (rank=8).

Overall trends show that most datasets exhibited a steady decline in training loss, indicating significant progress in learning and fitting the training data. However, the loss curves for the development and test sets displayed some fluctuations and instability, suggesting room for improvement in model generalization.

Regarding overfitting, datasets from folds 1, 2, and 4 showed training loss significantly lower than development and test set losses in later stages, indicating potential overfitting. This suggests the model may have become too focused on the training data, resulting in decreased performance on unseen data. While folds 3 and 5 did not show clear signs of overfitting, the gap between their loss curves also indicates some risk of overfitting.

The best performance was observed in the fifth fold, where the model achieved relatively low levels of loss across all measures. The second fold followed closely behind, with low development and test set losses. Conversely, performance was relatively poorer in folds 1, 3, and 4, particularly with higher losses observed in fold 4.

In terms of fluctuations, all folds except the fifth showed some degree of variability in their loss curves, especially noticeable in the test set curves. These fluctuations could stem from differences in data distribution or instability during model optimization. Fold 4 exhibited particularly significant.

5.5 Confusion Matrix

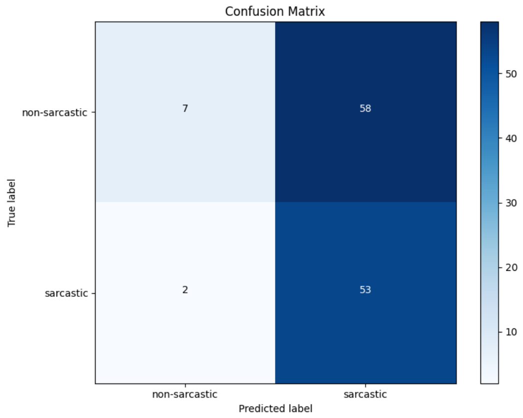

Figure 10: Confusion matrix at the beginning of training

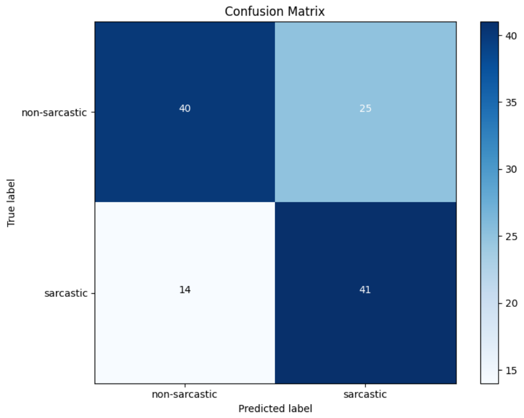

Figure 11: Confusion matrix at the end of training

I created confusion matrices to visually inspect the model's performance before and after training on a total of 120 test samples (65 non-sarcastic and 55 sarcastic).

The initial confusion matrix 10 shows that the model performs quite well in recognizing sarcastic samples, correctly identifying 53 out of 55 sarcastic samples. However, it struggles significantly in recognizing non-sarcastic samples, misclassifying many as sarcastic.

After a period of continued training 11, the model has made substantial progress in recognizing non-sarcastic samples. The number of correctly identified non-sarcastic samples has increased significantly, from only 7 initially to 40.

However, it's worth noting that while improving its ability to recognize non-sarcastic samples, the model's accuracy in identifying sarcastic samples has slightly declined. Initially identifying 53 sarcastic samples correctly, it now correctly identifies 41, misclassifying more sarcastic samples as non-sarcastic.

6 Discussion

In this section, I first analyzed the performance of the WavLM-LoRA method in sarcasm speech detection in subsection 6.1, aiming to answer the research question:"Can the parameter-efficient fine-tuning method LoRA adapt the self-supervised pre-trained speech model WavLM for speech sarcasm detection, and how effective is it?" The results demonstrate LoRA superiority over the baseline methods and other two PEFT methods across multiple evaluation metrics, supporting the hypothesis. LoRA (rank=8) strikes a good balance between performance and parameter efficiency, although it may not achieve the highest scores across all metrics. In subsection 6.2, I discussed the phenomenon of overfitting during training and its potential causes, and proposed future research directions including addressing data imbalance, mitigating speaker dependency bias, enhancing model robustness to noise, and optimizing fine-tuning strategies. These efforts aim to develop more accurate, robust, and efficient sarcasm speech detection systems, advancing research in this field.

6.1 Result Analysis

The results demonstrate the effectiveness of the WavLM-LoRA method, surpassing Baseline 1, which uses the WavLM pre-trained model for direct vector extraction in downstream model training, across evaluation metrics such as F-score, recall, and precision. This strong performance aligns with hypothesis 1, suggesting that fine-tuning generally enhances the performance of pre-trained models on specific tasks. The WavLM model with the LoRA-based PEFT method outperforms direct extraction of embedding features from the pre-trained WavLM for sarcasm speech detection.

Furthermore, LoRA at ranks 8 and 16 outperforms Baseline 2, the traditional fine-tuning method that freezes the encoder and only fine-tunes the downstream model. It also surpasses two other PEFT methods: adapter tuning and embedding prompt tuning. This comprehensive superiority supports hypotheses 2 and 3, indicating that applying the LoRA-based PEFT on the WavLM model enhances sarcasm detection performance compared to traditional methods and other PEFT approaches.

From an integrated analysis of the three tables, LoRA (rank=8) represents a balanced solution between performance and parameter efficiency. Although it may not achieve the best scores across all metrics, its lower parameter requirement (0.93M) strikes a good balance between performance and efficiency. Increasing the rank to 16 yields limited performance improvement for LoRA but significantly increases parameter demands, thus reducing computational efficiency. Therefore, unless solely pursuing optimal performance, rank 16 is not the ideal choice. The superior performance of LoRA can be attributed to the robust speech representations learned by WavLM through self-supervised pre-training (S. Chen et al., 2022), and LoRA efficiently adapts to sarcasm detection tasks while retaining most of the pre-training parameters. The introduction of task-specific inductive biases through rank decomposition parameter updates allows LoRA to effectively utilize the learned representations of WavLM (Hu et al., 2021).

From ranks 1 to 16, the WavLM-LoRA method shows better performance and the larger parameter count indicates that higher ranks capture more features and details from the data, albeit at the cost of increased computational resources. Additionally, adapter tuning shows relatively lower performance and parameter requirements compared to LoRA, making it another efficient fine-tuning method for sarcasm detection in speech, aside from LoRA. This analysis supports the hypothesis that the LoRA technique demonstrates superior performance in terms of F1 score, recall, and precision metrics in sarcasm speech detection compared to other PEFT methods.

Despite employing early stopping and regularization techniques, the training process exhibits some overfitting tendencies (Hawkins, 2004), evident in the cross-validation loss curves. This may be attributed to the relatively small MUStARD++ dataset size and the inherent complexity of sarcasm detection tasks. Potential solutions include exploring advanced regularization methods such as mixup (Zhang, Cisse, Dauphin, & Lopez-Paz, 2017), cutmix (Yun et al., 2019), or adversarial training (Hawkins, 2004) to enhance model generalization capabilities.

Furthermore, an interesting observation is that the confusion matrix shows the model efficiently detects sarcasm instances in the early stages of training. This observation might be related to the characteristics of the MUStARD++ dataset, which includes discourse from comedy shows and may be easier to distinguish based on exaggerated rhythmic clues (Aghajanyan et al., 2020). However, recognizing non-sarcastic instances poses challenges, possibly due to subtle rhythmic patterns or background noise in non-sarcastic speech.

6.2 Limitation and Future Research

Although this research has achieved encouraging results in sarcasm speech detection, there are still some notable limitations and directions for future research. In terms of data, the imbalance between sarcastic and non-sarcastic instances in the MUStARD++ dataset, the potential speaker dependency features, and the significant background noise from real-life comedy scenes may have impacted the model's generalization ability. In the future, techniques such as oversampling (Mohammed, Rawashdeh, & Abdullah, 2020), undersampling, and class-weighted loss functions could be explored to mitigate the data imbalance issue; speaker separation and normalization methods could be studied to alleviate speaker dependency bias; and targeted speech enhancement techniques could be adopted to improve the model's robustness against background noise, thereby better capturing the subtle prosodic cues required for sarcasm detection.

In terms of the model, while the WavLM Base+ achieved encouraging results, exploring larger model variants such as WavLM Large and other self-supervised models could potentially further improve performance, although at the cost of increased computational demands. Additionally, designing more complex downstream classifier architectures or ensembling multiple models may more effectively capture the complexities of the sarcasm detection task; and comprehensively tuning optimization hyperparameters such as learning rate scheduling and batch size could potentially yield further performance gains, but would require more time and computational resources for extensive experimentation.

Finally, in terms of the fine-tuning method, combining LoRA with other parameter-efficient adaptation methods (such as adapters or prompt tuning) could potentially provide orthogonal improvements and further enhance model performance; while in cases where computational resources permit, fully fine-tuning the WavLM model could serve as a performance upper bound reference. By addressing these existing limitations and thoroughly exploring optimization directions across data, model, and fine-tuning methods, it will be possible to build more accurate, robust, and efficient sarcasm speech detection systems, contributing to the further development of research in this field.

7 Conclusion

In this thesis, I explored the application of PEFT using LoRA with the self-supervised pre-trained speech model WavLM for sarcasm detection in speech. Through comprehensive experimentation and analysis, I demonstrated the effectiveness of this approach in improving model performance while significantly reducing parameter requirements.

My results showed that the LoRA-based PEFT approach outperformed the baselines, including directly using the pre-trained WavLM embeddings and traditionally fine-tuning only the downstream classifier. Additionally, LoRA exhibited superiority over other PEFT methods, such as adapter tuning and prompt tuning, in terms of F1 score, recall, and precision metrics for sarcasm speech detection. These findings highlight the potential of LoRA as an efficient and effective PEFT technique for adapting large-scale pre-trained models to downstream tasks, particularly in the domain of prosody-weighted speech classification.

While my work has demonstrated promising results, there are still limitations and opportunities for future research. Exploring more diverse datasets, incorporating contextual information, and investigating the interpretability of the models could further enhance the performance and applicability of sarcasm detection in speech. Additionally, extending the LoRA-based PEFT approach to other pre-trained speech models and tasks could broaden its impact in the field of voice technology.

Overall, this thesis has contributed to the advancement of sarcasm detection in speech by introducing the application of PEFT, specifically the LoRA technique, with the self-supervised pre-trained model WavLM. The findings and insights gained from this research pave the way for continued innovation and progress in this crucial area of natural language understanding and human-computer interaction.

References

Aboobaker, J., & Ilavarasan, E. (2020). A survey on sarcasm detection approaches. *Indian J. Comput. Sci. Eng*, *11*(6), 751–771.

Aghajanyan, A., Zettlemoyer, L., & Gupta, S. (2020). Intrinsic dimensionality explains the effectiveness of language model fine-tuning. *arXiv preprint arXiv:2012.13255*.

Ashwitha, A., Shruthi, G., Shruthi, H., Upadhyaya, M., Ray, A. P., & Manjunath, T. (2021). Sarcasm detection in natural language processing. *Materials Today: Proceedings*, *37*, 3324–3331.

Atmaja, B. T., & Sasou, A. (2022). Evaluating self-supervised speech representations for speech emotion recognition. *IEEE Access*, *10*, 124396–124407.

Baevski, A., Zhou, Y., Mohamed, A., & Auli, M. (2020). wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, *33*, 12449–12460.

Băroiu, A.-C., & Trăușan-Matu, (2022). Automatic sarcasm detection: Systematic literature review. *Information*, *13*(8), 399.

Bourlard, H. A., & Morgan, N. (2012). *Connectionist speech recognition: a hybrid approach* (Vol. 247). Springer Science & Business Media.

Bryant, G. A. (2010). Prosodic contrasts in ironic speech. *Discourse Processes*, *47*(7), 545–566.

Carvalho, P., Sarmento, L., Silva, M. J., & De Oliveira, E. (2009). Clues for detecting irony in user-generated contents: oh...!! it's" so easy";-. In *Proceedings of the 1st international cikm workshop on topic-sentiment analysis for mass opinion* (pp. 53–56).

Castro, S., Hazarika, D., Pérez-Rosas, V., Zimmermann, R., Mihalcea, R., & Poria, S. (2019). Towards multimodal sarcasm detection (an _obviously_ perfect paper). *arXiv preprint arXiv:1906.01815*.

Cheang, H. S., & Pell, M. D. (2008). The sound of sarcasm. *Speech communication*, *50*(5), 366–381.

Chen, G., Chai, S., Wang, G., Du, J., Zhang, W.-Q., Weng, C., ... others (2021). Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio. *arXiv preprint arXiv:2106.06909*.

Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., ... others (2022). Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, *16*(6), 1505–1518.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ding, N., Qin, Y., Yang, G., Wei, F., Yang, Z., Su, Y., ... others (2023). Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, *5*(3), 220–235.

El Ayadi, M., Kamel, M. S., & Karray, F. (2011). Survey on speech emotion recognition: Features, classification schemes, and databases. *Pattern recognition*, *44*(3), 572–587.

Feng, T., & Narayanan, S. (2023). Peft-ser: On the use of parameter efficient transfer learning approaches for speech emotion recognition using pre-trained speech models. In *2023 11th international conference on affective computing and intelligent interaction (acii)* (pp. 1–8).

Gao, X., Nayak, S., & Coler, M. (2022). Deep cnn-based inductive transfer learning for sarcasm detection in speech. In *23rd interspeech conference* (pp. 2323–2327).

Hawkins, D. M. (2004). The problem of overfitting. *Journal of chemical information and computer* *sciences*, *44*(1), 1–12.

He, J., Zhou, C., Ma, X., Berg-Kirkpatrick, T., & Neubig, G. (2021). Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*.

He, P., Liu, X., Gao, J., & Chen, W. (2020). Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., … others (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, *29*(6), 82–97.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., … Gelly, S. (2019). Parameter-efficient transfer learning for nlp. In *International conference on machine learning* (pp. 2790–2799).

Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., & Mohamed, A. (2021). Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, *29*, 3451–3460.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., … Chen, W. (2021). Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Joshi, A., Bhattacharyya, P., & Carman, M. J. (2017). Automatic sarcasm detection: A survey. *ACM Computing Surveys (CSUR)*, *50*(5), 1–22.

Kahn, J., Riviere, M., Zheng, W., Kharitonov, E., Xu, Q., Mazaré, P.-E., … others (2020). Libri-light: A benchmark for asr with limited or no supervision. In *Icassp 2020-2020 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 7669–7673).

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436–444.

Lester, B., Al-Rfou, R., & Constant, N. (2021). The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Li, X. L., & Liang, P. (2021). Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Liu, X., Zhang, F., Hou, Z., Mian, L., Wang, Z., Zhang, J., & Tang, J. (2021). Self-supervised learning: Generative or contrastive. *IEEE transactions on knowledge and data engineering*, *35*(1), 857–876.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., … Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Mohamed, A., Lee, H.-y., Borgholt, L., Havtorn, J. D., Edin, J., Igel, C., … others (2022). Self-supervised speech representation learning: A review. *IEEE Journal of Selected Topics in Signal Processing*, *16*(6), 1179–1210.

Mohammed, R., Rawashdeh, J., & Abdullah, M. (2020). Machine learning with oversampling and undersampling techniques: overview study and experimental results. In *2020 11th international conference on information and communication systems (icics)* (pp. 243–248).

Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015). Librispeech: an asr corpus based on public domain audio books. In *2015 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 5206–5210).

Potamias, R. A., Siolas, G., & Stafylopatis, A.-G. (2020). A transformer-based approach to irony and sarcasm detection. *Neural Computing and Applications*, *32*(23), 17309–17320.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.

Rakov, R., & Rosenberg, A. (2013). " sure, i did the right thing": a system for sarcasm detection in speech. In *Interspeech* (pp. 842–846).

Rockwell, P. (2000). Lower, slower, louder: Vocal cues of sarcasm. *Journal of Psycholinguistic research*, *29*, 483–495.

Suhaimin, M. S. M., Hijazi, M. H. A., Alfred, R., & Coenen, F. (2017). Natural language processing based features for sarcasm detection: An investigation using bilingual social media texts. In *2017 8th international conference on information technology (icit)* (pp. 703–709).

Tepperman, J., Traum, D., & Narayanan, S. (2006). " yeah right": sarcasm recognition for spoken dialogue systems. In *Ninth international conference on spoken language processing.*

Thomas, B., Kessler, S., & Karout, S. (2022). Efficient adapter transfer of self-supervised speech models for automatic speech recognition. In *Icassp 2022-2022 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 7102–7106).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Wang, C., Riviere, M., Lee, A., Wu, A., Talnikar, C., Haziza, D., ... Dupoux, E. (2021). Voxpopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. *arXiv preprint arXiv:2101.00390.*

Woodland, J., & Voyer, D. (2011). Context and intonation in the perception of sarcasm. *Metaphor and Symbol*, *26*(3), 227–239.

Yang, S.-w., Chi, P.-H., Chuang, Y.-S., Lai, C.-I. J., Lakhotia, K., Lin, Y. Y., ... others (2021). Superb: Speech processing universal performance benchmark. *arXiv preprint arXiv:2105.01051.*

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., & Yoo, Y. (2019). Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the ieee/cvf international conference on computer vision* (pp. 6023–6032).

Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412.*