UNIVERSITY OF CAMBRIDGE

Neural Spoken Language Assessment Systems

Submitted May 2020, in partial fulfillment of
the conditions for the award of the degree **MEng Engineering.**

Vyas Raina

Supervised by Dr Kate Knill

Department of Engineering

University of Cambridge

I hereby declare that this dissertation is all my own work, except as indicated in the text:

Signature _____

Date _____ / _____ / _____

Neural Spoken Language Assessment Systems

Supervised by Dr Kate Knill

Vyas Raina

May 2020

Technical Abstract

With increasing demand for English language learning, there has been growth in popularity of automated spoken language (SLA) systems. These systems take the speech from a candidate and predict a grade for that candidate. Additionally, it is useful for the SLA system to provide feedback to justify the grade given and to aid the learning progress of the candidate. SLA systems are required to assess many aspects of the candidate's speech, ranging from pronunciation and intonation, to use of English and grammatical structures. This work is primarily focused on what a candidate says, the word sequence uttered, rather than how the words are pronounced. These text features have been found to be the most important in grade prediction for SLA systems. As with many other areas in speech and language processing, deep-learning is used in many state-of-the-art SLA systems. This work examines two aspects of applying deep learning to SLA. First, rather than using expert derived features, this work examines using deep-learning approaches to replace these expert features. The second area is related to malpractice/cheating. Deep-learning systems are known to be susceptible to adversarial attacks: these are small, imperceptible, perturbations in the observation aimed at changing the system output. In the context of SLA systems, adversarial attacks can be performed at a text level (small changes in the words uttered) or in more general at a waveform level (perturbations in the audio signal), where the aim is to dramatically improve the predicted grade.

A challenge for the design of SLA systems is the variable length nature of the audio input. Traditional SLA systems use expert features, derived from the raw audio and the word sequence output of an automatic speech recognition system (ASR), to form a fixed length representation of the input. Typical audio features capture audio energy, intensity and speaking rate, whilst textual features include unique words, standard perplexity scores and simple word frequencies. A further advantage of these features is that they can

be used to provide comprehensive feedback to the candidate. As an alternative to the feature-based SLA systems, this work explores the use of deep learning (neural) systems to derive a fixed length representation of the input, meaning that the fixed length vector is directly optimised for the task. This approach has the benefit of extracting information from the input in a manner that is not pre-determined and can thus identify aspects of the input not explicitly considered before in SLA systems.

The challenge that arises for neural systems is their increased susceptibility to adversarial attacks. In this work, SLA systems that solely use the text output of the ASR system (text based SLA) are subject to a universal adversarial attack. In real applications, SLA systems are presented as black-box models that an adversary can query. Due to the large number of queries required, it is impractical to expect an adversary to identify a unique adversarial attack for each candidate's response, motivating the consideration of a universal black-box attack. In the context of text-based SLA systems, the form of attack is a short sequence of words appended to any candidate's standard response that causes an increase in the output predicted grade. Although neural systems are vulnerable, this work presents four detection schemes, including the use of perplexity scores, deep ensembles and SLA specific off-topic response detection. However, the most successful detection approach is based on detection shifts from the scores of a "traditional" Gaussian Process, feature-based Grader. Therefore, the challenge of adversarial attacks for text-based neural SLA systems can be overcome with well designed detection schemes.

A full SLA system considers not only the textual information output from an ASR but also the raw audio waveform. Therefore, this work generalises the text-based adversarial attacks to waveform adversarial attacks. As before, the challenge is with black-box SLA systems, compounded in this case by the continuous nature of a waveform input - discrete optimisation methods are no longer applicable. One approach to this is to use an "evolutionary" process, where only the inputs and outputs to the system are considered. It is revealed in this work, nonetheless, that even evolutionary processes struggle to find strong waveform attacks.

The SLA systems in this work are used for multi-level prompt-response free speaking tests, where candidates from a range of proficiency levels provide open responses to prompted questions.

Acknowledgements

First, I would like to thank my supervisor Dr Kate Knill for constant, unwavering support and guidance throughout this year. Without her affability and constant availability for even my most trivial obstacles, I would still be wrestling with the likes of `qsub`. Second, this acknowledgement would be incomplete without mentioning Prof. Mark Gales; an integral part of this journey, offering indispensable ideas and lots of enthusiasm.

It goes without saying that family and friends are pillars of support for any project, but here, in 2020, as I sit in lockdown at home with my family, their contribution demands an explicit mention. Interlacing the writing of this thesis with sport, Zoom calls, cooking, gardening and games, my parents, siblings, dog and friends (virtually) have made writing this thesis a truly enjoyable experience.

I would also like to thank Cambridge English Language Assessment for support and access to the Linguaskill data. Finally, I want to thank members of the ALTA Speech Team for help in a large number of areas, too numerous to enumerate here.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Due to the increasing demand for English language learning, there has been a growth in popularity of automated spoken language assessment (SLA) systems. The aim of a SLA system is to predict a grade for a recorded candidate response. Moreover, it is useful for the SLA system to provide feedback to aid the learning progress of the candidate. When assigning a grade, human examiners take into account various aspects of the response, encapsulated by the term "construct". The construct captures the following ideas: task achievement; coherence of the speech; use of grammar and vocabulary; the use of time for the replies; candidate hesitations and pronunciation including stress and rhythm. SLA systems seek to match the human predictions. The initial greatest challenge for SLA systems is the variable length nature of the audio input. Any automated grading system requires fixed size inputs, meaning the design of a SLA system requires a stage to transform variable length inputs into a fixed length vector.

Traditional approaches to tackle the challenge of variable length inputs use feature-based SLA systems. An automatic speech recognition (ASR) system converts the audio signal into a textual transcription. Then, explicitly defined features can be extracted from the textual output (e.g. number of unique words) and the original variable length audio (e.g. waveform energy) to form a fixed length vector representation that can be passed through a downstream Grader. The success of deep learning (neural) SLA systems in a number of speech [13, 41] and natural language processing (NLP) [38, 45] tasks motivates the use of neural systems as an alternative to the feature extraction process. Neural systems can be

designed to use both the ASR output and the original audio signals and then transform them into a fixed length vector via a deep structure.

However, the use of a deep learning approach to SLA brings with it a new challenge. Neural networks are known to be inherently susceptible to mal-practice in the form of adversarial attacks [39]. An adversarial attack is concerned with small, undetectable perturbations in the input yielding undesired changes in the output. Beyond assessing a candidate's English speaking ability, it is necessary to ensure that a system is robust to mal-practice. The integrity and reliability of an examination is threatened when there exist means by which a candidate can take actions to cheat the system.

When an adversary has access to the the internal structure of a system (network architecture), the form of adversarial attack is termed a *white-box* attack [3]. However, it is improbable that an adversary seeking to fool an automated spoken language assessment system will have access to the internal workings of the system. Therefore, this work restricts itself to black-box, targeted adversarial attacks, where the adversary has no knowledge of the system (only access to input and output pairs of the model) and seeks to fool the system into predicting a desired output. Black-box attacks are grouped into query based approaches [6, 14, 46] and transfer based approaches [33, 31, 10]. Due to the need for a large number of queries, the former approach is easy to defend against. The transfer approach relies on similar structure models being susceptible to the same adversarial samples [18]. Recent studies [24, 44, 15], have demonstrated successful transfer of attacks, but only in situations where the networks are extremely similar in structure.

For SLA systems it is possible to attack either the audio signal (waveform attack) or the word sequence uttered (text-based attack). As features, either expert or deep-learning based, derived from the word-sequence are found to accurately predict the grade, this work emphasises text-based attacks. A wide range of simple techniques [26, 17, 34] can be employed to construct adversarial attacks. However, due to the discrete nature of the input, the text sequence, gradient based adversarial attacks are difficult to implement [22]. A range of text based attacks and detection approaches have been described in the literature [42, 1]. For SLA systems this text is derived from a speech recognition system; thus the vocabulary is fixed. This means that attacks such as character-level replacement [11, 47] cannot be used. In this work a greedy discrete search method for the

adversarial attack is adopted. In particular a universal attack is considered [2], where a single phrase is found that, for SLA, will increase the predicted score. Using a universal attack reduces the opportunity for detection, as the attack needs to only be trained once and just requires the candidate to learn a set phrase.

To overcome the challenge presented by the susceptibility of neural systems to adversarial attacks, there exist a range of general approaches to attack detection [32, 27]. For text-based attacks, this work examines the use of perplexity scores [22] and deep ensembles [37, 27] approaches, as well as a SLA specific off-topic response detection approach [35]. Additionally, a detection approach based on a second, feature-based, SLA system is also described.

An automated assessment system is only considered valid and the outputs meaningful when the automated system employs similar assessment criteria as the human counterparts who designed the examination. Hence, it is necessary to consider not only text-based SLA systems but also more complete systems that use the raw audio signals. As before, neural approaches used for these complete SLA systems give rise to the challenge of threatening adversarial attacks, where the attack is at a waveform level (on the audio signal). In contrast to text-based attacks, waveform attacks exist in the continuous space of the original audio signals. Therefore, simple discrete optimisation approaches are not applicable. Often, for such black-box attacks, an evolutionary approach can be used [16]. However, it can be challenging to find a useful initialisation point for evolutionary approaches.

1.1 Thesis Outline

- Chapter 2 introduces the overall structure for an automated SLA system.
- Chapter 3 describes alternative architecture designs for the Grader stage of the overall structure.
- Chapter 4 describes adversarial attacks and defences performed at a text level.
- Chapter 5 describes more generalised adversarial attacks performed at a waveform level.
- Chapter 6 presents the results of experiments: baseline performance of Graders; text adversarial attacks and waveform adversarial attacks.
- Chapter 7 presents high-level conclusions and directions for future work.

Chapter 2

Spoken Language Assessment

2.1 Assessment Structure

A SLA system predicts a grade (e.g. six point CEFR grade scale [7]) from a candidate's spoken response. Typically, automated SLA systems use a regression approach, where the output is a score on a continuous scale that can be mapped to the fixed grade classifications. There are two main components to an automated SLA system: an ASR system and a Grader. Figure 2.1 shows that with these components, a variable-length audio signal, $o_{1:n}$, is mapped to a single score, y, associated with a grade. The ASR system is necessary for transforming an audio signal into a rich representation, $w^*_{1:L}$, consisting of the variable length word sequence $w_{1:L}$ and other useful information (section 2.2). The Grader $\mathcal{G}$ can use both the variable length audio signal and ASR output to obtain the score (Equation 2.1).

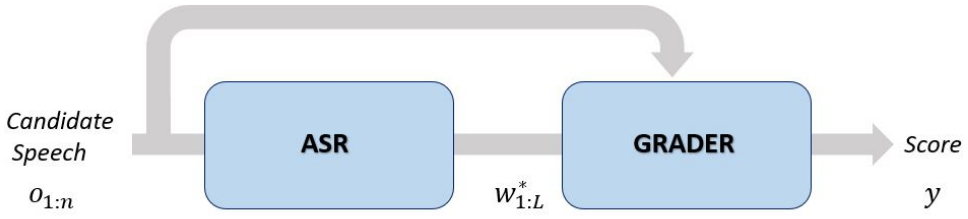

Figure 2.1: Automated Spoken Language Assessment structure

$$y = \mathcal{G}(o_{1:n}, w^*_{1:L}) \tag{2.1}$$

2.2 Automatic Speech Recognition

The process of deriving a transcription (word sequence) $w_{1:L}$, from the speech waveform $o_{1:n}$, is automatic speech recognition. As opposed to a single word sequence, state of the art ASR systems output a set of N-best hypotheses. These hypotheses can form a directed acyclic graph, termed a lattice (e.g. Figure 2.2), where arcs indicate hypothesised words and nodes represent the start and end times of the outgoing and incoming words. This lattice can then be used to output the one best sequence ($w_{1:L}$), associated phones [8], time stamps, word confidence scores and word durations (e.g. Figure 2.3). Thus, in a SLA system, the Grader can use a rich input $w^*_{1:L}$.

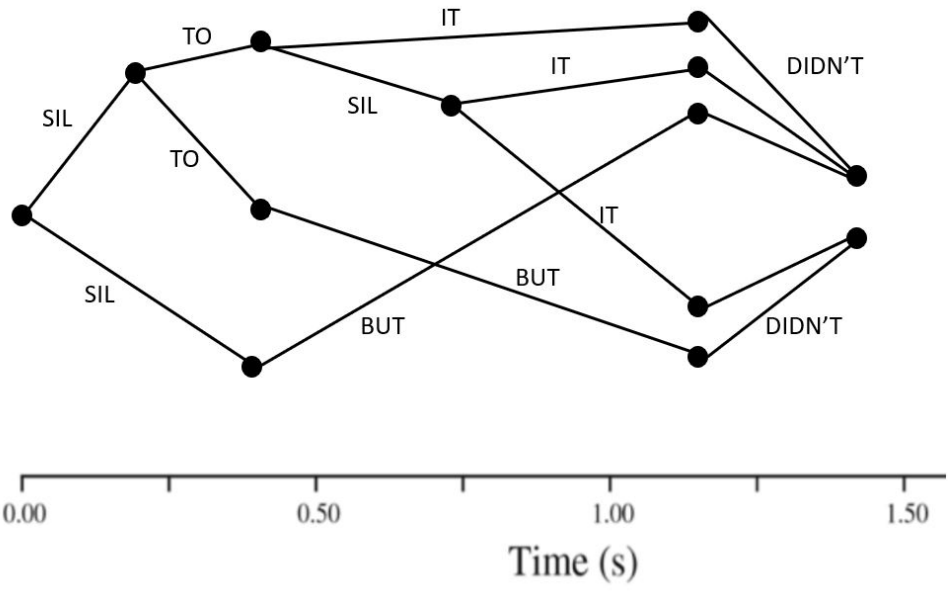

Figure 2.2: ASR Lattice Output.

2.3 Grader

As given in Equation 2.1, a Grader is required to output a single score y from variable length inputs $o_{1:n}$ and $w^*_{1:L}$. An obvious challenge for the automated systems is the variable

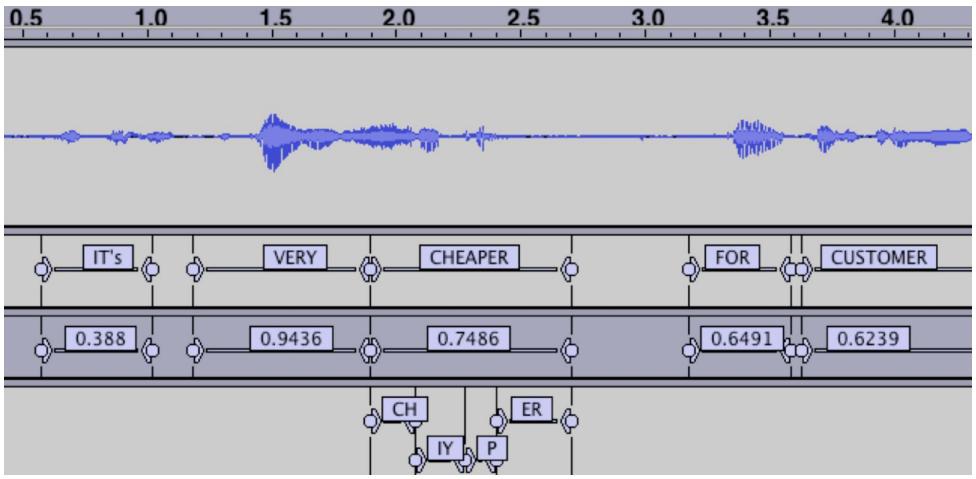

Figure 2.3: Example ASR outputs: one best sequence, phones, timestamps and word durations. Image sourced from ALTA Speech Team.

length nature of the inputs. This is traditionally overcome using feature-based Graders, where explicit human derived features are used to define a fixed length vector. Figure 2.4 shows that in feature-based Graders, feature-extractors (FE) can be used to extract text related features $\mathbf{x}^{(t)}$ and audio features $\mathbf{x}^{(a)}$, which can be concatenated to form a single fixed length vector $\mathbf{x}$. Equation 2.2 then shows how a feature Grader $\mathcal{G}_F$ determines the score y from the feature vector.

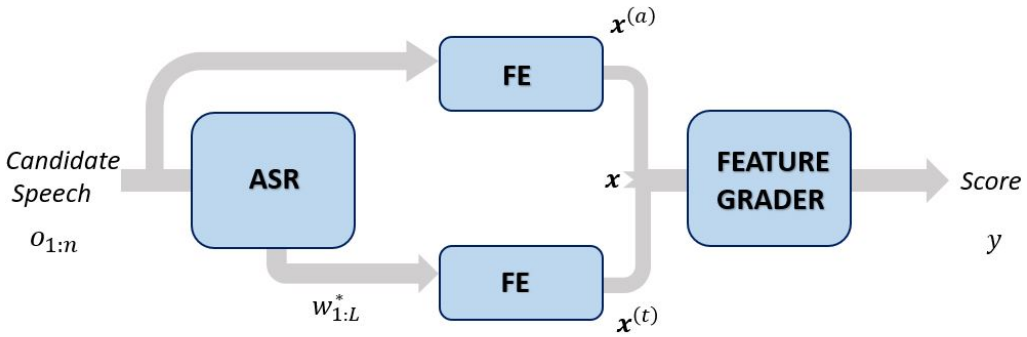

Figure 2.4: Automated Spoken Language Assessment with feature based grading.

$$y = \mathcal{G}_F(\mathbf{x}) \tag{2.2}$$

Beyond handling variable length inputs, these traditional feature based Graders are able to provide informative feedback. The human derived features can be specifically chosen to mimic human examiners (increases validity) and also then be used to identify the weaknesses in a candidate's speech. Typical audio features will capture a speaker's fluency,

audio fundamental frequency and audio energy, whilst text-based features may simply look at distinct word frequencies. Features can also be calculated from the richer ASR output to obtain features for time-aligned transcriptions at multiple levels, e.g. speaking rate.

Advanced Graders, avoiding the use of traditionally defined feature vectors, are discussed in section 3.2.

Chapter 3

Grader

This section details the specific structures of Graders used in SLA systems. Initially, different forms of traditional feature-based Graders, introduced in section 2.3, are described. Then, state of the art deep learning based Graders are discussed.

3.1 Feature-based Grader

3.1.1 GP Feature-based Grader

Any feature-based Grader uses $\mathbf{x}$ (from Figure 2.4) as a fixed length vector input. The Grader $\mathcal{G}_F$ (Equation 2.2) can be designed using a Gaussian Process (GP) system [40, 43]. A GP model is a non-parametric model used for regression. It can be understood as modelling a distribution over functions. As there are no parameters to the model, all the training points $\{\mathbf{x}^{(s)}, y^{(s)}\}_{s=1}^{S}$, of S speakers are stored. When used in a Grader, the GP system is to map a feature vector $\mathbf{x}$ to a score. A GP is defined over functions f and is fully specified by its mean function $m(\mathbf{x})$ (Equation 3.1) and covariance function $k(\mathbf{x}, \mathbf{x}')$ (Equation 3.2), where E is the expectation operator.

$$m(\mathbf{x}) = E[f(\mathbf{x})] \tag{3.1}$$

$$k(\mathbf{x}, \mathbf{x}') = E\left[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))\right] \tag{3.2}$$

Therefore, the entire Gaussian Process can be represented as a distribution as given in equation 3.3.

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \tag{3.3}$$

Predictions can then be made on an unseen data point $\mathbf{x}^{(*)}$ by computing the posterior over f. The output is assumed to be Gaussian distributed around a mean $f(\mathbf{x}^{(*)})$. For the purpose of obtaining a single score for the Grader $\mathcal{G}_F$, this output mean is used.

3.1.2 DNN Feature-based Grader

An alternative to the GP system for the feature-based Grader is to use a Deep Neural Net (DNN) system [29]. A DNN is a stack of fully connected layers, where each layer applies a linear transformation to an input fixed length vector, followed by a non-linear activation. Therefore, a DNN Grader can take the fixed length feature vector $\mathbf{x}$ as an input and output a single value y to be interpreted as the score.

As with the GP Grader, a set of training data points $\{\mathbf{x}^{(s)}, y^{(s)}\}_{s=1}^{S}$ are required for a DNN feature-based Grader. However, the purpose of this training data is to perform supervised training of the DNN model's parameters $\boldsymbol{\theta}$. Training of a DNN system $\mathcal{G}_F$ parameters is typically performed by minimising the mean squared error (MSE) loss function $\mathcal{L}$, given in Equation 3.4. The optimal, learnt parameters $\hat{\boldsymbol{\theta}}$ then precisely define the DNN system to be used for any new data points $\mathbf{x}^{(*)}$, such that the predicted score is simply $\mathcal{G}_F(\mathbf{x}^{(*)}; \hat{\boldsymbol{\theta}})$.

$$\mathcal{L}(\theta) = \sum_{s=1}^{S} (y^{(s)} - \mathcal{G}_F(\mathbf{x}^{(s)}; \boldsymbol{\theta}))^2 \tag{3.4}$$

3.2 Deep Neural Grader

Deep learning can be used to eliminate the need for features to be explicitly defined, as required for the feature-based Graders of section 3.1. However, once again, the challenge of dealing with variable length inputs $o_{1:n}$ and $w^*_{1:L}$ (Figure 2.1) arises. Figure 3.1 proposes a generic deep learning based (neural) structure to obtain a fixed length vector $\mathbf{h}$ from the variable inputs, which can then be input to a neural Grader to obtain the score y. It can be noted that this structure is similar to the feature-based grading structure of Figure 2.4. However, now there are no longer human derived features $\mathbf{x}$, but instead the fixed length representation $\mathbf{h}$ is determined from a deep learning based structure that is optimised in conjunction with a downstream neural Grader.

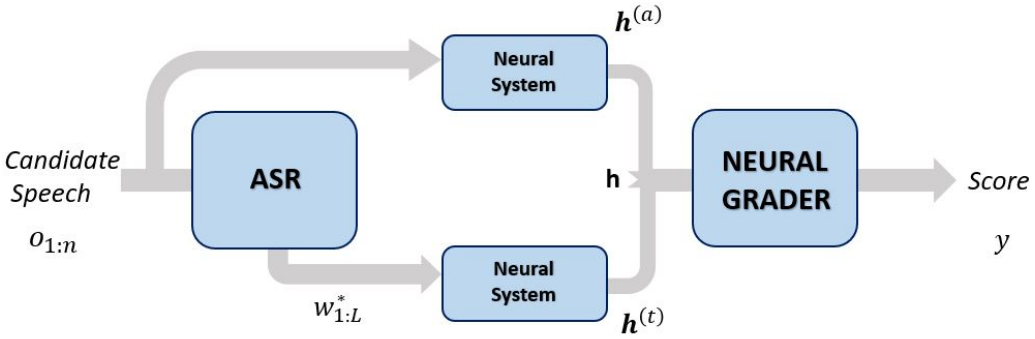

Figure 3.1: Automated Spoken Language Assessment with neural grading.

As is the case for the DNN feature-based Graders (section 3.1.2), neural Graders are parameterised by $\boldsymbol{\theta}$. However, for neural Graders, these parameters also parameterise the Neural systems (Figure 3.1) used to obtain the fixed length vector $\mathbf{h}$. Hence, the training data for supervised learning of neural Grader parameters, using MSE as the loss function, takes the form of $\{o^{(s)}_{1:n}, w^{*(s)}_{1:L}, y^{(s)}\}_{s=1}^{S}$.

This section focuses on neural Graders that have been restricted to using solely the ASR textual output $w_{1:L}$ (text neural Grader) or a richer form of the ASR textual output $w^*_{1:L}$ (Audio text neural Grader). This work does also considers a phone based neural Grader [20] in section 5 that uses the audio signal $o_{1:n}$, as well as the rich ASR output $w^*_{1:L}$ as its inputs.

3.2.1 Text Neural Grader

If the deep neural Grader is restricted such that the Grader can only use the ASR textual output to predict the grade, Equation 2.1 simplifies to Equation 3.5

$$y = \mathcal{G}(w_{1:L}) \tag{3.5}$$

As mentioned, to overcome the challenge of variable length inputs, a dedicated system is required to obtain the fixed length vector $\mathbf{h}$. The design of the Neural Grader can be split into three steps: embedding generation, multi-head attention and a DNN (3.2), where the first two stages achieve the fixed length representation of the input.

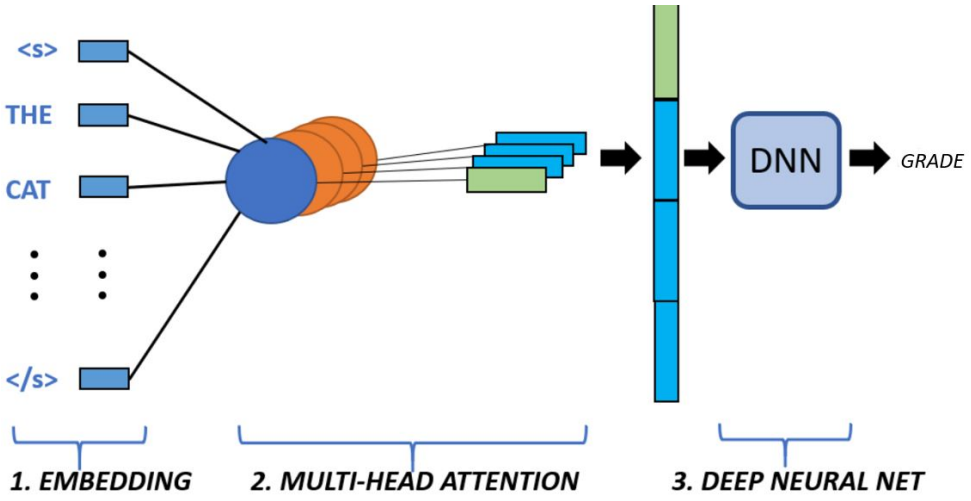

Figure 3.2: Text Neural Grader Architecture.

The embedding generation stage takes the variable L-length word sequence $w_{1:L}$ and creates L fixed size vectors $\mathbf{v}_{1:L}$. Often BERT [9] embedding generation perform better than other approaches such as word2vec [30], as BERT is able to capture context. In the second stage, the attention mechanism attends over the embeddings $\mathbf{v}_{1:L}$ and combines them linearly to produce a single fixed length embedding $\bar{\mathbf{v}}$. This linear combination is shown in Equation 3.6. The coefficients $\alpha_{1:L}$ (Equation 3.7) are found by performing the softmax function (to ensure the coefficients are positive and sum to one) on the attention scores $s_{1:L}$, where the attention scores are computed using a combining function $\mathcal{F}$ of a *key*, $\mathcal{K}_{1:L}$ and the values, $\mathbf{v}_{1:L}$ (Equation 3.8). Often self-attention is used, where the key is also the value, such that $\mathcal{K}_i = \mathbf{v}_i, \;\; i \in (1, L)$.

$$\bar{\mathbf{v}} = \sum_{i=1}^{L} \alpha_i \mathbf{v}_i \tag{3.6}$$

$$\alpha_i = \texttt{softmax}(s_{1:L}) = \frac{\exp(s_i)}{\sum_{j=1}^{L} \exp(s_j)} \tag{3.7}$$

$$s_i = \mathcal{F}(\mathcal{K}_i, \mathbf{v}_i) \tag{3.8}$$

Multi-head attention is where the attention mechanism is performed in parallel N_h (number of heads) times. The outputs from each attention head can then be simply concatenated. This vector is then the fixed-length vector $\mathbf{h}$ that can be passed through a DNN in the third stage required to obtain the score y.

3.2.2 Audio Text Neural Grader

The neural Grader described above in section 3.2.1 suffers from a lack of validity as human examiners would use both textual and audio features in determining the grade. Hence, the neural Grader can be extended to include audio information by using word duration and word confidence scores (Figure 3.3), obtained from the richer ASR output $w^*_{1:L}$ (section 2.2). The duration and confidence scores can be treated as separate two dimensional vectors $\mathbf{d}_{1:L}$. The difference in the architecture is the use of a Recurrent Neural Net (RNN) [36, 23], used to increase the dimension of $\mathbf{d}_{1:L}$. In its simplest form, a RNN is made of a sequence of L units for the L inputs $\boldsymbol{d}_{1:L}$. Each unit, has a hidden state $\mathbf{q}_i$, computed from linear transformations $\mathbf{A}$ of the current input and previous inputs (Equation 3.9), with an added bias term $\mathbf{b}$. Typically, bi-directional RNNs are used such that there exist hidden states $\tilde{\mathbf{q}}_{1:L}$, computed from the inputs in reverse (Equation 3.10). The ith output from the RNN can then simply be formed by concatenating $\mathbf{q}_i$ and $\tilde{\mathbf{q}}_i$.

$$\mathbf{q}_i = \mathbf{A}_1 \mathbf{d}_i + \mathbf{A}_2 \mathbf{q}_{i-1} + \mathbf{b} \tag{3.9}$$

$$\tilde{\mathbf{q}}_i = \tilde{\mathbf{A}}_1 \mathbf{d}_i + \tilde{\mathbf{A}}_2 \tilde{\mathbf{q}}_{i+1} + \tilde{\mathbf{b}} \tag{3.10}$$

The variable length output of the RNN is then passed into an attention mechanism as before (section 3.2.1) to obtain the fixed length representation $\mathbf{h}$ that is passed through a DNN to obtain the score y.

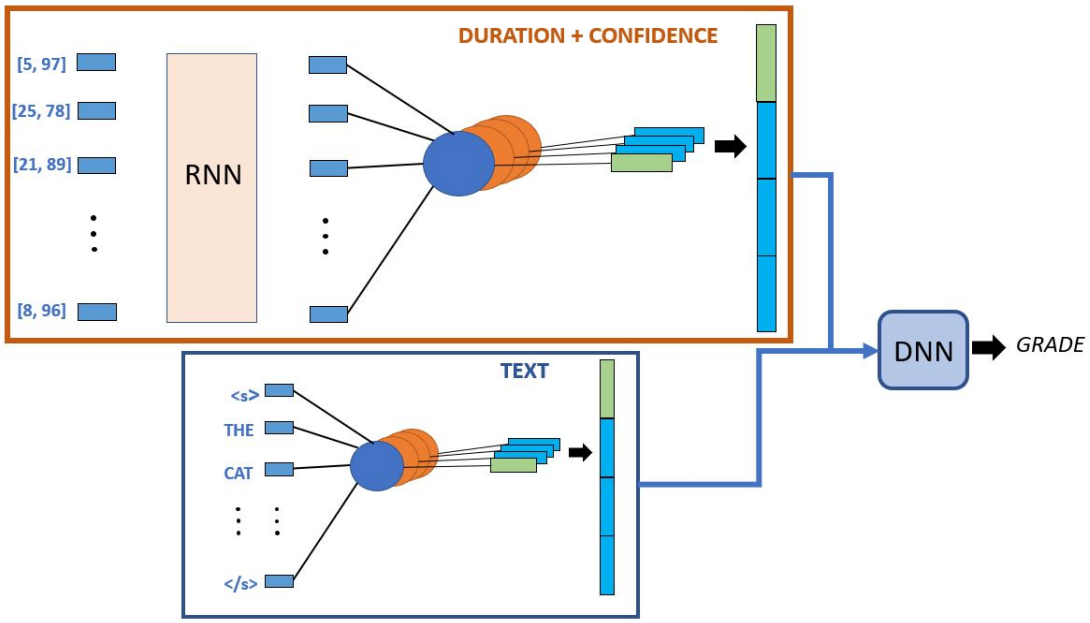

Figure 3.3: Audio Text Neural Grader Architecture.

Chapter 4

Text Adversarial Attacks

It is expected that the textual content of a candidate's speech has the most significant contribution to the score prediction y. Therefore, it is useful to initially focus on text-based Graders that only use the textual output $w_{1:L}$ of an ASR system. The parametric neural textual Graders in section 3 are designed to learn the optimal model parameters $\hat{\boldsymbol{\theta}}$ to perform well on the training data $\{w_{1:L}^{(s)}, y^{(s)}\}_{s=1}^{S}$ of S speakers, where the mean squared error loss function (defined for example in Equation 3.4) is used as the assessment criterion. However, the blind optimisation approach of these neural Graders means that they are susceptible to adversarial attack mal-practice. This section outlines an approach to overcome this challenge by identifying and detecting universal adversarial attacks, i.e. is there a single phrase that any candidate can utter along with their standard response to guarantee an increase in the predicted grade from the trained Graders $\mathcal{G}$ with parameters $\hat{\boldsymbol{\theta}}$.

4.1 Attack Strategy

In literature, the general form of a targeted attack is

$$\hat{\boldsymbol{\delta}} = \arg\min_{\boldsymbol{\delta}} \{\mathcal{G}(\boldsymbol{x} + \boldsymbol{\delta}) = t\} \qquad \text{s.t.} \quad \mathcal{H}(\boldsymbol{x}, \boldsymbol{x} + \boldsymbol{\delta}) < \epsilon \tag{4.1}$$

where t is the required target outcome from the classifier $\mathcal{G}()$, $\boldsymbol{x}$ is the observation to be attacked, $\mathcal{H}()$ is some "distance" between the observation, and ϵ is a threshold at which value the perturbation on the observation $\boldsymbol{\delta}$ is deemed to be noticeable.

As this work considers adversarially attacking the text-based Grader of a free-speaking spoken language assessment system, here rather than changing the classification outcome, the task is to maximally increase the predicted score of the regression based Graders from the ASR output, $w_{1:L}$ [1]. The aim is to for example append to the end of a legitimate response a short sequence of words that are in the vocabulary of the ASR system that maximally increase the average score. As only universal attacks are considered, the metric to maximise in adversarial attacks is the average (across speakers) score $\sum_{s=1}^{S} y^{(s)}$. Thus

$$w_{1:L} \oplus \boldsymbol{\delta}^{(k)} = w_1, \ldots, w_n, \tilde{w}_1, \ldots, \tilde{w}_k \tag{4.2}$$

where the k word adversarial attack is $\tilde{w}_1, \ldots, \tilde{w}_k$. The cost function to be optimised can then be written as

$$\hat{\boldsymbol{\delta}}^{(k)} = \arg\max_{\boldsymbol{\delta} \in \mathcal{V}^k} \left\{ \sum_{s=1}^{S} \mathcal{G}(\boldsymbol{w}^{(s)} \oplus \boldsymbol{\delta}^{(k)}; \hat{\boldsymbol{\theta}}) \right\} \tag{4.3}$$

where $\boldsymbol{w}^{(s)}$ is the recognised word sequence for candidate s, $\hat{\boldsymbol{\theta}}$ are the optimal Grader parameters learnt (section 3.2) and $\mathcal{V}^k$ is the set of all k length word-sequences that can be constructed using the ASR vocabulary $\mathcal{V}$. The constraint that the adversarial attack must comprise words in the vocabulary of the speech recognition system means that character-based adversarial attacks for example are not possible.

4.1.1 Discrete Optimization

It is assumed that a single adversarial phrase of length k is to be used for all candidates (universal attack). Though an adversarial attack could in theory be generated for each candidate's recognised word sequence, this is not a realistic scenario, as the given operating mode for this form of attack is likely to be in the form of a black-box attack.

[1] Exactly how the adversarial output from the ASR system is produced is not considered. It is possible that the candidate could simply speak the word sequence, assuming that the ASR is accurate, or the ASR system itself may be adversarially attacked.

As black-box adversarial attacks are most realistic for SLA systems, it is not possible to optimise the attack using knowledge of the network architecture. In this work an explicit, discrete optimisation approach is adopted. This is challenging as searching all possible words in the vocabulary is expensive requiring a large number of queries. Additionally if context-dependent word embeddings, such as BERT [9], are used then adding any word alters the embeddings for all other words. To address this problem a two stage approach is used. Initially a transfer-based approach is adopted, where a simple context-independent word-embedding based substitute system [33] is used to select a subset of words from the complete vocabulary [2]. This subset can then be used to query the real system to select the optimal word. This approach is felt to be realistic as only a single universal phrase is needed for all speakers. The adversarial attack is generated in a "greedy" fashion where

$$\hat{\boldsymbol{\delta}}^{(k)} = \arg\max_{\delta \in \tilde{\mathcal{V}}} \left\{ \sum_{s=1}^{S} \mathcal{G}(\boldsymbol{w}^{(s)} \oplus \hat{\boldsymbol{\delta}}^{(k-1)} \oplus \delta; \hat{\boldsymbol{\theta}}) \right\} \tag{4.4}$$

where $\tilde{\mathcal{V}}$ is the subset vocabulary determined by the substitute system. The number of system queries thus increases linearly with the length of the adversarial attack and the size of the subset vocabulary. The attack defined has only considered appending a phrase at the end of an utterance. In practice, appending to the end of a standard response is the simplest for a candidate, but other positions can be considered.

4.2 Defence Strategy

The objective of this section is to explore a plethora of techniques that can be used to identify textual inputs to a text based Grader that have been artificially manipulated (i.e. an adversarial attack).

[2] Here knowledge of the vocabulary is assumed. In practice, provided the selected subset is large enough, this knowledge is not necessary.

4.2.1 Topic Relevance

One of the standard approaches to detecting malpractice, as well as detecting when a candidate cannot generate an appropriate response to a prompt, is to use off-topic response detection. In terms of malpractice the aim is to detect when a candidate is uttering some pre-learnt phrase that is not related to the prompt. The task being considered here is more challenging. Here it is assumed that the candidate can generate an appropriate response $w_{1:L}$. Thus the off-topic response detection extracts the impact of the adversarial attack on the relevance. Therefore the detection mechanism is based on

$$P(\texttt{rel}|\boldsymbol{p}, w_{1:L} \oplus \tilde{w}_{1:k}) < \beta \tag{4.5}$$

where $\boldsymbol{p}$ is the prompt associated with the response. The β defines a threshold that can be tuned to achieve the best decision boundary.

Here a hierarchical attention based topic model (HATM) for off-topic spontaneous response detection [28, 35] is proposed. In addition it is useful if the system is trained on a held-out data set which contains prompt-response examples that cover all the prompts seen in the test set.

4.2.2 Perplexity

The form of adversarial attack in Equation 4.3 imposes no constraints on the words being appended to the original sequence. It is therefore possible that by using a language model of "standard" non-native speakers of English it is possible to detect the adversarial attacks. As it is not possible to guarantee the location of the attack it is necessary to consider the average perplexity of the complete sequence, normalised by the sequence length. As a refinement to this basic model, a grade-dependent language model can be used, which is based on the predicted grade from the neural assessment system. The metric used to assess whether an adversarial attack is being used is:

$$\texttt{perp}(w_{0:L} \oplus \tilde{w}_{1:k}; \hat{g}) > \beta \tag{4.6}$$

where an initial sentence start symbol is added as w_0, $\hat{g}$ is the predicted grade from the neural assessment system, and

$$\texttt{perp}(\boldsymbol{w}_{0:L}; \hat{g}) = \frac{1}{L} \sum_{i=1}^{L} \log(P(w_i | w_{0:i-1}; \hat{g}) \tag{4.7}$$

The decision boundary threshold β can be altered to obtain the most appropriate decision boundary.

It is common in many language modelling tasks that an LSTM-based recurrent neural network language model is used[3]. As the vocabulary of the output is restricted to be the vocabulary of the ASR system, there is no need to consider out-of-vocabulary words when computing the perplexities.

4.2.3 Ensemble Diversity

When an ensemble of M models (typically using different seed initialisations) are used for a Grader, the use of ensembling can be exploited in defending against adversarial attacks. One approach for detecting adversarial attacks is to examine the consistency of the ensemble of predictors. It is expected that inputs that have been manipulated will result in less agreement in the output of the ensemble of models. Therefore, for the regression task, the variance of the ensemble predictions can be used. Thus

$$\frac{1}{M} \sum_{i=1}^{M} \left[\mathcal{G}(w_{1:L}; \hat{\boldsymbol{\theta}}^{(i)}) \right]^2 - \left[\frac{1}{M} \sum_{i=1}^{M} \mathcal{G}(w_{1:L}; \hat{\boldsymbol{\theta}}^{(i)}) \right]^2 > \beta \tag{4.8}$$

The threshold β can be adjusted to select the most appropriate decision boundary.

4.2.4 GP Shift

Rather than using an ensemble of models based on the same form of model, it is possible to use a model with a completely different configuration. For example, it is probable that a neural Grader is susceptible to different types of adversarial phrases than a feature-based

[3]Example of a specific model is the CUED-RNNLM v1.1 toolkit [5]

GP Grader (section 3.1.1). Thus it may be suitable to use a GP Grader for determining an adversarial attack on a neural Grader. Figure 4.1 demonstrates that a benign input, when adversarially attacked for a deep neural Grader to form a malicious input, is likely to result in a larger increase in the deep neural prediction than the GP prediction. Thus, the distance to the Line Of Best (LOB) fit, parallel to the deep neural prediction axis, is expected to increase. Therefore, a simple approach can be adopted based on:

$$\mathcal{G}(w_{1:L}; \hat{\boldsymbol{\theta}}) - \texttt{map}(\mathcal{G}_{\text{gp}}(\boldsymbol{\phi}(w_{1:n}))) > \beta \tag{4.9}$$

where $\boldsymbol{\phi}()$ is the feature extraction process for the word-sequence. Rather than using the raw predicted GP-score, a mapped version is used based on a linear mapping (LOB), $\texttt{map}()$ from the GP-score to the neural assessment score estimated on a held-out data set. This should handle, for example, the mismatch in the average scores from the two forms of grading. As in other detection approaches, the decision boundary can be adjusted using the threshold β.

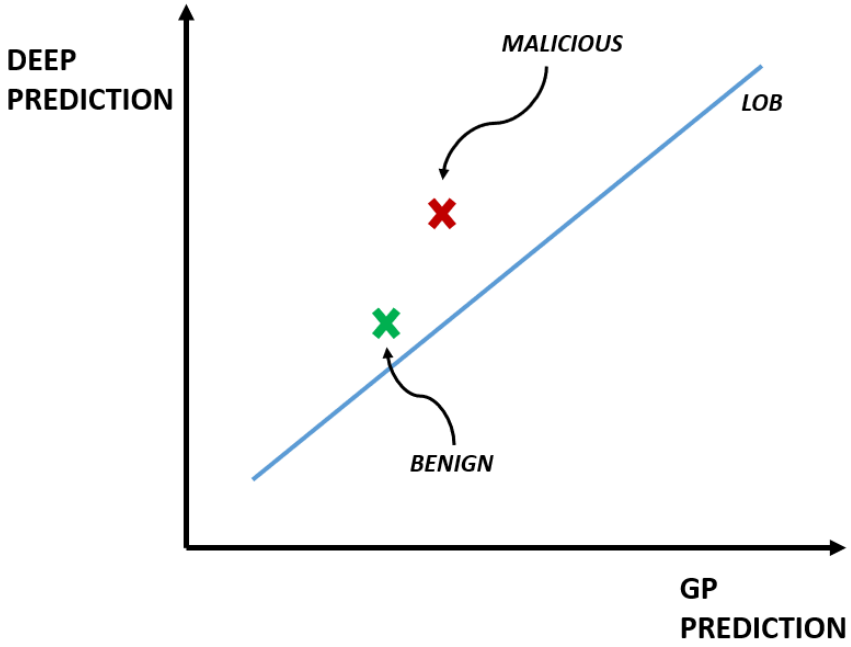

Figure 4.1: Schematic of impact of adversarial attack on Deep Neural Grader vs impact on GP Grader.

Chapter 5

Waveform Adversarial Attacks

Waveform adversarial attacks can be understood as a generalisation of the text adversarial attacks. Instead of a attacking a Grader that only takes the word sequence $w_{1:L}$ as the input, the attack concerns modifying the variable length audio input $o_{1:n}$. The greatest challenge for these attacks is that the SLA systems are treated as black-box systems, meaning only the input, output pairs $\{o_{1:n}^{(s)}, y^s\}_{s=1}^{S}$ are available. Thus, it is assumed that the optimised model parameters $\hat{\boldsymbol{\theta}}$ are fixed.

5.1 Universal Waveform Attack

Waveform adversarial attacks are concerned with manipulating the raw audio waveform $o_{1:n}$ inputs to an ASR system or directly to a Grader $\mathcal{G}$, where the aim is to make an imperceptible (to human ears) change to the input, whilst causing an increase in the output predicted score (consistent with Equation 4.1). In a similar structure to the text based adversarial attack (Equation 4.2), the change in the original raw audio segment $o_{1:n}$ is from an adversarial noise perturbation $\boldsymbol{\delta}$ (Equation 5.1).

$$o_{1:n} \oplus \boldsymbol{\delta}^{(n)} = o_1 + \delta_1, \ldots, o_n + \delta_n \tag{5.1}$$

As the adversarial attack has to be universal, the same noise signal has to be added to

any original audio segment. However, a challenge is that the input audio segments have a variable length n. Therefore, it simpler that $\boldsymbol{\delta}^{(n)}$ is not directly learnt, but a shorter fixed length subsection $\boldsymbol{\delta}^{(k)}$ is learnt instead, where the signal $\boldsymbol{\delta}^{(n)}$ is formed by periodically repeating $\boldsymbol{\delta}^{(k)}$, with length k timeframes.

With knowledge of the underlying fixed length signal $\boldsymbol{\delta}^{(k)}$, the objective of the waveform adversarial attack can be expressed in terms of the noise signal $\boldsymbol{\delta}^{(n)}$, which is optimised to maximise the average predicted score across all S different speakers (candidates), as shown in Equation 5.12. The $\texttt{ASR}()$ function represents the mapping from an audio signal $o_{1:n}$ to a rich textual output $w^*_{1:L}$ (Figure 2.1).

$$\hat{\boldsymbol{\delta}} = \arg\max_{\boldsymbol{\delta}} \left\{ \sum_{s=1}^{S} \mathcal{G}\Big(\boldsymbol{o}^{(s)} \oplus \boldsymbol{\delta}, \texttt{ASR}(\boldsymbol{o}^{(s)} \oplus \boldsymbol{\delta}); \hat{\boldsymbol{\theta}}\Big) \right\} \tag{5.2}$$

5.2 Evolutionary Attack

The greatest challenge for waveform adversarial attacks is that the SLA systems are treated as a black-box. One approach to attacking a black-box Grader is to use an evolutionary attack [16]. The evolutionary approach attempts to approximate the optimal update direction of a gradient approach by treating each training cycle as a generation, with the *fittest* members passing their *genes* to the next generation. Each member i of the population is a noise subsection $\boldsymbol{\delta}^{(k)}$ and *fitness* is measured by ability of a member to satisfy two objective functions: the adversarial sample must maximise acoustic similarity (imperceptible attack) and the adversarial attack must maximise the output predicted score.

The members of the population are often initialised randomly from a Gaussian distribution. After initialisation, the population is passes through multiple stages cyclically (Figure 5.1). The fitness evaluation stage uses two distinct *fitness* functions f_1 and f_2 as a quantitative measure of the two objectives (above) to be satisfied. The first function f_1 gives the Euclidean distance between the original and manipulated audio signals in a transformed space $\mathcal{H}$ (Equation 5.3). Often Mel Frequency Cepstral Coefficients (MFCCs) [25] are the choice of transformed space for audio signals. Therefore, the objec-

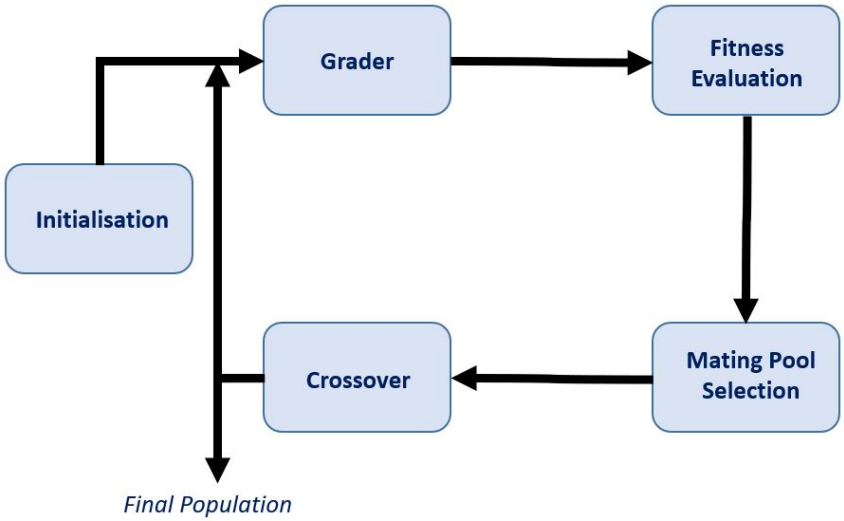

Figure 5.1: Evolutionary Approach to Black-box Adversarial Attack.

tive is to minimise f_1 for greatest acoustic similarity.

$$f_1(\boldsymbol{o} \oplus \boldsymbol{\delta}^{(n)}, \boldsymbol{o}) = ||\mathcal{H}(\boldsymbol{o} \oplus \boldsymbol{\delta}^{(n)}) - \mathcal{H}(\boldsymbol{o})||^2 \tag{5.3}$$

The second function f_2 is simply set to be the output score from the Grader (Equation 5.4). Hence, to maximise the output score, f_2 has to be maximised.

$$f_2(\boldsymbol{o} \oplus \boldsymbol{\delta}^{(n)}) = \mathcal{G}\left(\boldsymbol{o} \oplus \boldsymbol{\delta}^{(n)}, \texttt{ASR}(\boldsymbol{o} \oplus \boldsymbol{\delta}^{(n)}); \hat{\boldsymbol{\theta}}\right) \tag{5.4}$$

Both functions can be optimised simultaneously. The mating pool selection stage is concerned with selecting the pairs that will *mate* to create the next generation. The Multi-Objective Genetic Algorithm (MOGA) [12] offers three schemes to select appropriate pairs. This stage seeks to generate *children* from the parent pairs. A simple crossover strategy [16] is where each parent generates three children: one child similar to one parent each and the third acquires the parents characteristics in equal proportion. It is important to note that all operations are performed in the raw audio domain. To introduce robustness and maintain diversity in the population, probabilistic mutation is used, where a random value is added to every gene (single value in the noise signal) with a probability $\texttt{prob}_m$. The cycle is repeated till no significant improvement is seen in the fitness of the population, at which point a member of the population is randomly selected

as the *optimal* adversarial noise.

5.3 Guided Initialisation Gradient Attack

Due to the high dimensionality and complex nature of the input space, it is difficult to find a useful initialisation point for a black-box attack of a Grader using waveforms as the input. Therefore, it is natural to consider a separate simpler model that can mimic the performance of the black-box model. As the simpler model can be built in-house, the adversary has access to its internal workings and can therefore employ a white-box attack. Once the optimal adversarial waveform is obtained for the simpler model, this can be used as a useful initialisation point for the evolutionary attack on the more complex black-box Grader.

When designing the simpler model, it is useful to use a construct that contains a pipeline of differentiable stages that allow gradient based approaches to be used to learn the adversarial noise vector. Gradient based approaches are superior to evolutionary approaches because the gradient gives the optimal search direction for optimization. Therefore, as most Graders use a non-differentiable ASR stage, the gradient based attack has to be conducted in a manner that freezes the ASR output, exploiting the assumption that the adversarial attack causes such small perturbations there is no impact on the ASR output. Hence, the frozen ASR output means Equation 5.12 reduces to Equation 5.5.

$$\hat{\boldsymbol{\delta}} = \arg\max_{\boldsymbol{\delta}} \left\{ \sum_{s=1}^{S} \mathcal{G}\Big(\boldsymbol{o}^{(s)} \oplus \boldsymbol{\delta}, \texttt{ASR}(\boldsymbol{o}^{(s)}); \hat{\boldsymbol{\theta}}\Big) \right\} \tag{5.5}$$

This work uses a shallow phone distance Grader trained on MFCC features [19] as the simpler model of choice. The transformation of a segment of a waveform $o_{1:n} = \boldsymbol{o}$ into a MFCC vector $\mathbf{z}$ can be separated into three stages, where only the first stage is a linear transformation, taking inputs to the linear spectral space. The stages are shown in Equation 5.6.

$$\mathbf{o} \xrightarrow[\text{Filter, } \mathcal{L}()]{} \breve{\mathbf{z}} \xrightarrow[\log()]{} \tilde{\mathbf{z}} \xrightarrow[\text{DCT, } \mathbf{C}]{} \mathbf{z} \tag{5.6}$$

$$\uparrow$$

Linear Spectral Space

To reduce the search space for the adversarial attack, the universal adversarial noise $\boldsymbol{\delta}^{(\text{spec})}$ can be found in the linear spectral space. It is necessary that the spectral space is linear, so that the universal noise in the spectral space can pass though an inverse filter $\mathcal{L}^{-1}$ to obtain a universal adversarial noise in the audio space (Equation 5.7).

$$\boldsymbol{\delta} = \mathcal{L}^{-1}\left(\boldsymbol{\delta}^{(\text{spec})}\right) \tag{5.7}$$

The model uses shallow phone distance features as its input. For each speaker s and each of the 47 phones p, the MFCC vector is modelled as a simple multi-variate Gaussian distribution. The covariance matrix is of little interest. Only the mean vector $\mu^{(sp)}$ is of interest. Having computed all the phone mean vectors (in the MFCC space) for a particular speaker, the phone distance features can be computed, as shown in Equation 5.8.

$$m_{ij}^{(s)} = \mathcal{D}(\boldsymbol{\mu}^{(si)}, \boldsymbol{\mu}^{(sj)}) \tag{5.8}$$

Using the 47^2 phone distances, a phone matrix $\mathbf{M}$ can be constructed and this is used to predict the score $y^{(s)}$ by the model $\mathcal{G}()$, as given in Equation 5.9, where $\hat{\boldsymbol{\theta}}$ are the fixed model parameters optimised on a separate training dataset (using MSE as in Equation 3.4). It is useful to note that $\mathbf{M}$ essentially is a fixed length representation $\boldsymbol{h}$ of the variable length input $o_{1:n}$.

$$y^{(s)} = \mathcal{G}(\mathbf{M}^{(s)}; \hat{\boldsymbol{\theta}}) \tag{5.9}$$

The objective of the universal adversarial attack is to find a noise vector $\hat{\boldsymbol{\delta}}^{(\text{spec})}$, to be added

in the linear spectral space, independent of the speaker, to maximise average predicted score across all the speakers. Equation 5.10 shows how the noise vector can be added to a particular phone mean in the linear spectral space.

$$\boldsymbol{\mu}_a^{(sp)} = \mathbf{C} \log \left(\exp \left(\mathbf{C^{-1}} \boldsymbol{\mu}^{(\mathbf{sp})} \right) + \hat{\boldsymbol{\delta}}^{(\text{spec})} \right) \tag{5.10}$$

Equation 5.11 then shows how the adversarially attacked phone mean vectors can be used to compute the adversarially attacked phone-distance matrix $\mathbf{M}_a^{(s)}$ for each speaker.

$$\mathbf{M}_a^{(s)} = \mathcal{M} \left(\boldsymbol{\mu}^{(s1)}, \ldots, \boldsymbol{\mu}^{(sP)}, \hat{\boldsymbol{\delta}}^{(\text{spec})} \right) \tag{5.11}$$

Therefore, the overall objective is summarised by Equation 5.12.

$$\hat{\boldsymbol{\delta}}^{(\text{spec})} = \underset{\boldsymbol{\delta}^{(\text{spec})}}{\arg\max} \left\{ \sum_{s=1}^{S} \mathcal{G} \Big(\mathcal{M} \left(\boldsymbol{\mu}^{(s1)}, \ldots, \boldsymbol{\mu}^{(sP)}, \boldsymbol{\delta}^{(\text{spec})} \right) ; \hat{\boldsymbol{\theta}} \Big) \right\} \tag{5.12}$$

Chapter 6

Experimental Results

6.1 Experimental Setup

6.1.1 Datasets

Experiments were run for multi-level prompt-response free speaking tests i.e. candidates from a range of proficiency levels provide open responses to prompted questions. Based on this audio input (for which the transcriptions are unknown) the assessment systems must predict a score of 0-6 corresponding to the 6 CEFR [7] grades. Particularly for practice scenarios, it is useful for the learner to receive a separate score for each section of the test. The dataset consists of 21 responses divided into 5 sections (A-E). For this work a single section (C) of the Linguaskill-Business (L-Bus) test [4] is considered in greater detail, where candidates can talk for up to 60 seconds on a prompted topic. The training and test data consists of non-native English spoken by candidates from 6 L1s (first language). A held-out evaluation set of 202 speakers, approximately balanced for L1 and across the CEFR grades was used for testing. For this data the ASR system had an average word error rate of 19.5%. Reference scores were provided by expert graders. The Graders were trained on a set of ∼900 speakers from the same set of L1s, using operational grader reference scores.

For the purpose of comparison, Graders trained on an alternative dataset, Linguaskill-

General (L-Gen) are also considered. This structure of L-Gen data is similar to L-Bus, where the differences are due to the style of prompts. Average response lengths for both datasets are approximately 80 words.

6.1.2 Assessment Metrics

The performance of the SLA Graders is measured against four metrics: Pearson Correlation Coefficient (PCC); Root Mean Squared Error (RMSE); percentage within half a grade (<0.5) and percentage within one grade (<1.0), where the reference candidate CEFR grades are from expert human examiners.

The PCC indicates the linear correlation between the predicted and reference scores, where +1 is a perfect positive correlation and 0 is no correlation. The RMSE is simply the average (across all candidate data points) of the error (between prediction and reference score) squared. This is the metric used in the loss function to train all the Graders. The percentage within half and full grade metrics indicate what percentage of candidates are likely to be classified correctly by the Grader.

The adversarial attack impacts are best measured by the increase in the average (across candidates) predicted score. To assess the success of the adversarial attack detection processes, precision-recall curves are used. For the binary classification task of identifying an input as adversarially attacked (or not), the precision is the ratio of true-positive (correctly classified as adversarially attacked) over the total identified as adversarially attacked (Equation 6.1). Recall, however, gives the ratio of the true-positive to the total actual number of adversarially attacked inputs (Equation 6.2). For a given classification boundary threshold β, the precision and recall values can be computed. The precision-recall curves can then be produced by varying β.

$$\text{Precision} = \frac{\texttt{True Positive}}{\texttt{True Positive} + \texttt{False Positive}} \tag{6.1}$$

$$\text{Recall} = \frac{\texttt{True Positive}}{\texttt{True Positive} + \texttt{False Negative}} \tag{6.2}$$

A single point summary of precision recall-curves can be obtained using a F_k score (Equation 6.3), where k weighs the importance of precision over recall. This works uses a $F_{0.5}$ score.

$$F_k = \frac{(1+k^2) \times \text{Precision} \times \text{Recall}}{(k^2 \times \text{Precision}) + \text{Recall}} \tag{6.3}$$

6.2 Experiments and Discussion

6.2.1 Baseline

The first set of experiments determine the baseline performance of different Graders, ranging from feature-based GP Graders (section 3.1.1) to neural Graders using either text, $\text{Neur}_{\texttt{txt}}$[1] (section 3.2.1); pronunciation, $\text{Neur}_{\texttt{pron}}$ [20]; rhythm, $\text{Neur}_{\texttt{rytm}}$ [21] and text+duration+confidence, $\text{Neur}_{\texttt{txt+dc}}$ (section 3.2.2). For the feature-based Graders, two types of GP Graders are considered: a GP system using all audio and text features, GP_{all} [43] and a text-based GP system, GP_{txt} using only seven features based on the ASR transcript. Different linear combinations of these graders are considered to establish which Graders contribute most to the overall performance. All neural Grader ensembles are obtained by using 10 different seed initialisations during training and the resulting 10 ensembles' outputs simply averaged to get the ensemble prediction.

Table 6.1 presents the performance of text-based, audio text-based, rhythm-based and pronunciation based neural systems, as well as versions of a feature-based GP system to the overall task of grading speakers based on responses to 21 questions in 5 sections (A-E). When combining the neural Graders, $\text{Neur}_{\text{txt+dc}}$ is not included as its structure is extremely similar to Neur_{txt} with marginal performance improvements. Combining the neural systems in a linearly optimal manner yields the best results - a Pearson Correlation Coefficient (PCC) of 0.884 is achieved, where the biggest contributor to the output prediction is the Neur_{txt} system. For the feature based Graders, GP_{txt} performs nearly as well as GP_{all}, despite using only the text features. Hence, the results of Table 6.1

[1]Model Available at: https://github.com/rainavyas/NeurTxtGrader

motivate attacking the text-based systems when seeking to obtain an adversarial sample for a complete (combination of all systems) spoken language assessment system.

System	PCC	RMSE	<0.5	<1.0
Neur_{txt}	0.878	0.587	66.8	91.4
$\text{Neur}_{\text{txt+dc}}$	0.883	0.564	70.3	91.9
$\text{Neur}_{\text{pron}}$	0.819	0.699	54.1	85.5
$\text{Neur}_{\text{rytm}}$	0.815	0.697	55.9	86.4
$\text{Neur}_{\{\text{txt}\oplus\text{pron}\oplus\text{rytm}\}}$ $\alpha = [0.83, 0.04, 0.13]$	0.884	0.581	67.0	90.5
GP_{txt}	0.855	0.643	60.4	87.7
GP_{all}	0.881	0.606	60.5	91.4

Table 6.1: Performance of different ensemble systems trained on all sections (A-E) of L-Bus. α gives the linear combination coefficients.

Grader	Score	PCC	RMSE	<0.5	<1.0
GP_{txt}	3.88	0.749	0.786	54.0	80.7
Neur_{txt}	$\underset{\pm 0.14}{3.49}$	$\underset{\pm 0.01}{0.744}$	$\underset{\pm 0.06}{0.818}$	$\underset{\pm 6.55}{48.9}$	$\underset{\pm 2.86}{79.4}$
-ensemble	3.49	0.749	0.727	59.9	83.2
$\text{GP}_{\text{txt}}\oplus\text{Neur}_{\text{txt}}$	3.69	0.774	0.678	61.4	83.7

Table 6.2: Baseline performance (on section C of L-Bus) of the text-based feature and deep neural Graders. ± indicates the standard deviation.

Table 6.2 shows the baseline performance of the deep neural model and the feature-based GP model on a single part of the examination. The PCCs for each system are very consistent. A wider spread in predictions can be observed for the GP Grader with a higher RMSE and lower percentages of scores within 0.5 (<0.5) and 1.0 (<1.0) of the reference scores i.e. within half and one grade. The average score for the GP Grader (3.88) is also the most offset to the reference average of 3.54, with the neural-ensemble Grader having the closest match (3.49). Combining the GP and neural-ensemble Graders, GP⊕Neural, yielded the best overall performance, suggesting that these systems are complementary.

6.2.2 Text Adversarial Attack

To construct the adversarial attack only one of the members of the ensemble of the neural Grader, Neur_{txt} was used. It was found that this attack transferred to all members of the ensemble, as expected from the relatively small standard deviation (Table 6.2). As

shown in Equation 4.4, a greedy search is performed over a subset vocabulary. The subset vocabulary (experimentation revealed 100 was an adequate size for the subset vocabulary) is generated from a greedy search over the ASR vocabulary on a substitute system: a text neural Grader using word2vec [30] at the word embedding stage. The attack was used to obtain an adversarial phrase of increasing lengths k, until the output predicted score saturated. This adversarial phrase was found for the neural model trained on section C of the L-Bus dataset. Figure 6.1 shows that saturation occurs at $k = 20$, where on average the output score is increased by 2.2 (on a scale of 0-6). Figure 6.1 also shows how well the attack transfers to the same model trained on a different dataset (L-Gen) and a different section (D). The increase of more than a grade in 20 words for both these models suggests that this attack phrase has some universal properties that transfer well.

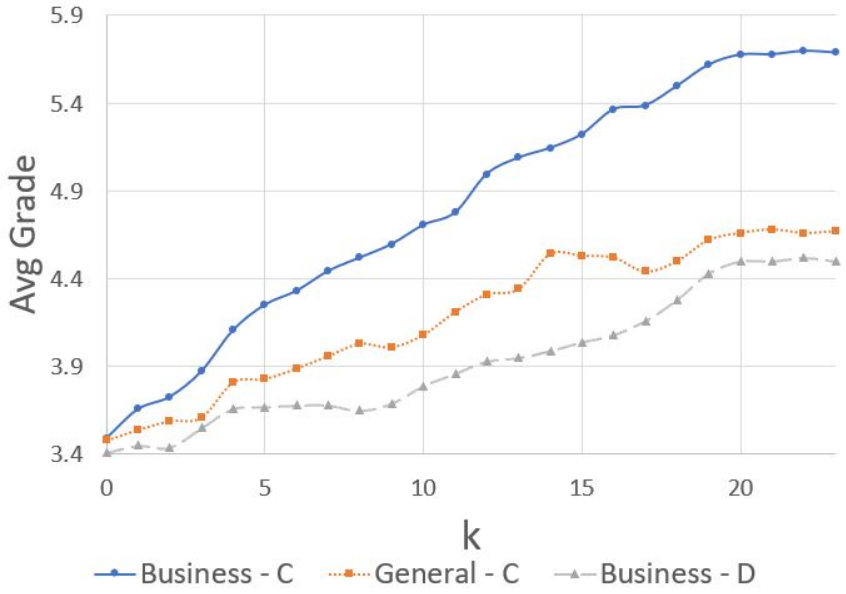

Figure 6.1: Transferability of k-word attack phrase found for the neural model trained on L-Bus, section C

This work carries out a more in-depth analysis of a shorter (more realistic) adversarial phrase length of 6 words. A greedy search was used on the neural Grader to find an adversarial phrase of 6 words in length[2]. Similarly, a 6 word adversarial phrase was determined for the GP Grader[3]. Table 6.3 shows that both the GP and Neural adversarial attacks' word sequences, targeted at the corresponding assessment system, significantly increased the score. For the neural assessment system, however, the increase after adding six words was over twice that of the GP based system. The impact of the adversarial

[2]Neural adversarial attack phrase: OFFENSIVELY OBESE ASTRONAUTS AMAZINGLY CRITICIZES AMAZINGLY.
[3]GP adversarial attack phrase: QUITE PRETTY INDIVIDUALLY INTO PHYSICALLY QUITE.

attack phrases is also reported for a DNN-based feature Grader, DNN_{txt} [29] (described in section 3.1.2), using the same 7 text features as GP_{txt}. Interestingly the neural adversarial attack yielded only a small increase in the average score in the latter for both the GP and DNN Graders. This indicates that the transferability of the attack from the neural assessment system to the feature-based systems is small.

Grader (+adv)	Score	PCC	RMSE	<0.5	<1.0
GP_{txt}	3.88	0.749	0.786	54.0	80.7
+ GP-adv	4.27	0.744	1.037	30.7	68.3
+ NEUR-adv	4.02	0.749	0.863	49.0	76.7
DNN_{txt}	3.70	0.750	0.732	56.9	83.7
+ GP-adv	4.23	0.691	1.038	31.7	72.3
+ NEUR-adv	3.84	0.750	0.772	53.0	82.7
Neur_{txt}	3.49	0.749	0.727	59.9	83.2
+ GP-adv	3.54	0.753	0.702	58.4	83.2
+ NEUR-adv	4.33	0.700	1.110	27.2	62.9

Table 6.3: Impact of the 6 word Neural adversarial attack NEUR-adv or GP adversarial attack GP-adv on different Graders.

Table 6.3 also presents the impact of the adversarial attacks on standard performance metrics. For these experiments the GP and ensembles of DNNs and neural systems were used. As expected, the largest score gain, and associated impact on performance, is from attacking the neural assessment system. The percentage of candidates with scores within one grade-level decreased from 83.2% to 62.9% as the scores were pushed up across the grade range. It is interesting that the GP attack transfers well to the DNN-based system, whereas the neural attack only yields a small change in performance.

As the penalty for incorrectly detecting an adversarial attack is high, precision is more important than recall for this task. Thus $F_{0.5}$ is used to give a single point summary of the system performance. Figure 6.2 shows the precision and recall curves and the optimal $F_{0.5}$ value for the four detection schemes. As expected, the performance of the off-topic response detection is the worst, as the short adversarial attack is appended to a valid, on-topic response. Although ensemble diversity, and perplexity, show reasonable performance, the best performing detection scheme is GP shift.

The superior performance of the GP shift detection scheme is further verified by Table 6.4, where a $F_{0.5}$ score of 0.955 can be achieved for a 20 word adversarial attack. It is interesting to note that both topic relevance and ensemble diversity $F_{0.5}$ scores have worsened from

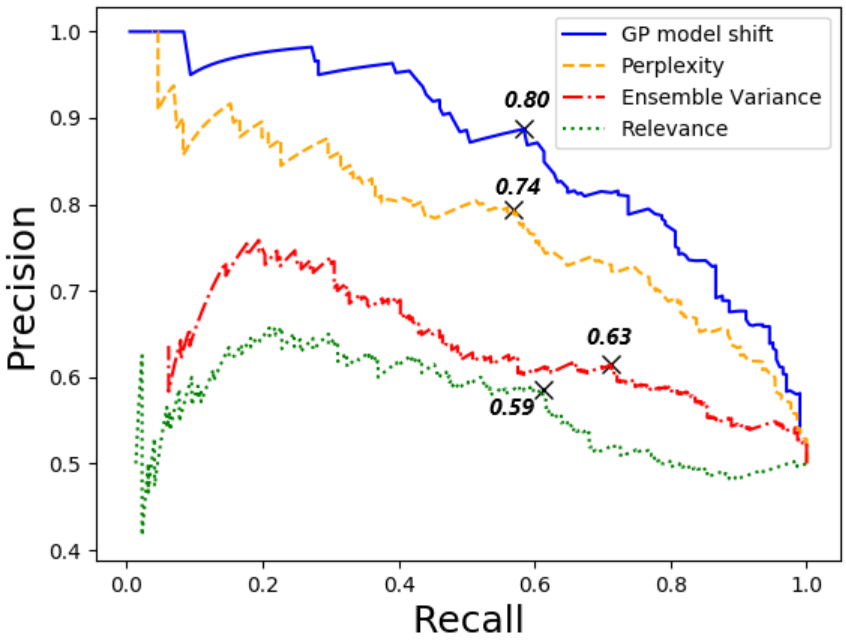

Figure 6.2: Precision-Recall curves for different detection approaches for the Neural Grader with 6 NEUR-adv words

the 6 word adversarial attack. Topic relevance may suffer due to inherent length biases in the model used. To understand the failure of ensemble diversity for the longer phrase, it is useful to refer to the gradients of the curves in Figure 6.1. As the curves saturate, it is only expected that the absolute variation in ensemble prediction decreases, rendering ensemble diversity a meaningless detection scheme for longer adversarial phrases.

Detection Scheme	$F_{0.5}$	Precision	Recall
GP Shift	0.955	0.983	0.856
Perplexity	0.906	0.935	0.806
Topic Relevance	0.556	0.500	1.0
Ensemble Diversity	0.558	0.502	1.0

Table 6.4: Performance of different detection schemes on the 20 NEUR-adv adversarial words for the Neural grader.

One of the features that is used in th featured-based GP Grader is the set of five grade-dependent language model scores. These have the same form as the perplexity detection system and thus the GP Grader adversarial attack GP-adv is expected to be harder to detect using perplexity. This is the case as the $F_{0.5}$ score drops from 0.74 to 0.634.

The current adversarial attack is based on appending the adversarial attack to the end of the response. In order to assess the transferability of this attack to different positions, the same 6 word phrase was appended to the beginning and to the middle of the original

response for the NEUR-adv attack. This yielded average score values of 4.08 and 4.13 respectively, compared to appending to the end of 4.33. Thus a large grade increase of greater than half a grade was possible even for these sub-optimal attacks for the neural Grader.

Grader (+adv)	Score	PCC	RMSE	<0.5	<1.0
Neur_{txt}	3.49	0.749	0.727	59.9	83.2
+ NEUR-adv	4.33	0.700	1.110	27.2	62.9
+ GP-Det-adv	4.15	0.715	0.975	35.1	69.8
+ PERP-Det-adv	4.22	0.711	1.020	32.7	66.3

Table 6.5: Detection evasion attacks on the neural Grader.

It is possible to attack a system with knowledge of the defence mechanism. Attacks were generated independently to evade the GP Shift (GP-Det-adv) and perplexity (PERP-Det-adv) detection processes by ensuring that the final GP shift and perplexity were less than the corresponding thresholds used to generate the $F_{0.5}$ scores in Figure 6.2. These attacks (Table 6.5) yield lower increases in the average score (4.15 and 4.22 correspondingly) than the unconstrained NEUR-adv attack. It is of course possible to operate at lower thresholds for the detection evasion attacks, or combine the detection approaches, to further reduce the impact of adversarial attacks.

6.2.3 Waveform Adversarial Attack

Finally, waveform adversarial attacks are designed to be carried out on a deep-phone neural Grader [20]. However, to perform the initialisation of the evolutionary attack, a white-box gradient based attack is carried out on a simpler shallow-phone neural Grader (described in section 5.3).

An initial, naive attempt was made on attacking a black-box deep-phone neural Grader [20] using the default evolutionary approach (section 5.2). However, experiments revealed that the waveform search space was too large and complex to discover any adversarial attacks that had a significant impact on the predicted score. Therefore, as dictated by Equation 5.12, a white-box gradient based attack was carried out on a simpler shallow phone distance model (described in section 5.3), with the aim of transferring this attack to be the initialisation point for the deep phone distance model.

It is important to note that initial experiments yielded a negative adversarial noise in the

spectral space, which has no meaning. Therefore, the training algorithm was adjusted to exponentiate the noise vector to ensure it remained positive. Table 6.6 reveals that despite training for 100 epochs, the average score is increased by less than 0.1. This suggests that this structure of a phone distance model is inherently robust against this form of adversarial attack. Hence, it is only natural to conclude that the black-box, deep-phone Grader, although likely to be more susceptible to attacks due to its *deep* structure, is also relatively robust to adversarial attacks that attempt to use universal noise vectors to modify the inputs.

Epoch	Increase in Avg. Score
0	0.00
1	0.04
2	0.07
10	0.08
20	0.07
50	0.08
100	0.08

Table 6.6: Training of 24-dim spectral noise vector in gradient-based waveform attack of shallow phone-distance Grader.

Chapter 7

Conclusions and Future Work

Driven in part by the performance improvements that have come from deep learning based approaches, automated Spoken Language Assessment (SLA) systems have seen an increase in their popularity and demand. This work has considered in specific SLA systems for a multi-level prompt-response free speaking test, Linguaskill-Business.

The challenge for SLA systems stems from the variable length nature of a candidate's audio recording. This challenge is overcome through the use of traditional feature-based Graders and design of more state of the art deep learning (neural) based Graders. Experiments reveal that for an overall grading system, whether it is feature based or neural, it is the text-based Graders that contribute significantly to the strong performance of the SLA systems. Therefore, this work focuses on text based Graders predominantly.

In contrast to the feature-based Graders, it is known that neural Graders are susceptible to adversarial attacks and thus it is important to investigate the robustness of these Graders. Specifically for text-based neural Graders, a simple, universal black-box adversarial attack is examined. The aim is to generate a single, universal phrase that when uttered at the end of a valid response to a prompt will improve the performance of any candidate. The work shows that spoken language assessment systems are indeed susceptible to these universal attacks. Even a short six word phrase can yield nearly a one point increase in the average grade for the test speakers. The impact of adversarial attacks for these neural systems is compared to the more traditional feature-based systems, which as expected

are found to be less sensitive to adversarial attacks. Four defence mechanisms, including the standard perplexity score, as well as assessment specific schemes, are also discussed. These can accurately detect attacks, but can also be used as part of the adversarial attack generation, if the form of detection is known.

To account for more a complete SLA system that uses both textual information and the raw audio signal, a waveform attack is considered for a deep phone distance model, treated as a non-differentiable black-box system. Using an evolutionary approach to learn the universal adversarial noise yields an insignificant increase in the average predicted score. Therefore, a shallow phone distance model is chosen as a simple model to be subject to a white-box attack, with the intention of using this to initialise the attack for the deep phone distance model. However, surprisingly, this attack yields little improvements, suggesting that phone distance models are relatively robust to such universal waveform adversarial attacks.

Future work is useful for identifying other non-text based deep neural models, such as rhythm based models, that are susceptible to waveform adversarial attacks. If such systems are susceptible to these attacks, it is necessary to determine appropriate detection methods. Initials experiments can explore the use of qualitative surveys with human examiners to detect the manipulated input audio signals. Other detection processes can include the use of the ASR output word confidence scores to indicate adversarial attacks or seek to learn a metric that measures the *naturalness* of audio signals, to mimic the use of perplexity in the text attacks. Finally, adversarial training can be explored as a method to preemptively increase the robustness of systems.

Bibliography

[1] Alzantot, M., Sharma, Y., Elgohary, A., Ho, B.-J., Srivastava, M., and Chang, K.-W. Generating natural language adversarial examples. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (Brussels, Belgium, Oct.-Nov. 2018), Association for Computational Linguistics, pp. 2890–2896.

[2] Behjati, M., Moosavi-Dezfooli, S., Baghshah, M. S., and Frossard, P. Universal adversarial attacks on text classifiers. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (May 2019), pp. 7345–7349.

[3] Carlini, N., and Wagner, D. A. Towards evaluating the robustness of neural networks. *CoRR abs/1608.04644* (2016).

[4] Chambers, L., and Ingham, K. The BULATS online speaking test. *Research Notes 43* (2011), 21–25.

[5] Chen, X., Liu, X., Gales, M., and Woodland, P. CUED-RNNLM – an open-source toolkit for efficient training and evaluation of recurrent neural network language models. In *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2015).

[6] Cheng, S., Dong, Y., Pang, T., Su, H., and Zhu, J. Improving black-box adversarial attacks with a transfer-based prior. *CoRR abs/1906.06919* (2019).

[7] Council of Europe. *Common European Framework of Reference for Languages: Learning, Teaching, Assessment*. Cambridge University Press, 2001.

[8] Cruttenden, A., and Gimson, A. C. *Gimsons pronunciation of English*. Hodder Education, 2008.

[9] Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proc. of 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)* (2019), pp. 4171–4186.

[10] Dong, Y., Pang, T., Su, H., and Zhu, J. Evading defenses to transferable adversarial examples by translation-invariant attacks. *CoRR abs/1904.02884* (2019).

[11] Ebrahimi, J., Rao, A., Lowd, D., and Dou, D. Hotflip: White-box adversarial examples for NLP. *CoRR abs/1712.06751* (2017).

[12] Fonseca, C., and Fleming, P. Multiobjective genetic algorithms. *Genetic Algorithms in Engineering Systems*, 63–78.

[13] Hinton, G., Deng, L., Yu, D., Dahl, G., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Kingsbury, B., and Sainath, T. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine 29* (November 2012), 82–97.

[14] Ilyas, A., Engstrom, L., and Madry, A. Prior convictions: Black-box adversarial attacks with bandits and priors, 2018.

[15] Inkawhich, N., Wen, W. J., Li, H., and Chen, Y. Feature space perturbations yield more transferable adversarial examples. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), 7059–7067.

[16] Khare, S., Aralikatte, R., and Mani, S. Adversarial black-box attacks for automatic speech recognition systems using multi-objective genetic optimization. *CoRR abs/1811.01312* (2018).

[17] Kurakin, A., Goodfellow, I. J., and Bengio, S. Adversarial examples in the physical world. In *Proc. of International Conference on Learning Representations (ICLR)* (2017).

[18] Kurakin, A., Goodfellow, I. J., and Bengio, S. Adversarial machine learning at scale. In *Proc. of International Conference on Learning Representations (ICLR)* (2017).

[19] Kyriakopoulos, K., Gales, M., and Knill, K. Automatic characterisation of the pronunciation of non-native english speakers using phone distance features. pp. 59–64.

[20] Kyriakopoulos, K., Knill, K., and Gales, M. J. F. A deep learning approach to assessing non-native pronunciation of english using phone distances. pp. 1626–1630.

[21] Kyriakopoulos, K., Knill, K. M., and Gales, M. J. F. A deep learning approach to automatic characterisation of rhythm in non-native English speech. pp. 1836–1840.

[22] Lei, Q., Wu, L., Chen, P.-Y., Dimakis, A., Dhillon, I. S., and Witbrock, M. J. Discrete adversarial attacks and submodular optimization with applications to text classification. In *Proceedings of Machine Learning and Systems 2019* (2019), pp. 146–165.

[23] Lipton, Z. C. A critical review of recurrent neural networks for sequence learning. *CoRR abs/1506.00019* (2015).

[24] Liu, Y., Chen, X., Liu, C., and Song, D. Delving into transferable adversarial examples and black-box attacks. *CoRR abs/1611.02770* (2016).

[25] M., L. N., and Kopparapu, S. K. Choice of mel filter bank in computing MFCC of a resampled speech. *CoRR abs/1410.6903* (2014).

[26] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *ArXiv abs/1706.06083* (2017).

[27] Malinin, A., and Gales, M. J. Reverse KL-divergence training of prior networks: Improved uncertainty and adversarial robustness. In *Advances in Neural Information Processing Systems 32* (2019), pp. 14520–14531.

[28] Malinin, A., Knill, K., and Gales, M. J. F. A hierarchical attention based model for off-topic spontaneous spoken response detection. In *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)* (Dec 2017), pp. 397–403.

[29] Malinin, A., Ragni, A., Knill, K., and Gales, M. J. F. Incorporating uncertainty into deep learning for spoken language assessment. In *Proc. of 55th Annual Meeting of the Association for Computational Linguistics (ACL)* (2017), pp. 45–50.

[30] Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient Estimation of Word Representations in Vector Space, 2013. arXiv:1301.3781.

[31] Moosavi-Dezfooli, S., Fawzi, A., Fawzi, O., and Frossard, P. Universal adversarial perturbations. *CoRR abs/1610.08401* (2016).

[32] Pang, T., Du, C., and Zhu, J. Robust deep learning via reverse cross-entropy training and thresholding test. *CoRR abs/1706.00633* (2017).

[33] Papernot, N., McDaniel, P. D., Goodfellow, I. J., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against deep learning systems using adversarial examples. *CoRR abs/1602.02697* (2016).

[34] Papernot, N., McDaniel, P. D., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. The limitations of deep learning in adversarial settings. *2016 IEEE European Symposium on Security and Privacy (EuroS&P)* (2015), 372–387.

[35] Raina, V., Gales, M. J., and Knill, K. M. Complementary systems for off-topic spoken response detection. In *to be included Proc. 15th Workshop on Innovative Use of NLP for Building Educational Applications* (2020).

[36] Sherstinsky, A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *CoRR abs/1808.03314* (2018).

[37] Strauss, T., Hanselmann, M., Junginger, A., and Ulmer, H. Ensemble methods as a defense to adversarial perturbations against deep neural networks, 2017.

[38] Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In *Proc. of 27th International Conference on Neural Information Processing Systems* (Cambridge, MA, USA, 2014), MIT Press, pp. 3104–3112.

[39] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks.

[40] van Dalen, R., Knill, K., and Gales, M. Automatically grading learners' English using a gaussian process. In *Proc. of ISCA Workshop on Speech and Language Technology for Education (SLaTE)* (2015).

[41] van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. W., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. In *Proc. of Speech Synthesis Workshop (SSW)* (2016).

[42] Wang, W., Wang, L., Wang, R., Wang, Z., and Ye, A. Towards a robust deep neural network in texts: A survey, 2019.

[43] Wang, Y., Gales, M., Knill, K. M., Kyriakopoulos, K., Malinin, A., van Dalen, R., and Rashid, M. Towards automatic assessment of spontaneous spoken english. *Speech Communication 104* (2018), 47–56.

[44] Xie, C., Zhang, Z., Wang, J., Zhou, Y., Ren, Z., and Yuille, A. L. Improving transferability of adversarial examples with input diversity. *CoRR abs/1803.06978* (2018).

[45] Xu, H., Dong, M., Zhu, D., Kotov, A., Carcone, A. I., and Naar-King, S. Text classification with topic-based word embedding and convolutional neural networks. In *Proc. of 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics* (New York, NY, USA, 2016), BCB '16, ACM, pp. 88–97.

[46] Yan, Z., Guo, Y., and Zhang, C. Subspace attack: Exploiting promising subspaces for query-efficient black-box attacks. *CoRR abs/1906.04392* (2019).

[47] Yang, P., Chen, J., Hsieh, C., Wang, J., and Jordan, M. I. Greedy attack and gumbel attack: Generating adversarial examples for discrete data. *CoRR abs/1805.12316* (2018).

Appendix A

A.1 Risk Assessment

This project was entirely computational. The risk is excessive computer use and the adoption of a detrimental posture. The first risk was minimised through a strict policy of regular breaks. The second risk was reduced by obeying standard seating posture guidelines. Further, the workstation was setup to conform with the Display Screen Equipment regulations.

A.2 Log Book

In this project, an online log book was maintained through weekly reports submitted at http://mi.eng.cam.ac.uk/raven/babel/UGrad/.