2019-04-25

Training Noise-Robust Spoken Phrase Detectors with Scarce and Private Data: An Application to Classroom Observation Videos

Brian Matthew Zylich
Worcester Polytechnic Institute

Training Noise-Robust Spoken Phrase Detectors with Scarce and Private Data: An Application to Classroom Observation Videos

by

Brian Zylich

A Thesis

Submitted to the Faculty

of the

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Master of Science

in

Computer Science

by

April 2019

APPROVED:

Professor Jacob R. Whitehill, Master Thesis Advisor

Professor Gillian Smith, Thesis Reader

Professor Craig E. Wills, Head of Department

Abstract

We explore how to automatically detect specific phrases in audio from noisy, multi-speaker videos using deep neural networks. Specifically, we focus on classroom observation videos that contain a few adult teachers and several small children (< 5 years old). At any point in these videos, multiple people may be talking, shouting, crying, or singing simultaneously. Our goal is to recognize *polite speech* phrases such as "Good job", "Thank you", "Please", and "You're welcome", as the occurrence of such speech is one of the behavioral markers used in classroom observation coding via the Classroom Assessment Scoring System (CLASS) protocol [1]. Commercial speech recognition services such as Google Cloud Speech are impractical because of data privacy concerns. Therefore, we train and test our own custom models using a combination of publicly available classroom videos from YouTube [2], as well as a private dataset of real classroom observation videos collected by our colleagues at the University of Virginia. We also crowdsource an additional 1152 recordings of polite speech phrases to augment our training dataset. Our contributions are the following: **(1)** we design a crowdsourcing task for efficiently labeling speech events in classroom videos, **(2)** we develop a neural network-based architecture for speech recognition, robust to noise and overlapping speech, and **(3)** we explore methods to synthesize new and authentic audio data, both to increase the training set size and reduce the class imbalance. Finally, using our trained polite speech detector, **(4)** we investigate the relationship between polite speech and CLASS scores and enable teachers to visualize their use of polite language.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this thesis, we explore speech recognition in noisy, crowded environments. Speech recognition is a field that has been extensively researched. However, there are still settings in which existing speech recognition solutions struggle to accurately detect phrases or predict exactly what is being said. In some cases, there is too much background noise or overlapping speech, and the model cannot separate out the speech from the noise. Furthermore, sometimes commercial speech solutions cannot be used due to data privacy concerns. One particular type of video that fits into both of these categories is the classroom observation video, i.e., videos that are recorded inside a school classroom both for educational research and teachers professional development purposes. Classroom observation videos, specifically in preschool classrooms, are the application focus of this thesis.

In videos recorded in preschool classrooms, there are often a few teachers and several young children, each of whom may be speaking, yelling, crying, or singing at any point during the classroom video (Fig. 1.1). These attributes make classroom videos a difficult setting for speech recognition. However, speech recognition of certain key phrases in classroom videos could potentially be used to provide teachers with various forms of feedback that might help them learn from their interactions with students and improve their teaching in the future.

For feedback based on classroom observation videos to be useful to teachers or educational researchers, it is important that there be a system in place to ensure ratings are objective and consistent between raters. Without consistency and objectivity, such ratings could not be used as measures of growth or change over time, as two raters might interpret a given period of classroom interaction very differently without well-defined criteria and behavioral indicators. There are various classroom observation protocols for providing teachers with feedback, and one of the most commonly used is the Classroom Assessment Scoring System (CLASS) [1]. CLASS spans 10 different dimensions (depending on the age group of the classroom), and each dimension is scored on a scale of 1-7 based on the presence (or absence) of specified behavioral markers. For toddler classrooms, CLASS has eight dimensions that are categorized as either Emotional and Behavioral Support or Engaged Support for Learning (see Fig. 1.2). In this research, we focus on detecting *polite speech* phrases in classroom videos because polite speech is a behavioral

Figure 1.1: An example of a preschool classroom observation video where several toddlers are making noise while their teacher reads them a story.

marker used in multiple dimensions within the CLASS protocol, and our collaborators believe that the interactions taking place when *polite speech* is used may be insightful for teachers and researchers.

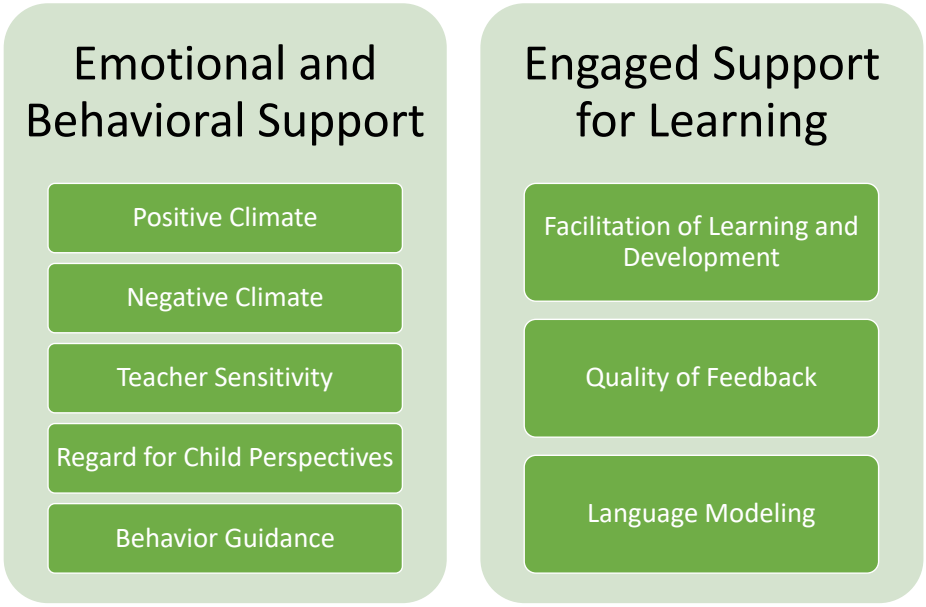


Figure 1.2: The Classroom Assessment Scoring System (CLASS) for toddler classrooms features eight dimensions in two main categories [1]. Polite speech is a behavioral marker used in CLASS coding for several of these dimensions.

Providing teachers with feedback across the dimensions covered by CLASS is important because students' cognitive and emotional development is dependent on the emotional and behavioral support that they receive in the classroom [1, 4]. Typically, the CLASS is used to assess a teacher's quality of instruction through in-person observation or manual review of video taken in the classroom. This observation and evaluation is time-intensive and requires evaluators who have completed extensive training with the CLASS. Automated evaluation of classroom videos would make feedback more readily available to teachers, allowing them to see what they are doing right and what changes they can make to improve their teaching. Previous work has investigated the automated labeling of

classroom videos according to CLASS dimensions using smile-detection and simple audio features [3].

Our research focused on using the audio component of classroom videos to construct complementary higher-level features, namely instances of polite speech, for the prediction of CLASS score. We develop our own deep neural network-based speech recognition system that is robust to noisy, multi-speaker classroom environments and compare the performance of our system with Google Cloud Speech to demonstrate the task's difficulty. For training and testing our detector, we use a combination of public available YouTube videos, crowdsourced speech recordings, and real classroom observation videos collected by our collaborators at the University of Virginia. These datasets are described in greater detail in Chapter 3; however, in general, these datasets present challenges based on the scarcity of polite speech events and the class imbalance that exists between different categories of polite speech. Data scarcity stems from the fact that polite speech is used in only a small fraction of classroom videos, so although a video may be an hour long, it might contain less than 10 instances of polite speech. Class imbalance occurs because some types of polite speech are used much more commonly than others, especially given the preschool classroom setting. Both data scarcity and class imbalance make training neural networks more difficult, as there are fewer examples over which to generalize, and there is less incentive for the model to learn to identify the minority class(es) because they contribute less to the model's loss function. To overcome the challenges of class imbalance and data scarcity, we experiment with different types of data augmentation and different deep neural network architectures.

1.1 Polite Language Definition

In this research, we identify polite language based solely on the words being spoken. This means that we do not consider the context or tone of voice in which words are spoken when determining whether speech constitutes polite language. Although these factors are likely important for determining whether a teacher-student interaction is truly respectful, identifying the context or tone used for these phrases is outside the scope of this thesis.

After consulting the CLASS Manual, watching several videos of teacher-student interaction in toddler classrooms and discussing with our collaborator, Dr. Jennifer LoCasale-Crouch with the Curry School of Education and Human Development at the University of Virginia, we decided on four categories of polite language phrases: "Good job", "Thank you", "Please", and "Thank you". The CLASS Manual specifies that teachers should use "language that communicates respect such as, 'please,' 'thank you,' and 'you are welcome'," and a "teacher freely responds to students' efforts and participation in activities...with positive comments" [5]. Within our four categories, we also hope to detect many other phrases (Table 1.1) that are used in the same spirit as these phrases (i.e. "Excellent" instead of "Good job", etc.). We do notice, however, that in toddler classrooms some of these types of phrases are much more common than others. For instance, "You're welcome" is very rarely used because toddlers do not typically use polite language and this phrase is commonly used as a response to someone saying "Thank you".

In building our detectors of polite language, we do not intentionally discriminate between polite speech spoken

by the teacher and polite speech spoken by children. However, as was previously mentioned, we expect the overwhelming majority of polite speech to come from the teacher when focusing on toddler classrooms. Additionally, a very large proportion of the training examples used to teach our polite language detection model come from adult speech, so it is possible that the model would perform poorly if asked to identify children's polite speech.

Table 1.1: Polite Language Taxonomy

Polite	Category	Phrase
Polite	Good job (Praise)	Very good Good job Good Great Great job Awesome Excellent Perfect Well done Fantastic Yes! Alright! That's it! Super Wonderful Nice job
	Thank you	Thank you Thanks
	Please	Please
	You're welcome	You're welcome No problem
Not Polite	Other	Other

1.2 Research Questions

For this thesis, we focused on the following research questions:

1. What is an effective deep learning architecture for speech recognition in a classroom environment?
2. How can we overcome challenges such as data scarcity and class imbalance?
3. Can augmentation of crowdsourced audio be used to improve robustness against overlapping speech and ambient noise in a classroom?
4. Does harnessing the pattern of polite language improve CLASS score prediction?

1.3 Outline

The rest of the thesis is organized as follows: Chapter 2 introduces background information and prior research. Chapter 3 describes our private dataset of CLASS-coded videos, as well as the datasets we use for training our polite speech detector. Chapter 4 describes the tool that we developed for efficiently crowdsourcing speech event labels from videos. Chapter 5 discusses the rationale for the neural architectures explored for polite language recognition. Chapter 6 provides an overview of the various experiments that we ran when training our deep neural network to distinguish between different types of polite speech. Chapter 7 highlights the key results of our experimentation on our dataset of crowdsourced clean speech. Chapter 8 explains how we attempt to address the challenge of class imbalance in our dataset of noisy classroom videos. Chapter 9 indicates which techniques for handling class imbalance were the most effective on the YouTube classroom dataset. Chapter 10 shows how well the model trained on YouTube videos performs on the CLASS dataset and explores the relationship between polite speech and CLASS scores. Chapter 11 discusses the ethics and biases that were part of this project. Chapter 12 provides a summary of our key results and concludes the paper.

Chapter 2

Prior Work

In a survey, broad evidence showed that deep learning has surpassed traditional approaches, such as gaussian mixture models and hidden markov models, for the task of converting speech to text [6]. Today, deep learning approaches have started using the raw audio as input instead of hand-crafted features such as MFCC, Chroma, or Log Mel-Filterbank (FBANK) features, allowing them to make use of all information related to the task at hand [7]. However, many approaches still use hand-crafted features as input [8, 9, 10]. Similarly, whereas before front-end (e.g. speech enhancement, source separation) and back-end (e.g. acoustic and language modeling) systems were trained separately, today jointly trained end-to-end systems outperform those that are trained separately [8, 9, 10].

2.1 Features for Speech Recognition

FBANK and MFCC are two commonly used feature types for speech recognition and machine learning for audio tasks [11]. FBANK features are calculated using the following steps [12, 13]:

1. Apply a pre-emphasis filter to amplify the high frequencies within the signal
2. Split the signal into frames (usually 25ms long, 10ms stride)
3. Apply a Hamming window function to each frame
4. Compute the Short-Time Fourier Transform on each frame to convert to the frequency spectrum
5. Apply a specified number of triangular filters on a Mel-scale (motivated by human perception of sound- more discriminative at lower frequencies and less discriminative at higher frequencies) to extract frequency bands

All of the steps used to calculate FBANK features are directly motivated by human perception of sound. Meanwhile, to calculate MFCCs a Discrete Cosine Transform is applied to the FBANK features, in order to remove the correlation between the features. Then, typically only the top 12 MFCCs are kept, and the rest are not useful for

speech recognition. These steps applied to convert FBANK into MFCCs are not consistent with human perception, but rather is motivated by the need for uncorrelated features for some machine learning models.

For instance, Gaussian Mixture Models and Hidden Markov Models were commonly combined to perform speech recognition before the rise in popularity of deep learning [11]. These models relied on MFCCs as input because they assume no correlation between input features. Now that deep neural networks have become the dominant method of speech recognition, there is evidence that FBANK features permit models to more accurately detect speech than if they solely rely on the MFCC [14]. Therefore, we try both approaches in order to compare their performance when training data is scarce and class imbalance is high.

2.2 End-to-End Speech Recognition

In 2013, Sainath et al. [15] were among the first to show that convolutional neural networks outperformed deep neural networks on large vocabulary continuous speech recognition (LVCSR). Their CNN was composed of only two convolutional layers and used 11×40 FBANK features. By limiting the number of parameters in their system, they determined that their CNN gave a 4-12% relative improvement over DNNs on the Switchboard corpus for LVCSR. Further, Saineth et al. were the first to find that pooling in time was helpful for speech tasks [8].

Later, Bi et al. [16] explored using much deeper convolutional neural networks for speech recognition, following the advances in CNN architectures used for image classification[17]. Their network consisted of 10 convolutional layers and 17×64 FBANK features. Through their experiments, Bi et al. find a 4-7% relative improvement compared to a baseline CNN [15] when using their much deeper CNN on the Callhome and Switchboard corpora for LVCSR. Sercu et al. [18] independently pursued a similar approach to deep convolutional neural networks for speech recognition, also inspired by the VGG architecture for image classification [17]. Their model makes use of small 3×3 kernels and uses multiple convolutional layers between each pooling layer, drawing inspiration from the VGG network. Sercu et al. also confirm that deep convolutional neural networks outperform prior, more shallow convolutional networks on speech tasks.

In 2016, Qian et al. achieved a new best word error rate (WER) on the Aurora-4 additive noise task and was competitive with LSTM acoustic modeling approaches on the AMI meeting transcription task, both of which are standard benchmarks [10] for speech recognition in noisy, multispeaker environments. Qian et al.'s model is trained on $\sim$14k utterances, half of which are clean speech and the other half are augmented with noise. Their approach converts speech to 17×64 FBANK features which are the input to a succession of ten convolutional layers, similar to their previous work in [16]. Following the convolutional layers are four densely connected layers. Finally, a softmax function outputs the senones (subphonetic units) that are given as input to a language model for word prediction. CNNs, combined with pooling, allow the model to have translational invariance, better capturing shifts in both time and frequency, and the feature extension from the typical 11×40 feature space [15] to the 17×64

feature space allows four additional convolutional layers that is shown to improve performance.

Since 2016, Wang et al. [19] and Tan et al. [20] have explored adding residual blocks to deep convolutional neural networks for speech recognition, similar to He et al.'s residual CNN model for image classification [21]. Wang et al. also explored the addition of a connectionist temporal classification loss function to infer speech-label alignments without an additional procedure following the output of the network [19]. Meanwhile, Tan et al. [20] explore factor aware training and cluster adaptive training in order to reduce the harmful effect of non-speech variability and improve the model's robustness.

For our purposes, we will use a similar architecture to that of Qian et al. [10], as the model is simple to implement, can easily be converted to our spoken phrase detection task, and was shown to perform well on corpora containing noisy speech with multiple overlapping speakers. In the future, we hope to explore techniques such as those used by Wang et al. [19] and Tan et al. [20].

2.3 Multispeaker Speech Recognition

Along with noisy environments, multispeaker environments are one of the settings where speech recognition systems still tend to perform poorly. Multispeaker speech recognition is of particular interest due to its application for meeting or conversation transcriptions. Modern approaches can be divided into two categories depending on the type of input: those that take input from a microphone array [22, 23] and those that use a single-channel approach [24, 25]. Using a microphone array, multispeaker speech recognition systems are able to separate speech from different speakers better and achieve a lower word error rate [6, 22]. However, for our purposes, we will focus on single-channel approaches, as this will ensure our model is universally applicable, regardless of recording equipment or setup.

One approach to single-channel multispeaker speech recognition was proposed by Suzuki et al. [24]. They generate single-channel audio from multi-channel recordings, as these are much easier to obtain transcripts for than monaural recordings of multiple speakers. To do this, they combine the signals together to form one audio file and then combine the transcripts into one unified transcript with special tokens that represent overlapping words and phonemes. They use a signal-to-noise ratio threshold of 10 dB to determine whether speech is overlapping or whether one speaker is dominant. Together with separate acoustic and language models, a garbage model is used to throw out speech decoded as overlapping to avoid negatively affecting results, choosing not to recognize overlapping speech itself. Rather than choosing not to make a prediction whenever overlapping speech occurs, we want our model to always provide a prediction for the probabilities of polite speech at a given point in time.

On the other hand, Chen et al. divide multispeaker speech recognition into three subproblems: frame-wise interpreting, speaker tracing, and speech recognition [25]. Frame-wise interpreting extracts features from the raw audio that facilitate the separation of overlapped speech into audio corresponding with individual speakers. Speaker

tracing processes these features to determine who is speaking at any given time. Finally, the speech recognition model draws on both prior modules to transcribe the speech from each speaker. To train their model, Chen et al. use a teacher-student transfer learning approach. The student must learn to produce outputs for each speaker given a mixed signal, while the teacher has access to the clean signals corresponding to each speaker. While this type of model be interesting to explore in the future, we choose to first establish a baseline performance on our classroom spoken phrase detection task before exploring more specialized architectures and training methods.

2.4 Multitask Learning Applied to Speech Recognition

Multitask Learning (MTL) has been used in a wide variety of cases to increase deep learning models' ability to learn generalizable features [26], especially when applied to target tasks where data is scarce [27]. Furthermore, there are several studies that have successfully used MTL to improve on speech recognition tasks. Jain et al. [28] demonstrated that jointly learning an accent classifier alongside speech recognition gave a relative performance boost of 10% on a test set with unseen accents when compared to a multi-accent baseline system. Kyun Kim et al. [29] demonstrated the effectiveness of using MTL with convolutional neural networks for emotion recognition in speech when data is scarce. Krishna et al. [30] compared hierarchical MTL with standard MTL when using connectionist temporal classification-based speech recognition, finding that hierarchical multitask training outperformed standard multitask training in experiments with large datasets but performed no better when data is more scarce. In our research, we apply MTL to determine whether it will benefit our model, given that our dataset is limited in size and exhibits class imbalance.

2.5 Speech Augmentation for Robust Recognition

A number of methods have been proposed for augmenting speech recognition datasets to improve robustness of models against different voices [31, 32, 33], different recording media and locations [23, 34, 35, 36], ambient noise [34, 36], and multiple speakers [23, 35].

One of the first approaches to augmenting speech recognition datasets to improve their ability to learn was vocal tract length perturbation (VTLP) [31]. In VTLP, a random warp factor is chosen for each utterance and applied via the vocal tract length normalization (VTLN) technique [37] to warp the utterance along the frequency dimension.

Compared with VTLP, Stochastic feature mapping (SFM) can be used to produce a larger quantity of extra training data before the gains begin to plateau [32]. SFM is inspired by voice conversion, statistically converting utterances spoken by one speaker to contain attributes of another speaker (i.e. age and accent).

Further augmentation can be accomplished by producing multiple versions of audio signals with different speed factors. Ko et al. [33] found that speed factors of 0.9, 1.0, and 1.1 helped improve on speech recognition tasks.

In our research, we employ speed, pitch, and background noise augmentation. Each of these types of augmentation is simple to implement, and we evaluate their effectiveness at improving model performance and generalization given the small size of our original training dataset.

2.6 Classroom Assessment Scoring System (CLASS)

The Classroom Assessment Scoring System (CLASS) is a protocol used by trained experts to evaluate teachers on several aspects of their teaching. For teachers to receive feedback using CLASS, human experts must either be present in the classroom or watch video recorded by the teacher. There are different CLASS dimensions and behavioral indicators for different age groups. In this research we focus on CLASS for toddler classrooms. The eight dimensions of CLASS for toddler classrooms are divided into **Emotional and Behavioral Support** and **Engaged Support for Learning** [1].

Dimensions in the Emotional and Behavioral Support category are as follows:

- **Positive Climate** reflects the warmth, respect, and enjoyment exhibited in the classroom through verbal and nonverbal communication. Some indicators of a positive learning climate include smiling, laughter, and respectful language between teachers and students.
- **Negative Climate** is characterized by teacher or student negativity, irritability, anger, or punitive control. Behavioral indicators include yelling, threats, harsh voices, sarcasm, and disputes.
- **Teacher Sensitivity** incorporates a teacher's awareness of their classroom, their responsiveness to children's needs, and the comfort of children with their teacher.
- **Regard for Child Perspectives** evaluates a teacher's ability to be flexible and adjust to their students, allow children to make choices, and support child independence.
- **Behavior Guidance** reflects a teacher's ability to monitor children's behavior in a proactive manner, supporting positive behavior and preventing problem behavior.

Meanwhile, Engaged Support for Learning includes the following dimensions:

- **Facilitation of Learning and Development** encapsulates how well a teacher provides opportunities for children to learn and explore, be engaged in class, and understand how information they are taught relates to them.
- **Quality of Feedback** is characterized by the amount of "scaffolding" provided to students, as well as encouragement and affirmation. One key behavioral indicator is recognition of effort or accomplishments (i.e. Good job or Thank you).
- **Language Modeling** describes how well teachers encourage the development of students' language skills.

2.7 Automated Classroom Observation and Feedback

Ramakrishnan et al. [3] proposed a system for analyzing classroom videos and producing estimated scores for the Positive and Negative Climate dimensions of CLASS. Their approach focused primarily on the visual component of classroom videos, developing a model to detect if a face is smiling or not smiling and whether that face belongs to a child or an adult. In this manner, they are able to get the average smile value for teachers and the average smile value for students at each frame in the video, feeding that information to an LSTM network to predict CLASS scores.

However, it is often the case that the teacher or students are off-camera or have their backs turned to the camera. When this occurs, visual-based prediction of CLASS scores performs poorly, as there are no visual cues available. Ramakrishnan et al. [3] incorporated energies and frequencies (MFCCs and Chroma features) from the audio component of classroom videos into an ensemble model for CLASS prediction.

Research has also investigated the use of audio from classrooms to automatically categorize classroom activities [38, 39, 40], providing teachers with feedback on how they spend their class time. These approaches focus on classifying the current state of classroom activity, as well as segmenting audio into teacher and student speech. Our work is unique in that we detect a specific category of speech, *polite speech*, and investigate the utility of these features for CLASS score prediction.

Additionally, although Ramakrishnan et al. [3] incorporated simple audio features into their model for CLASS prediction, it is not clear what elements of the audio this detector is using in its predictions, meaning the detector would not be interpretable by teachers. Furthermore, it is quite possible that this simple audio model picks up on moments of chaos and commotion within the classroom, which may then be correlated with Positive or Negative Climate. Therefore, we believe *polite speech* detection is worthy of investigation both due to the interpretability it would provide and the potential for it to add predictive power to Ramakrishnan et al.'s CLASS score predictor.

Chapter 3

Datasets

Through our collaborator at the University of Virginia, we have access to a dataset of CLASS-coded videos of real classroom interaction between teachers and students. Eventually we want to apply our polite speech detector to this dataset in order to progress towards the goal of automated CLASS score prediction. However, this dataset is not labeled for polite speech events and is private, preventing us from crowdsourcing polite speech labels. Therefore, we crowdsource polite speech labels for a dataset of public classroom videos from YouTube that are similar in content to the CLASS-coded videos, using this dataset for training. Finally, since the number of polite speech labels that occur in our YouTube dataset is still relatively low and some types of polite speech are very rare, we crowdsource a new dataset of exclusively polite speech recordings to further increase the amount of training data available for our detector.

3.1 CLASS-coded Dataset of Real Classroom Videos

Our target dataset for this research consists of 106 CLASS-coded videos collected by our collaborators at the University of Virginia. These videos were collected in American preschool classrooms around the University of Virginia. A few adults and several young children ($<$ 5 years old) are in each video, and at any point in the videos, multiple people might be talking, shouting, crying, or singing simultaneously. As previously, mentioned these videos have been coded according to the CLASS protocol, meaning that for every 15 minute segment within each video, there are labels for each of the eight CLASS dimensions for toddler classrooms. Therefore, by detecting polite language in the videos in this dataset, we can determine if and how polite language relates to each CLASS dimension. However, these videos have not been labeled for polite speech events, and due to data privacy concerns, we cannot crowdsource polite speech event labels for this dataset. Therefore, we must find a public dataset, that is similar to this private dataset, to use for the training of our detectors.

3.2 YouTube Dataset of Noisy Speech

While we have a dataset of CLASS-labeled videos, these videos have not been manually transcribed to permit us to use them for training our speech recognition model. Furthermore, we cannot crowdsource transcriptions for these videos due to privacy concerns. Therefore, we use a dataset of 57 YouTube videos [2] of teacher-student interaction in toddler classrooms, similar to the type of interaction found in the CLASS videos. In these YouTube videos, as in the private CLASS-labeled videos, overlapping speech and ambient noise is quite common. Unlike in the CLASS video dataset, many of the YouTube videos chosen are from english language learner classrooms. What this does mean, however, is that the primary language used in these videos is still English, just as in the CLASS videos. We also note that, as there are relatively few YouTube videos in our dataset, the number of unique speakers is low, especially since some speakers are the same across multiple videos. This might affect the ability of our model to generalize if trained only on the unaugmented YouTube dataset.

We used Amazon Mechanical Turk to obtain annotations for each video whenever a phrase indicating polite or respectful language was used. We focused on four categories of polite language: *Good job*, *Thank you*, *Please*, and *You're welcome*. As described in Chapter 4, workers annotated polite language phrases throughout the entire duration of each video.

3.3 Speech Event Label Reconciliation: YouTube Dataset

After conducting pilot Human Intelligence Tasks (HITs) on Mechanical Turk with 5 workers assigned to each of 3 videos, we created HITs for the remaining 54 videos, seeking annotations from 3 workers for each video. Upon receiving 3 (or 5) sets of annotations per video, we were faced with the task of reconciling the similarities and differences between them to create one set of annotations to use for the training and validation of our model. We use a heuristic-based approach to perform the set union operation on the different annotation sets, as seen in Fig. 3.1. As different labels for the same occurrence of polite language may be shifted slightly in time due to human error, performing the set union is nontrivial. To combine labels, we first merge the annotation sets for a given video, keeping them sorted by the time at which they occurred, and then traverse the combined annotations, merging two or more consecutive labels from the same category if they are within two seconds of each other. Finally, we review the audio around each crowdsourced annotation to verify that the correct type of polite speech is indeed used at this time during the video. This last step was both made feasible and important due to the limited size of our dataset.

As shown in Table 3.1, *Good job* received the most labels with 703 distinct appearances in the YouTube dataset while *Thank you*, *Please*, and *You're welcome* only appeared 62, 46, and 3 times respectively over all 57 videos. While this data might be sufficient for training and validation for the *Good job* class, we must explore other options to obtain sufficient training data for the other classes.

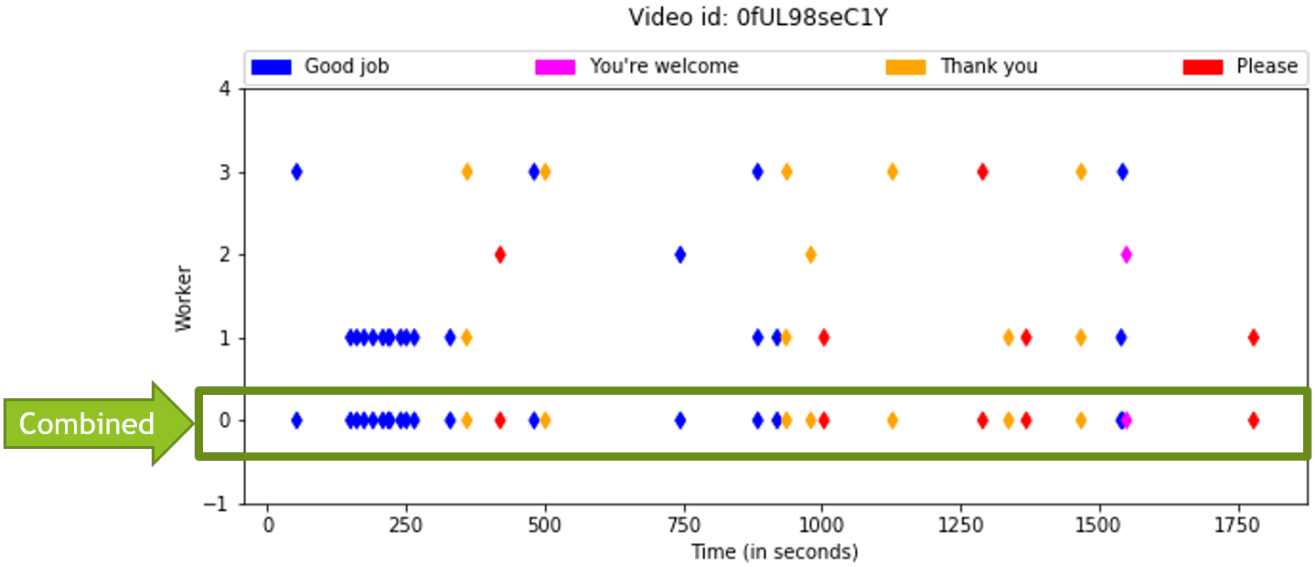


Figure 3.1: This figure shows how workers' annotations were combined using the set union operation to get the labels for polite language usage in the YouTube dataset.

Table 3.1: Polite Language in YouTube Dataset

Type of Polite Language	Number of Labels
Good job	703
Thank you	62
Please	46
You're welcome	3

3.4 MTurk Dataset of Clean Speech

As we have insufficient annotated training data for detecting polite language in a classroom environment, we explored other datasets and methods to acquire a more expansive set of training examples. At first, we viewed speech recognition datasets such as Mozilla and LibriSpeech as promising sources since they are freely available, are composed of crowdsourced recordings with some ambient noise and many different speakers, and provide many more training examples for the *Good job*, *Thank you*, and *Please* classes of polite language. However, these datasets do not provide word-level timestamps that would allow us to easily isolate polite language phrases within the audio files given the transcripts. Therefore, we instead crowdsource our own dataset of polite language phrases.

Beginning with a pilot task and later in an additional collection phase, we asked workers on Mechanical Turk to record themselves while they say a specified phrase. In particular, we launched separate assignments for 21 different phrases (see Table 1.1), such that one worker could not complete an assignment belonging to the same phrase multiple times. This ensured that we would gather a more diverse dataset with more distinct voices.

After collecting these recordings, we note that all of these recordings are made by adult speakers, as workers on Mechanical Turk are required to be at least 18 years old. Furthermore, most audio contains little or no background noise or overlapping speech, although a few recordings had music in the background. Also, we acknowledge that

the manner in which someone speaks when recording their voice and talking to a computer may be very different from how they would speak respectfully to a toddler. However, we attempt to partially mitigate this discrepancy by instructing Mechanical Turk workers to first listen to an example of polite language from the YouTube dataset before recording themselves saying the same phrase. By priming the workers in this manner, we hope that their speech will exhibit increased authenticity and better capture the qualities of the polite speech found in the YouTube and CLASS videos.

When launching these tasks on Mechanical Turk, we knew that the recordings collected would consist of mostly clean speech, without many interruptions or significant background noise. In fact, we specifically instructed workers that background noise was encouraged for our purposes. Still, we knew that further augmentation would be required to increase the utility of the data we were gathering. Thus, we instructed workers to begin recording, wait one second before saying the assigned phrase, and then wait one more second before ending the recording. This approximately one second buffer allows us to easily capture multiple training examples from the same recording by shifting along the time dimension. Further, this allowed us to easily overlay background noise to simulate the ambient noise that might be present in a classroom before, during, and after a teacher uses polite speech. For further details, see Chapter 5.

Chapter 4

Design of a Speech Event Labeling Tool

In order to crowdsource labels for polite speech in our dataset of publicly available YouTube videos, we first had to make a custom tool through which workers on Mechanical Turk would interact with the videos, as Amazon does not have a premade template that would allow workers to easily and efficiently complete this task.

Therefore, we use Amazon's ExternalQuestion interface to insert a custom HTML/JavaScript page onto a pane on the Human Intelligence Task (HIT) page. The custom HTML/JavaScript page provides workers with detailed instructions (Fig. 4.1), allows them to easily watch the video that they are assigned, and annotate each time they hear a polite speech phrase in the video's audio (Fig. 4.2).

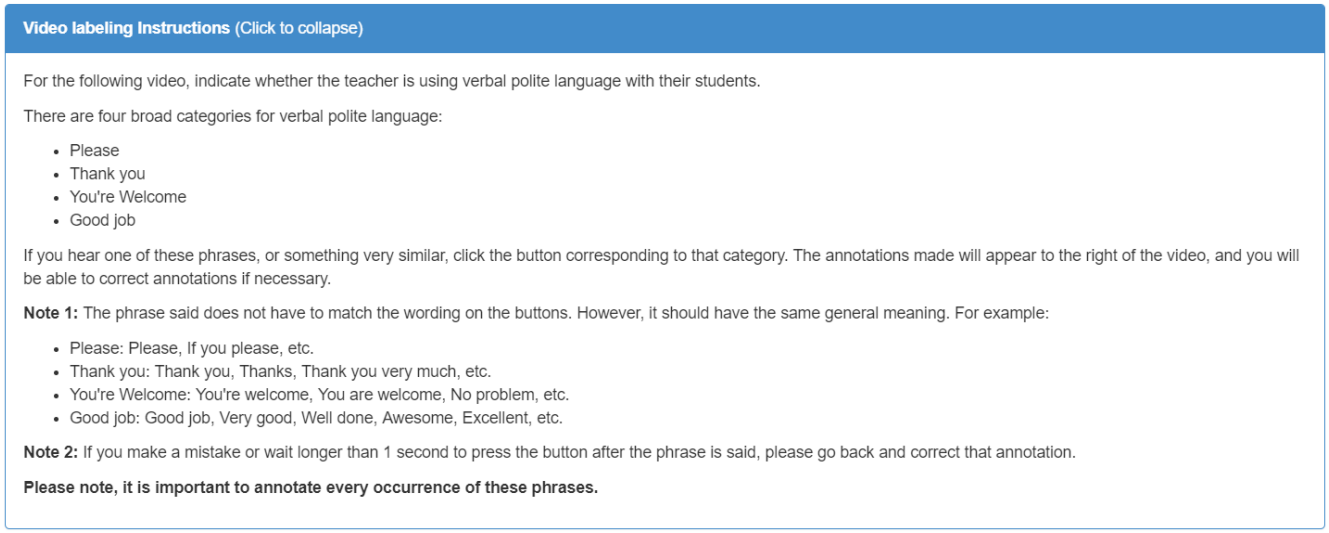


Figure 4.1: These are the instructions workers were provided with before labeling speech events in YouTube videos.

In our instructions, we indicate the categories of polite speech that are of interest to our research, and we also specify that we would like them to label polite speech phrases that are similar in meaning to the categories: Good job, Thank you, Please, and You're welcome, providing them with several examples of different phrases for each category. Further, we explain that a list of their current annotations will appear on the side of the page, and that

they can remove annotations if they were made incorrectly. Finally, we instruct workers to label polite speech events with a tolerance of 1 second and ask them to correct the annotation if it is more than 1 second after the phrase was said.

Figure 4.2: Workers on Mechanical Turk pressed a button each time they heard a phrase similar to one of our four classes: *Please*, *Thank you*, *You're welcome*, and *Good job*.

To actually perform the speech event labeling, workers simply press play on the YouTube video, and listen for polite speech phrases. If they hear a polite speech phrase, they will click the button corresponding to the phrase's category on the right side of the interface. Then, the video will be paused automatically and the annotation will appear in the list on the right. This pause allows labelers to go back and correct the annotation if needed. Otherwise, they can resume playing the video and listening for annotations. Finally, they will submit their annotations using the button at the bottom of the interface. We do not require that there be a minimum number of annotations for a worker to submit because it is quite possible that there is no polite speech in some of our YouTube videos.

When a labeler submits their annotations, we receive an xml document containing an array of binary tuples. Each tuple is a category-timestamp pair, i.e. ("Good job", 13.64), indicating one instance of polite speech in a specific video. These annotations provide the approximate end-time of the polite speech utterance.

Chapter 5

Proposed Neural Network and Training Method

As a starting point, we use a similar convolutional architecture to that proposed by Qian et al. [10], which consists of 10 convolutional layers followed by four fully connected layers, as seen in Fig. 5.1. Following Qian et al., we use non-overlapping pooling layers and pad in both the temporal and feature dimensions in each convolutional layer. After each convolutional layer and each fully-connected layer, we apply batch normalization [41] and then use ReLU activation [42, 43].

5.1 Very Deep Convolutional Neural Network

When FBANK features are concatenated between several consecutive timesteps, the resulting two-dimensional matrix can be visualized as a spectrogram, displaying the filterbanks for the audio during that window of time. This spectrogram will appear different and contain different "visual" characteristics depending on what is occurring in the audio it represents. Therefore, convolutional neural networks can be used in a similar manner to those used for object recognition in the field of computer vision.

The underlying motivation for the use of convolutional neural networks in speech recognition, as in computer vision, is their ability to incorporate translational, size, and distortion invariance into the model [44]. This is accomplished through properties and constraints of convolutional layers: local receptive fields, weight sharing, and spatial sub-sampling. Because of the local receptive field, earlier convolutional layers will detect simpler features that occur within very small subsets of the overall input. Then, when subsequent convolutional layers are added, these layers are capable of detecting more complex, higher-level features. Meanwhile, weight sharing is imposed based on the assumption that features learned on one part of the input may very well be useful when applied to other areas of the input. Thus, these learned features are replicated multiple times within the weights matrix.

This structure of convolutional layers already makes them robust to shifts and distortions in the input; meanwhile, sub-sampling further reduces the sensitivity to shifts and distortions. Sub-sampling occurs through the addition of pooling layers between convolutional layers. These pooling layers reduce the resolution of feature maps by taking a sub-sample of the feature map and replacing that sampled area with a single number, commonly either the average or maximum of those sampled values. By reducing the spatial resolution, the emphasis on the precise location of any detected features is diminished.

The concepts of translational, size, and distortion invariance are helpful in speech recognition, as well as in object detection. In our neural network model, if polite speech occurs within a given context window, we do not care where the polite speech is actually located temporally. Similarly, the volume at which polite speech is spoken should not affect our model's ability to detect it, given that the polite speech was loud enough to be captured by the microphone used for recording. Finally, distortion of the audio may occur due to background noise, difference between voices and accents, or difference between recording devices; ideally, this distortion should not impact the ability to recognize polite speech. Therefore, convolutional neural networks, such as the architecture proposed by Qian et al. [10] seem like a logical fit for our research.

5.2 Temporal Pooling

Based on the intuition that pooling in the temporal dimension provides further robustness against shifts and distortions in time, we investigate extending the model proposed by Qian et al. [10] to add more pooling in time and further convolutional layers as seen in Fig. 5.1. In prior works, temporal pooling was actually found to decrease performance on speech recognition tasks [15]. However, Qian et al. [10] demonstrated that temporal pooling was beneficial for very deep convolutional neural networks when using temporal padding. Given that our input feature dimensions are 158×64, whereas the input dimensions in Qian et al.'s model were 17×64, we experiment with adding various levels of further pooling in time due to our substantially larger time dimension. The results of this experimentation can be found in Chapter 7.

5.3 Preprocessing

Before training our model, we first convert the audio files into mono-channel, 48000 Hz, wav files using ffmpeg. Then, we process the wav files into 64-dimensional FBANK features, as recommended by [10], using a time window of 25ms and a time step of 10ms. Next, we create context windows from consecutive FBANK feature vectors (Fig. 5.2). We choose to concatenate 158 consecutive FBANK feature vectors, as this corresponds with a time span of approximately 1.6 seconds, which we observed to be the longest time taken for an utterance to be spoken out of 25 randomly selected polite speech utterances from the YouTube dataset. Between context windows, we use a time step of 100ms. Each 158×64 context window is then assigned a label corresponding to one of our categories of polite

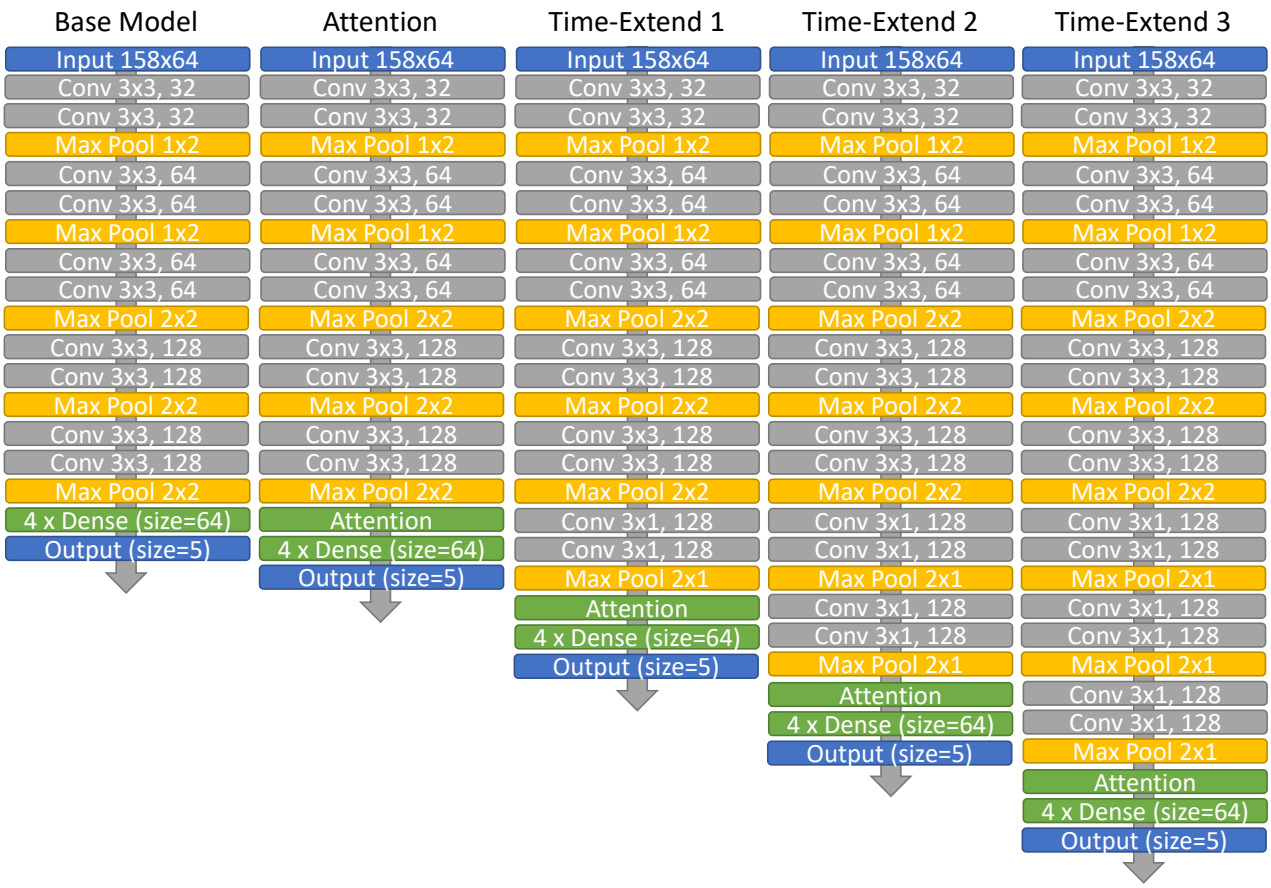


Figure 5.1: The different architectures that we compare in Chapter 7 are shown here. These diagrams show only the singletask versions of the architectures. Batch Normalization is applied (after each convolutional/fully-connected layer) and ReLU Activation is used (after each Batch Normalization layer), but both are omitted here for clarity.

language or a label indicating that no polite language is present, using the labels provided by MTurk workers.

5.4 Train/Test Split

In our YouTube dataset of noisy speech, we split the data into training and testing sets by video, meaning that no video in the training set can also appear in the testing set. We split the videos based on file size, such that approximately two-thirds of the videos are in the training dataset and the remaining videos are in the testing dataset. It is worth noting, however, that there are teachers that appear in multiple videos in the YouTube dataset, and we do not enforce that these videos are both put into the same dataset (training/testing).

Similarly, for the Mechanical Turk dataset of clean speech, we split the dataset into training and testing sets by file size, with an approximate ratio of two-thirds to one-third. Here again, it is possible that the same speaker appears in both train and test sets, but if this happens the phrase that the speaker is saying must be different because we only allow each worker on Mechanical Turk to submit up to one recording of each phrase.

In both datasets, when data augmentation is applied, the data augmentation is only applied to the training

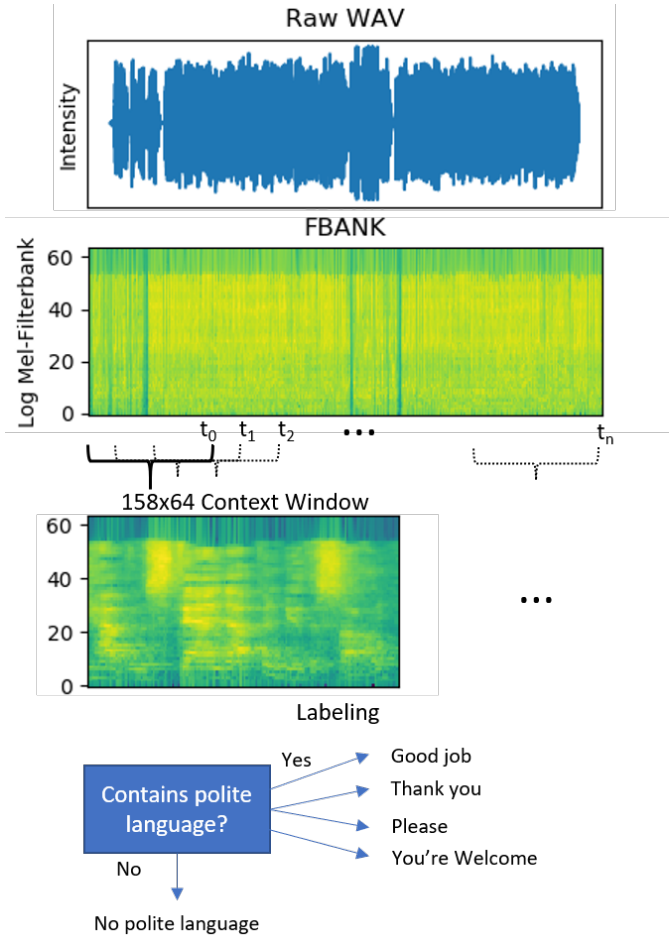


Figure 5.2: This figure shows how we process audio into "context windows" of concatenated FBANK features, which serve as the input for our CNN.

dataset, and the testing dataset remains unaltered.

5.5 Data Augmentation

The audio crowdsourced from Mechanical Turk, while not necessarily in a controlled environment still contains much less background noise and generally only features one speaker at any point in time. We therefore augment these audio tracks with the ambient noise and multiple simultaneous speakers common in classroom environments. To accomplish this, we simply overlay two audio tracks, the clean speech recording from Mechanical Turk and a clip containing ambient noise and overlapping speech that is taken from one of the YouTube videos that is not in the YouTube Test Set.

5.5.1 Varying Pitch and Speech

Previous work has explored data augmentation for speech recognition via small shifts in pitch and in speed from the original audio. Similarly, we augment our dataset through both of these approaches.

For pitch-based augmentation, we use ffmpeg to adjust the tempo (3/4 and 4/3) and the sampling rate accordingly (48000 * 4/3 and 48000 * 3/4, respectively). Then, the audio is resampled back to 48000 Hz to ensure it is consistent with the rest of the normal dataset. This procedure leaves us with three copies of each audio clip, one normal, one slightly lower-pitched, and one slightly higher-pitched.

For speed-based augmentation, we again use ffmpeg to adjust the tempo (this time using values 0.9 and 1.1). However, this time we hold the sampling rate constant, causing the only change to be the speed of the audio. This procedure will result in an additional two copies of the original audio, one that is spoken slightly slower, and one that is spoken slightly faster.

5.5.2 Adding Background Noise

As a third type of augmentation, we decided to take the crowdsourced audio and add background noise to make it sound more like a classroom environment (Fig. 5.3). The simplest way to do this is to overlay the two audio clips. We do this using the pydub python library. The background noise that we use for augmentation comes from the YouTube dataset, and we ensure that only background clips that are in the training dataset are used to prevent overlap between the train and test sets that might confound evaluation.

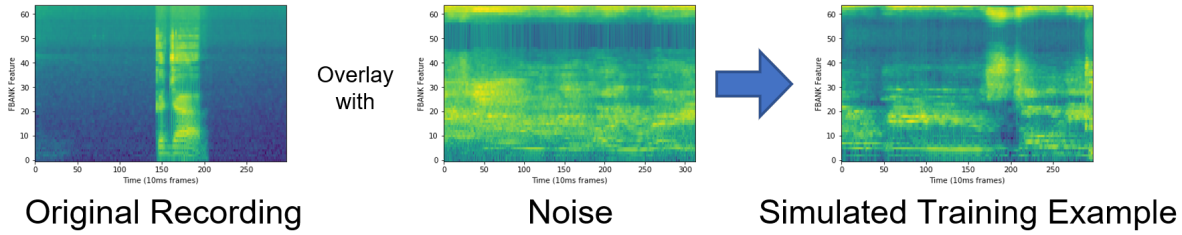


Figure 5.3: Clean speech recordings are augmented with background noise sampled from our YouTube dataset to produce more interesting examples.

As the background clips are longer than the crowdsourced audio clips, we truncate them to be the same length as the crowdsourced audio. For each crowdsourced clip, we randomly select background clips to use for augmentation. In this manner, we can create many diverse training examples with relative ease by using different background noise. Further, we suspect that by using the "background noise" training examples as the tracks that we overlay for augmentation, we encourage the deep learning model to learn that the distinction between classes is related to what is said in the crowdsourced speech rather than what is said in the "other" speech. In other words, the model should learn to classify speech as "other" if none of the polite language phrases are present in the speech, rather than looking for the specific phrases that are said in the "other" examples.

Chapter 6

Experiments

We conduct experiments in order to attempt to find an effective deep learning architecture for speech recognition in a classroom (Research Question 1), as well as to compare different approaches for audio augmentation (Research Question 3).

6.1 Evaluation Metrics

The metrics we will use for evaluation are as follows:

- **Cross-Entropy Loss (Test set)** - Measure of how much error there is in the prediction probabilities for the test set.
- **Accuracy (Test set)** - Indicates percent of test set examples that are predicted correctly.
- **AUC Score** - AUC gives us a threshold-independent metric that is unaffected by class imbalance with which to compare models.

6.2 FBANK vs. MFCC

As mentioned in Chapter 2, FBANK and MFCC are two commonly used feature types for speech recognition and machine learning for audio tasks. The calculation of FBANK features mimics the process used for human perception of sound. Meanwhile, MFCCs are calculated by applying a Discrete Cosing Transform to decorrelate the FBANK features, as machine learning models that were previously used for speech recognition assumed no correlation between different input features.

Now that deep neural networks have become widely used for speech recognition, there is evidence that FBANK features permit models to more accurately detect speech than if they solely rely on the MFCCs. Therefore, we try

both approaches in order to compare their performance when training data is scarce and class imbalance is high. We find that FBANK features improve our model's performance, and therefore proceed with FBANK features for the remainder of our experiments.

6.3 Normalization

After computing FBANK features, we experiment with applying normalization to the log filterbank features to determine whether normalization will allow our models to train more reliably. Specifically, we compare two types of normalization, normalization within a training example and normalization across all training examples.

For normalization within a training example, we find the minimum and maximum filterbank features across the 64 features in the training example. Then, we subtract the minimum feature from each of the 64 features and divide by the difference between the maximum and the minimum features, bounding feature values between 0 and 1. We then multiply by 2 and subtract one to zero the mean feature value and bound feature values between -1 and 1. This strategy also has the advantage of being easy to implement for testing our model on new data, as the normalization is not dependent on features extracted from other audio files.

For normalization across training examples, we use a similar approach- however, we find the minimum and maximum filterbank features across all of the training examples rather than just the one we are looking at. Then, we store these "global" values and for each training example, subtract the minimum and divide by the difference (maximum - minimum), as before, later centering the mean at zero and the lower and upper bounds at -1 and 1, respectively. For this strategy, we just have to ensure that we save the minimum/maximum values found from the training data, in order to apply the same normalization to new data after the fact.

Through our experiments, we find that input normalization does not seem to affect model performance. Therefore, we proceed using unnormalized FBANK features for the experiments that followed.

6.4 Attention

Within the broader machine learning community, attention mechanisms have been shown to improve model performance and improve the ability of deep learning models to identify salient features [45]. We are interested in whether a simple form of attention, seen in Fig. 6.1, will improve the performance of our models for speech recognition. In this case, we find that attention located immediately after the convolutional features are flattened increases our model's performance. Therefore, attention is used in all of our other experiments.

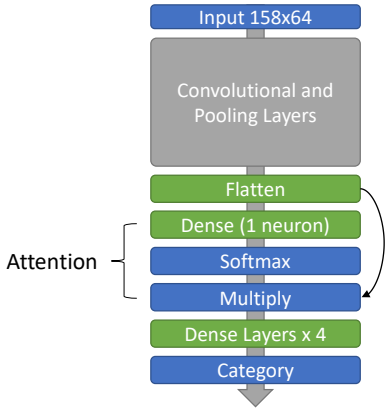


Figure 6.1: Attention is applied after the convolutional filters are flattened, potentially learning to identify the most relevant features created by the convolutional network.

6.5 Multitask Learning

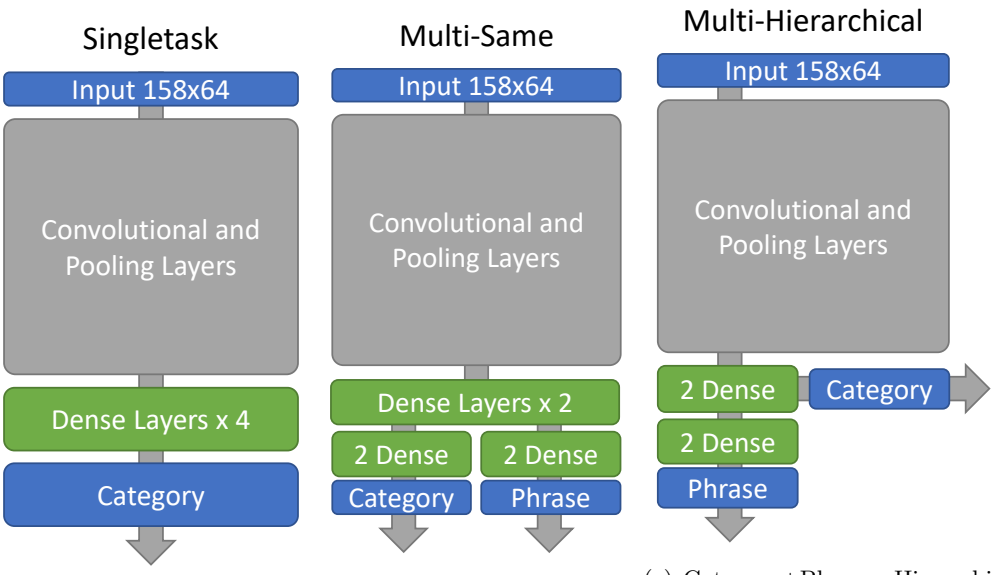
As our dataset is small, we employ multitask learning with the hope that this will improve the ability of our model to generalize beyond the training dataset. Additionally, we explore whether there is a difference between hierarchical (Fig. 6.2c) and traditional multitask learning (Fig. 6.2b) for our problem, since it is in a data-scarce setting. Our intuition for trying hierarchical MTL also lies in forcing the network to preserve information in the lower, convolutional layers, rather than learning useless convolutional filter maps and simply predicting the dominant class in the final layers of the network.

Further, traditionally MTL involves learning related but distinct tasks in the sense that one task is not simply a reformulation of another task. In our research, we have labels for polite speech categories, as well as the specific polite speech phrases. We therefore want to determine whether using MTL with multiple versions of the same task (coarse to fine grain) will still be effective at improving generalization of learned representations.

6.6 Further Pooling in Time Dimension

As our feature dimensions (158x64) are considerably larger in the time dimension than the features used by Qian et al. [10] (15x64), we experiment with adding additional convolutional layers and max pooling layers that focus on the time dimension. Motivation for adding additional max pooling layers is that these layers provide increased translational invariance in the time dimension and allow for learned features in the later convolutional layers to identify patterns over a longer time span in the original input features.

Specifically, we experiment with adding between one and three additional max pooling layers. Before each additional pooling layer, we add two more convolutional layers. Each added convolutional layer has a filter size of 3x1, and each added pooling layer has a window size of 2x1. When comparing models with varying levels of additional time pooling, we constrain that the number of parameters be similar in order to facilitate a fair



(a) Category only

(b) Category+Phrase: Same

(c) Category+Phrase: Hierarchical

Figure 6.2: **(a)** shows the singletask architecture, where the model's only objective is to predict whether audio belongs to the *Good job*, *Thank you*, *Please*, or *You're welcome* classes. **(b)** adds an additional objective using the traditional MTL formulation- to determine exactly what the phrase is that was said. Lastly, **(c)** creates a hierarchical multitask framework, staggering category and phrase-level predictions in a coarse-to-fine manner.

comparison.

6.7 Types of Augmentation

In this research, we apply three types of augmentation to the training dataset: pitch augmentation, speed augmentation, and background noise augmentation. We conduct experiments to determine the individual effects of each of the three types of augmentation, as well as the combined effect of all three. The dataset composition when using each approach is shown in Table 6.1.

	Good job	Thank you	Please	You're welcome
No augmentation	4230	522	383	276
With pitch augmentation	12569	1603	836	1148
With speed augmentation	12346	1575	809	1103
With background augmentation	25380	10962	5796	8043
Combined training	46065	13618	7165	9911
Evaluation	1880	330	147	307

Table 6.1: MTurk Augmentation Composition

6.8 Amount of Crowdsourced Data

When crowdsourcing training data, it would be useful to know how much benefit will be gained from paying for additional data. Therefore, we will test how the performance of our model suffers when we decrease the training dataset to 25%, 50%, and 75% of its full size. Note that all three types of data augmentation are applied to each of the "clean" examples in each of these conditions that we compare. Table 6.2 shows the dataset compositions when varying the percentage of the crowdsourced data that is utilized for training.

	Unique Speakers	Good job	Thank you	Please	You're welcome
25%	57	10951	3531	2023	2545
50%	61	22227	7092	3541	4612
75%	62	34746	11110	5189	7527
100%	62	46065	13618	7165	9911
Evaluation	60	1880	330	147	307

Table 6.2: MTurk Composition: Varying Percentage of Utilized Data

Chapter 7

Results: Mechanical Turk Clean Speech Dataset

7.1 FBANK vs. MFCC

Table 7.1 shows the results of our comparison between using MFCC and FBANK features with our Time-extend 1 model (Fig. 5.1) using the Multi-Hierarchical MTL formulation. As was expected, we find that using FBANK features provides a 4% relative increase (Avg. AUC) in the model's ability to distinguish between polite speech categories. This is likely due to the extra available information stored in FBANK features that is lost when the discrete cosine transform is used to decorrelate FBANK features and transform them into MFCCs.

Table 7.1: Feature Type Comparison

Model	Loss	Acc	AUC-GJ	AUC-TY	AUC-PLS	AUC-YW	AUC-AVG
MFCC	0.4872	0.7845	0.9258	0.9299	0.9713	0.9883	0.9538
FBANK	**0.0758**	**0.9790**	**0.9971**	**0.9983**	**0.9996**	**0.9986**	**0.9984**

7.2 Normalization

After comparing different methods for normalizing FBANK features (Table 7.2), we find that there is not much difference between raw FBANK features and normalized FBANK features with regard to their ability to be used for training deep convolutional neural networks. This finding is in contrast to traditional wisdom within machine learning that advocates for input normalization to improve model training and convergence. As seen in Fig. 7.1, the model trained using unnormalized features actually seemed to exhibit a smoother decrease in test loss and increase in test accuracy over all training epochs. Therefore, in the remainder of our experiments, we use unnormalized features as the input to our models.

Table 7.2: Normalization Type Comparison

Model	Loss	Acc	AUC-GJ	AUC-TY	AUC-PLS	AUC-YW	AUC-AVG
Unnormalized	0.1352	**0.9565**	**0.9931**	0.9938	**0.9970**	**0.9980**	**0.9955**
Normalized-Within	**0.1351**	**0.9565**	0.9892	**0.9950**	0.9946	0.9967	0.9939
Normalized-Across	0.2027	0.9489	0.9882	0.9917	0.9958	0.9930	0.9921
Normalized-Within-Across	0.1806	0.9437	0.9875	0.9919	0.9940	0.9927	0.9915

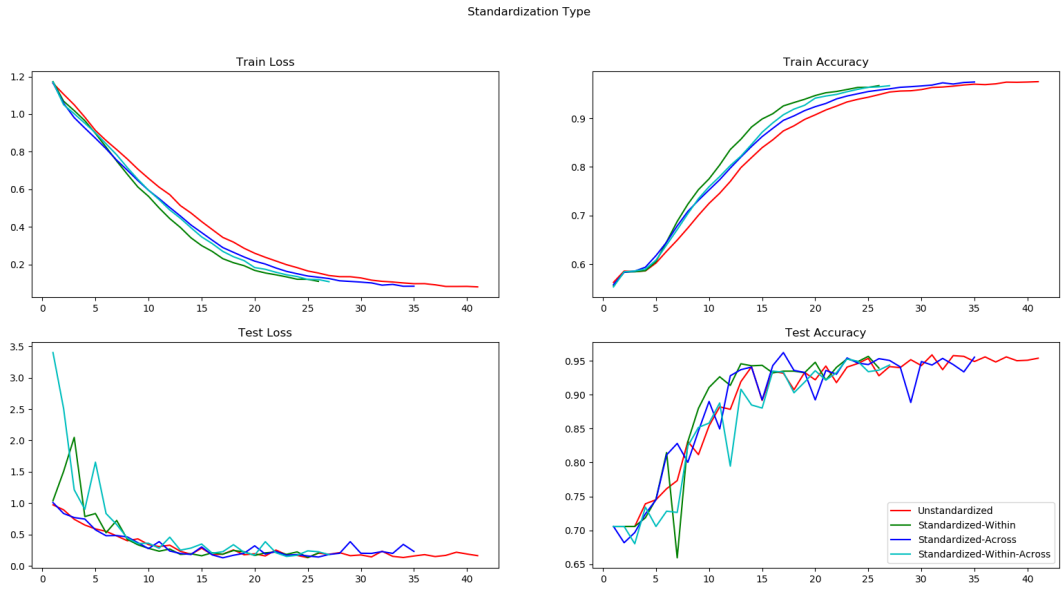


Figure 7.1: Unnormalized and Normalized-Within conditions reached the lowest test loss and highest test accuracy. The model trained on unnormalized FBANK features had a smoother decline in test loss and smoother increase in test accuracy over the course of training.

7.3 Attention

In our experiment comparing models with and without attention that are otherwise identical, we find that attention leads to an increase in overall AUC and a decrease in cross-entropy loss (Table 7.3), suggesting that an attention layer immediately following the flattening of the convolutional feature maps might allow the model to better learn which features to weight more than others. Thus, all models in other experiments will use an attention layer following the convolutional and pooling layers.

Table 7.3: Attention Comparison

Model	Loss	Acc	AUC-GJ	AUC-TY	AUC-PLS	AUC-YW	AUC-AVG
No Attention	0.1457	0.9508	0.9889	0.9882	**0.9982**	**0.9985**	0.9935
Attention	**0.1352**	**0.9565**	**0.9931**	**0.9938**	0.9970	0.9980	**0.9955**

7.4 Multitask Learning

On the Mechanical Turk dataset of clean speech, we find that introducing phrase classification as a secondary task to polite speech category classification improves the model's ability to distinguish between classes of polite speech, with a relative improvement of % (Table 7.4). We also find that hierarchical MTL, moving category-level classification upstream of phrase-level classification, provides further improvement upon the traditional MTL formulation. This result contradicts the findings of Krishna et al. [30], as we find hierarchical MTL to outperform traditional MTL even in small datasets. These findings also suggest that MTL can be useful even if the multiple tasks are very similar in nature. In subsequent experiments, hierarchical MTL is indicated by "+ Category-Phrase", and the model is otherwise assumed to use traditional MTL.

Table 7.4: Multitask Learning Comparison

Model	Loss	Acc	AUC-GJ	AUC-TY	AUC-PLS	AUC-YW	AUC-AVG
SingleTask	0.1449	0.9497	0.9907	0.9877	0.9948	0.9981	0.9928
Multitask-Same-Level	0.1352	0.9565	0.9931	0.9938	0.9970	0.9980	0.9955
Multitask-Category-Phrase	**0.1138**	**0.9595**	**0.9934**	**0.9957**	**0.9978**	**0.9987**	**0.9964**

7.5 Further Pooling in Time Dimension

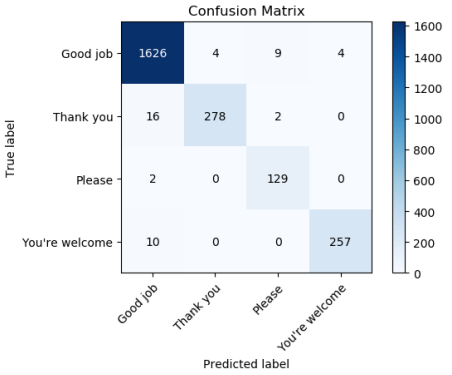
In Table 7.5, we find that additional pooling in time and the corresponding increased convolutional depth improve model performance as 1 and 2 further pooling layers are added (and the corresponding 2 and 4 convolutional layers, respectively). When 3 pooling layers and 6 convolutional layers are added, model performance decreases. Meanwhile, the best three models for polite language classification on the Mechanical Turk dataset of clean speech occur when hierarchical MTL is used with 1 and 2 additional pooling layers, as well as when traditional MTL is used with 2 additional pooling layers. In subsequent experiments, we choose to use the Conv-12-Pool-6 + Category-Phrase model, as it has the highest average AUC of all our models.

Table 7.5: Further Pooling in Time Dimension

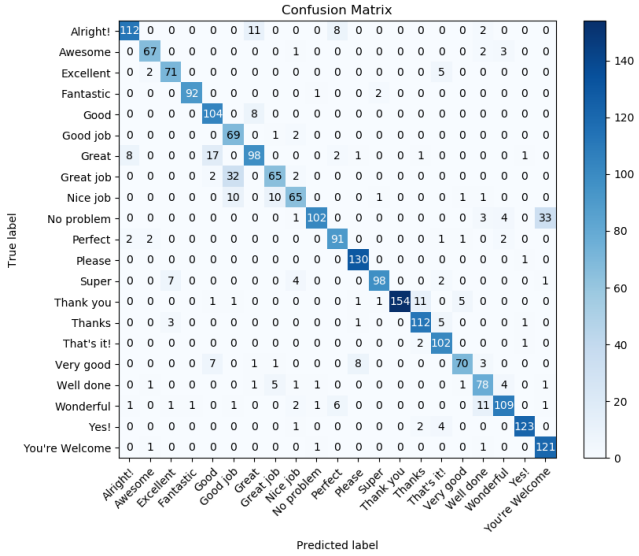
Model	Params	Loss	Acc	AUC-GJ	AUC-TY	AUC-PLS	AUC-YW	AUC-AVG
Conv-10-Pool-5	996k	0.1352	0.9565	0.9931	0.9938	0.9970	0.9980	0.9955
Conv-12-Pool-6	928k	0.1074	0.9673	0.9940	0.9966	0.9997	0.9962	0.9966
+ Category-Phrase	920k	0.0758	0.9790	**0.9971**	**0.9983**	0.9996	0.9986	**0.9984**
Conv-14-Pool-7	944k	**0.0729**	0.9790	0.9964	0.9976	**0.9999**	0.9978	0.9979
+ Category-Phrase	935k	0.0750	**0.9831**	0.9965	0.9959	0.9996	**0.9994**	0.9978
Conv-16-Pool-8	1009k	0.1120	0.9692	0.9952	0.9960	0.9983	0.9984	0.9970
+ Category-Phrase	1000k	0.1124	0.9703	0.9963	0.9977	0.9953	0.9986	0.9970

Fig. 7.2a and Fig. 7.2b show the confusion matrices for the Conv-12-Pool-6 + Category-Phrase model for category and phrase prediction respectively. In Fig.7.2a, we observe that the model is able to distinguish between types of polite speech in our dataset of clean speech recordings. Meanwhile, in Fig. 7.2b, we notice that there

are some combinations of phrases that are tougher for the model to distinguish between, but overall the model is able to distinguish between the specific phrases that compose each polite speech category. Some pairs of phrases that are commonly confused by the model include: "Good" and "Great", "Great job" and "Good job", and "No problem" and "You're welcome". In some cases, part of the phrase is identical to the confused phrase, while in all such common instances, the two confused phrases belongs to the same category.



(a) MTurk Category Confusion Matrix



(b) MTurk Phrase Confusion Matrix

Figure 7.2: Both confusion matrices show predictions using the Conv-12-Pool-6 + Category-Phrase architecture, as this model achieves the highest average AUC. **(a)** shows the predictions vs. ground truth values for the category of polite speech. **(b)** shows predictions vs. ground truth values for the specific polite speech phrases.

7.6 Types of Augmentation

Upon breaking down the data used for augmentation by augmentation type, we get the results shown in Table 7.6. From these results, we can observe which types of augmentation help us improve our model and how effective each type of augmentation is. Based on the results, pitch augmentation actually seemed to hurt the model's performance. This result was surprising, as we manually checked several examples of pitch-augmented audio to confirm that they were label-preserving. However, it is possible that in some cases the augmentation is too extreme for the model to recognize. Next, we find that speed augmentation gave the best performance boost of any one augmentation type, providing comparable results to the model trained using all three types of augmentation. This seems to suggest that there is substantial value in varying the speed of audio, especially in smaller datasets with fewer examples of each phrase and fewer unique voices. Background noise augmentation also improved the model's performance over

the unaugmented dataset, as was expected. After reviewing these results, it is worth investigating in the future why pitch augmentation did not help, as well as if the model would have learned better features if only speed and background noise augmentation were used.

Table 7.6: Augmentation Type Comparison

Augmentation Type	Loss	Acc	AUC-GJ	AUC-TY	AUC-PLS	AUC-YW	AUC-AVG
No Augmentation	0.1777	0.9583	0.9873	0.9870	0.9968	0.9985	0.9924
Just Pitch	0.2348	0.9497	0.9800	0.9826	0.9995	0.9854	0.9869
Just Speed	0.0794	0.9722	**0.9974**	**0.9987**	0.9995	**0.9990**	**0.9986**
Just Background Noise	0.1184	0.9512	0.9918	0.9970	0.9991	0.9963	0.9960
Combined	**0.0758**	**0.9790**	0.9971	0.9983	**0.9996**	0.9986	0.9984

7.7 Amount of Crowdsourced Data

In our final experiment on the MTurk dataset of clean speech recordings, we compare varying amounts of training data in order to determine whether we could have crowdsourced fewer examples or if the additional examples provided improved performance on our task. Table 7.7 shows the results of training on 25%, 50%, 75% and 100% of our crowdsourcing training data, keeping the test set constant throughout. Our results show that in our case (1152 total crowdsourced recordings), the additional data consistently improved the models performance in terms of test loss, test accuracy, and every AUC metric, even though the number of unique speakers is very similar throughout. The results do suggest, however, that as the amount of data increases, the improvements by adding more data will diminish as a ceiling is reached.

Table 7.7: Data Quantity Comparison

Data Quantity	Loss	Acc	AUC-GJ	AUC-TY	AUC-PLS	AUC-YW	AUC-AVG
25%	0.5015	0.8506	0.9391	0.9452	0.9588	0.9857	0.9572
50%	0.3259	0.9174	0.9654	0.9793	0.9722	0.9874	0.9761
75%	0.1890	0.9444	0.9896	0.9928	0.9946	0.9965	0.9934
100%	**0.0758**	**0.9790**	**0.9971**	**0.9983**	**0.9996**	**0.9986**	**0.9984**

Chapter 8

Handling Class Imbalance: YouTube Dataset

In this research, as is often the case in real-world machine learning problems, we faced the challenge of class imbalance with a relatively small overall dataset size (Table 8.1). In our case, after crowdsourcing polite language labels for our YouTube dataset, we discovered that there were very few examples from the classes of polite language other than *Good job*. Furthermore, we had many more examples of audio that did not contain polite language than audio that did. Thus, it was a challenge to encourage the model to learn a policy more useful than always choosing the dominant class (i.e. audio without polite language). To combat class imbalance, and address Research Question 2, we compare approaches that include downsampling, upsampling, multitask formulation, and data augmentation.

	Good job	Thank you	Please	You're welcome	Other
Training	2096	182	121	8	12036
Evaluation	700	57	60	4	4233

Table 8.1: YouTube Dataset Composition

8.1 Baseline

For the speech recognition component of our project, we use Google Cloud Speech as the primary baseline. To use Google Cloud Speech for polite language detection, we first acquired the complete transcriptions provided by the Google API using the specialized "video" model that is intended for transcribing videos with multiple speakers and background noise. Then, we performed a search for phrases indicative of polite language within the transcriptions.

8.2 Transfer Learning

Transfer learning is conducted by first training a model on a task that is related to the target task, but for which there typically exists a larger, more balanced dataset. Then, some of the learned weights from that model are used to initialize a model for training on the true dataset of interest. In our case, we first learn to distinguish between categories of polite language (a 4-class problem) and specific phrases (a 21-class problem) on our crowdsourced dataset, as we controlled the number of examples collected for each of the phrases that we specified (see Table 1.1). Then, we apply the weights for the convolutional feature maps and first two fully connected layers to our new model that must also discriminate between polite/non-polite language (i.e. 5-class category problem). With transfer learning, the transferred weights are sometimes frozen, only permitting the added layers to update based on the gradient. However, we found that this leads to overfitting in our model, so we leave all layers of the network as trainable.

8.3 Downsampling

Downsampling involves selecting a subset of the training data in order to reduce the training dataset size. We hypothesized that downsampling the "background noise" class until it is roughly equal in size to the *Good job* class (Table 8.2) would force the model to learn how to distinguish between the classes rather than simply labeling all examples as the majority class. The downside of this approach is that it does not make use of all of the available training data, potentially leaving out valuable training examples.

	Good job	Thank you	Please	You're welcome	Other
Training	2096	182	121	8	1479
Evaluation	700	57	60	4	4233

Table 8.2: YouTube Dataset Composition: Downsampling

8.4 Upsampling

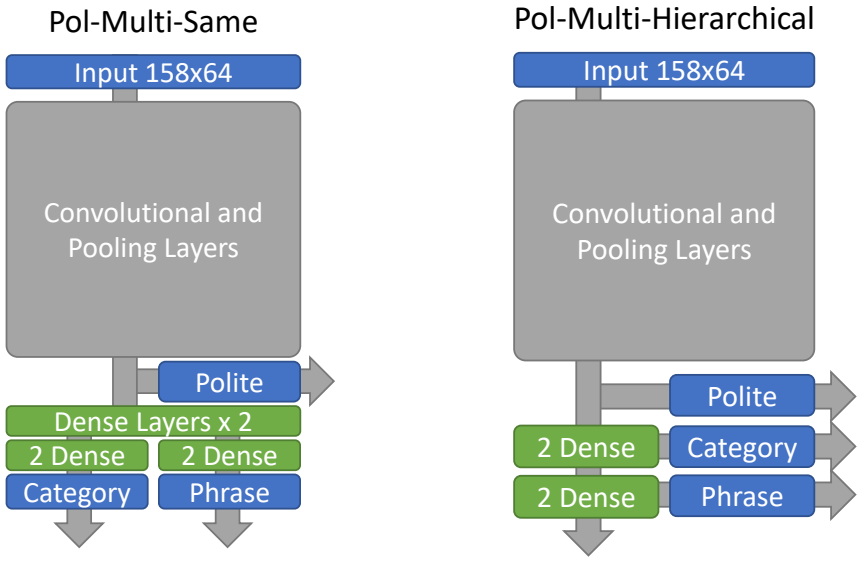
Similarly, upsampling is the process of sampling the training dataset with replacement in order to increase the training dataset size. We conjectured that upsampling the polite language classes to be roughly equal in size to the "background noise" class (Table 8.3) would help the model to better distinguish between the classes. Upsampling solves the problem of potentially leaving out useful training examples but raises the possibility of overfitting on the examples that are sampled more than once since they influence the gradient updates proportionally more than training examples that are not resampled.

	Good job	Thank you	Please	You're welcome	Other
Training	13379	1119	676	120	12036
Evaluation	700	57	60	4	4233

Table 8.3: YouTube Dataset Composition: Upsampling

8.5 Multitask Learning

While class imbalance is present when we formulate the learning task as distinguishing between 5 classes: 4 classes of polite language and non-polite language, the class imbalance is not nearly as large if we want to learn tasks such as predicting whether speech is polite or not polite, or predicting the exact phrase that is used (rather than the class of synonymous phrases). Therefore, we hypothesize that combining these three tasks into a multitask learning objective for our models will encourage them to learn robust, general-purpose features that can be used to reliably predict each of these tasks. We stagger each of these predictors (Fig. 8.1b), such that polite/not-polite prediction (most-general) occurs immediately after the convolutional layers, then two fully connected layers precede category prediction for polite language, followed by another two fully connected layers before the final phrase prediction (most-specific). We believe that this structure further encourages the development of more generalizable features in the first several layers of the model, with more specialized features added later on by interpreting the general features in different ways. In addition to the two MTL formulations found in Fig. 6.2b and Fig. 6.2c, we compare this three-task hierarchical model architecture with a traditional MTL architecture containing the binary polite task as well (Fig. 8.1a).



(a) Polite, Category and Phrase: Same

(b) Polite, Category and Phrase: Hierarchical

Figure 8.1: **(a)** shows the traditional version of the three-task MTL formulation. **(b)** shows the hierarchical version of the three-task MTL formulation.

8.6 Data Augmentation

As discussed in Chapter 3.4, data augmentation is used to perform label-preserving transformations on training examples to produce examples that vary slightly from the original, with the hope that these "augmented" examples will help the model to generalize better. In our case, we augment audio by modifying the speed at which it is played back, the audio's pitch, and by combining background noise with existing speech. Using data augmentation, we can take one example and make many different versions simply by varying the type of background noise that is added to the audio. In this way, data augmentation can be used to upsample a subset of the training dataset without simply repeating the exact same examples. Based on this intuition, we hypothesize that data augmentation would lead to improved model generalization than simply using upsampling. In the future we plan to explore data augmentation on the YouTube dataset, but for now we only apply data augmentation to the MTurk dataset of crowdsourced polite speech phrases.

Chapter 9

Results: YouTube Dataset

Before creating our own model for detecting polite language, we first tested the performance of Google Cloud Speech on our YouTube dataset, using the crowdsourced label transcriptions for each video as the ground truth labels. The results, shown in Table 9.1, indicate that even though Google has access to a much larger amount of data than we do for training their speech recognition model, we are able to outperform them (Good job AUC, Thank you AUC, and Avg. AUC) at recognizing polite speech in a noisy classroom setting.

Furthermore, as seen in Fig. 9.1, Google Cloud Speech recovers far fewer instances of polite language compared with those identified by human annotators. This error may be due to ambient noise and overlapping speech and further justifies the need for a polite language detection system robust to the challenges imposed by classroom environments.

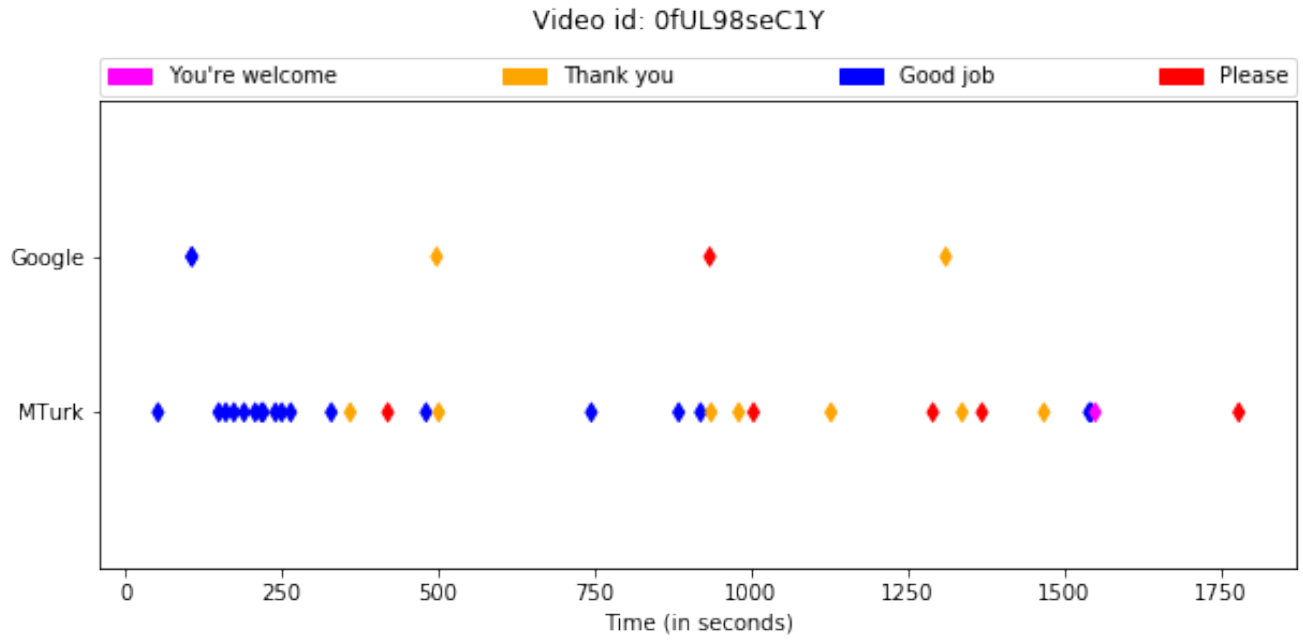


Figure 9.1: This figure compares Google Cloud Speech results with workers' annotations for one video.

To develop our custom model for speech recognition in classrooms, we apply the architecture for the model with the highest Avg. AUC on the MTurk dataset of clean speech (Conv-12-Pool 6 + Category-Phrase) to the YouTube dataset. We then compare transfer learning, downsampling, upsampling, and multitask formulations as methods for handling the inherent class imbalance and data scarcity within the YouTube dataset.

Table 9.1 shows the results for each of these experiments. "T" indicates that transfer learning was applied by using the weights learned on the MTurk dataset to initialize the weights for the 12 convolutional layers and first two fully-connected layers. "Down" signifies that downsampling of the background noise class was used while "Up" means that upsampling of the polite speech classes was used. "Multi-Same" represents the traditional MTL formulation, with both category and phrase classifications occuring at the same level within the network. Meanwhile, "Multi-Hierarchical" means that the network predicts the category of polite speech at an earlier layer than the prediction of the specific phrase. If the "Pol" modifier is used, a binary "Polite/Not polite" classifer was added immediately after the last pooling layer.

Table 9.1: Comparing Methods of Handling Class Imbalance: YouTube Dataset

Model	Loss	AUC-GJ	AUC-TY	AUC-PLS	AUC-YW	AUC-Other	AUC-AVG
Google Cloud Speech	-	0.5253	0.6599	1.0000	0.0906	0.9275	0.6407
Singletask	0.5474	0.4591	0.6164	**0.6957**	0.1823	0.4498	0.4807
T (Singletask)	0.5421	0.6037	0.4838	0.4975	**0.9207**	0.5807	0.6173
Down (Singletask)	0.9895	0.6660	0.5439	0.6119	0.3256	0.4947	0.5284
T+Down (Singletask)	0.8269	0.7205	0.6843	0.5831	0.6613	0.7334	0.6765
Up (Singletask)	0.8365	0.5873	**0.7598**	0.4300	0.5483	0.5865	0.5824
T+Up (Singletask)	0.5727	0.7633	0.5472	0.6236	0.3952	0.7596	0.6178
Up+Multi-Same	0.8249	0.5858	0.5576	0.4358	0.4373	0.5382	0.5109
T+Up+Multi-Same	0.5428	0.7791	0.6391	0.5058	0.4771	0.7391	0.6281
Up+Multi-Hierarchical	0.7988	0.5799	0.6401	0.3348	0.5000	0.5488	0.5207
T+Up+Multi-Hierarchical	**0.5191**	**0.8093**	0.7568	0.5106	0.3811	0.7777	0.6471
Up+Pol-Multi-Same	0.7543	0.5684	0.5870	0.4883	0.6726	0.5307	0.5694
T+Up+Pol-Multi-Same	0.5284	0.8064	0.6815	0.4697	0.5521	0.7792	0.6578
Up+Pol-Multi-Hierarchical	0.7222	0.4522	0.3883	0.3514	0.6264	0.4236	0.4484
T+Up+Pol-Multi-Hierarchical	0.5223	0.7928	0.7103	0.6296	0.8275	**0.7892**	**0.7499**

9.1 Transfer Learning

Immediately, it is clear that transfer learning improved the performance of the model (relative improvement of 6-67%), regardless of any additional techniques that were applied, indicating that transfer learning may be useful to combat class imbalance and data scarcity, even when the dataset for pretraining is also relatively small.

9.2 Downsampling and Upsampling

Downsampling resulted in an increased average AUC, but also increased the cross-entropy loss significantly from the basic transfer learning model. At the same time, upsampling improved performance in distinguishing between the major categories: Good job and Other. Furthermore, upsampling avoids the problem where potentially important examples are thrown out in order to balance the classes. Therefore, we choose to upsample for the remaining experiments.

9.3 Multitask Learning

We find that MTL provides a consistent benefit to the models that use transfer learning for initialization. However, MTL appears to decrease the performance of models that are not initalized using the pretrained weights from the MTurk clean speech model. Further, hierarchical MTL appears to improve AUC over traditional MTL when transfer-learning is applied, and the addition of a binary polite speech/not polite speech classifier appears to provide further benefit by forcing the model to predict in a coarse-to-fine fashion. Therefore, it is possible that multitask learning can be useful even when tasks are similar to the point where they could be considered subtasks. This supports our intuition that if the subtasks have different distributions of classes, then the harmful effect of class imbalance might be mitigated.

9.4 Discussion

Finally, we acknowledge that these results are recent and we have not had time to fully explore or understand them in their entirety. Moreover, these results are all "optimal" in the sense that the models were optimized directly on the test set rather than though a separate validation set. We also only train each model once, so we are unable to account for the variability that occurs based on weight initialization; if we noticed that the initialization was so poor that the cross entropy on the test set never decreased, we simply reinitialized the model and restarted the training process. Therefore, further experimentation is required before we can identify whether these techniques will be generalizable and useful outside of their current context.

Fig. 9.2 shows the effects of class imbalance that persist despite our attempts to combat it. For example, the minority polite speech classes are almost always predicted incorrectly, while the "Good job" and "Background noise" classes are correctly predicted at a much higher rate.

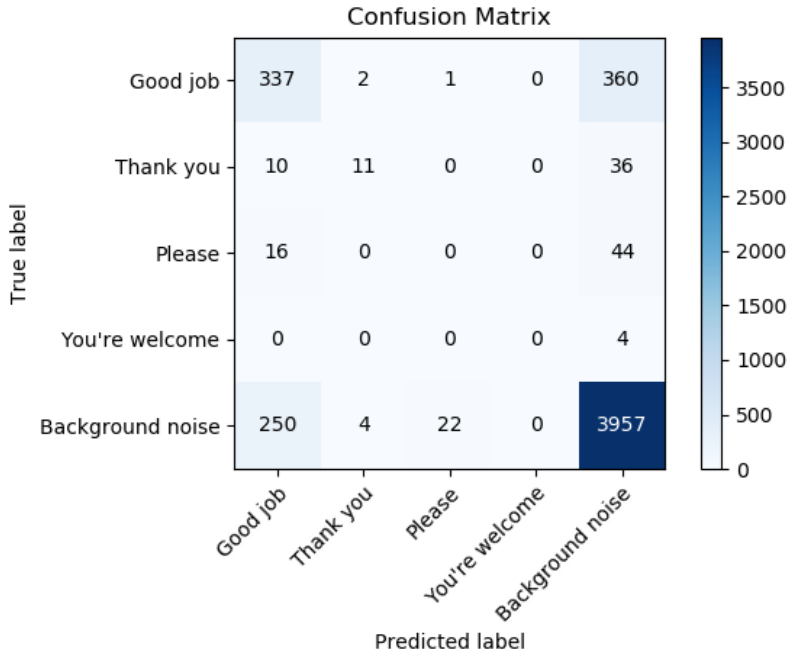


Figure 9.2: YouTube Category Confusion Matrix

Figure 9.3: This confusion matrix shows category predictions vs. ground truth values using the T+Up+Pol-Multi-Hierarchical architecture, as this model achieves the highest average AUC.

Chapter 10

Results: CLASS Dataset

To determine the correlation between polite language probabilities and CLASS scores, and therefore address Research Question 4, we consider three main approaches:

1. Calculate average polite language probabilities over the entire segment of CLASS-coded video
2. Count the number of predicted polite language instances within the video segment
3. Count the number of polite language probabilities that exceed some threshold

We chose to use the first approach, as it simply uses the average probability over the entire video segment, rather than relying on hand-picked thresholds. By employing the averaging approach to aggregate the polite language detection model's outputs for each video, we then calculate the Pearson correlation with each CLASS dimension. We also look at the correlation between specific types of polite language and each CLASS dimension.

10.1 Polite Language Prediction

Using our best model fine-tuned on the YouTube dataset of noisy, classroom speech, we compute polite language probabilities on our dataset of 106 CLASS-labeled videos. Then, we select the 5 moments with the highest probabilities of polite speech and the 5 moments with the lowest probabilities of polite speech from each video, imposing the constraint that these moments may not be within 2 seconds of each other to prevent looking at the same exact utterance multiple times. Next, we listened to each of these 1060 moments and manually labeled them as polite speech or not polite speech. In this manner, we are able to calculate an AUC for the performance of our polite speech detectors on our target dataset, the CLASS labeled videos. We find that this AUC is 0.7568, conditioned on choosing the top 5 and lowest 5 probabilities from each video as previously mentioned.

While, in fact, this number is likely optimistic, it does suggest that the features pre-trained on the clean speech dataset and fine-tuned on the YouTube classroom dataset are applicable to the CLASS dataset.

During this manual inspection process, we make a few observations about potential cues that the model is recognizing. First, as several of our polite language phrases contain multiple words, it seems that our model will predict a high polite language probability even if only one word from the phrase is captured in the audio, i.e. "It" from "That's it!" or "No" from "No problem". This could occur because we time shift each audio example slightly in either direction, and some examples could potentially be partially cut off. It could also just be that in our MTurk dataset, there are no other phrases with the same words, so "No" is good enough to distinguish between "No problem" and any other phrase. In the future, we will take more careful consideration when developing the set of phrases that we hope to detect.

We also notice that many names are predicted with high probability of polite speech. This could be in part due to the tendency to say names directly proceeding phrases indicative of praise, i.e. "Good job, [name]!". Or, in some cases, it may be that the name is similar to part of one of our polite speech phrases, i.e. John, Joey are similar to the "Job" in "Good job", and Grayson, Grant are similar to "Great".

Lastly, we find that behaviors such as clapping and laughter are commonly predicted with high probabilities of polite speech, even if no polite speech is present. Perhaps these events occur together often in the YouTube dataset, and as such, the model learns to predict these events as more likely to be polite speech.

10.2 Polite Language Visualization

Another product of our polite language model that is potentially useful for teachers is the ability to visualize polite language use over time. For instance, Fig. 10.1 shows the predicted polite language probabilities for each of our four classes of polite language over a 15-minute CLASS video segment. Teachers and CLASS-coders could use this type of visualization to more quickly find interesting interactions within the videos and potentially discover ways to replicate those positive interactions more often in the future.

10.3 CLASS Score Evaluation Metric

For CLASS score prediction, we will use Pearson (r) correlations and two-tailed p-values to compare with the results of [3] and determine if the correlation between polite language and CLASS score for a particular dimension is statistically significant.

10.4 CLASS Score Correlation

After developing our approaches for polite language detection, we evaluate the correlation of polite language with CLASS scores for Positive and Negative climate and compare with the results obtained by Ramakrishnan et al. [3] (Table 10.1).

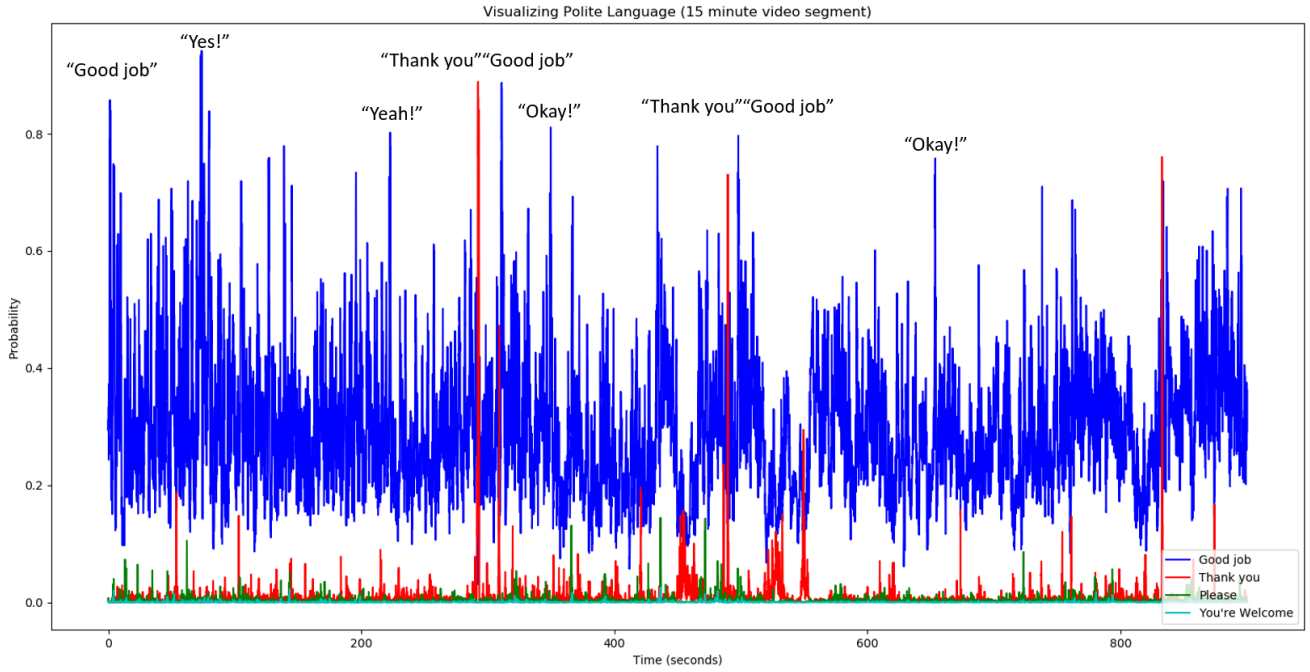


Figure 10.1: This visualization shows the probabilities for each of our four classes of polite language over a 15-minute CLASS video. The annotations indicate the actual phrases that are said at those points in time.

Table 10.1: Previous CLASS Correlation [3]

Model	Positive Climate		Negative Climate	
	r	p	r	p
CNN (Audio) [3]	0.308	0.005	0.290	0.003
Bi-LSTM (Smile) [3]	0.17	0.052	0.240	0.004
Ensemble [3]	0.381	0.002	0.456	0.0005

Then, we also look at correlations between polite language (and each specific type of polite language) and each CLASS dimension, as seen in Table 10.2. An asterisk (*) indicates statistical significance at the 0.05 level or below. To our surprise, we found no significant correlation between polite speech and positive or negative climate. After discussing this with our collaborators, they suggested that this could be due to the interactions that occur between the teacher and the student when polite speech is occurring. For example, in one CLASS video, a teacher is saying "Thank you" (an instance of polite speech) while pulling a student off of a table (a negative action).

We do find a statistically-significant positive correlation between instances of "You're welcome" and Regard for Child Perspectives. Meanwhile, instances of "Please" and "You're welcome" are negatively correlated with Facilitation of Learning and Development, Quality of Feedback, and Language Modeling. Perhaps this is because teachers use "Please" in an attempt to redirect students' attention from off-task activities back to the class activities, or perhaps the teachers use these polite speech phrases in a sarcastic tone of voice. It is also possible that these correlations are a product of erroneous predictions by our model. Further investigation is necessary to understand these correlations and incorporate them into an ensemble predictor of CLASS scores, but we ultimately do find

Table 10.2: CLASS Dimension Correlations

	Positive Climate	Negative Climate	Teacher Sensitivity	Regard for Child Perspectives	Behavior Guidance	Facilitation of Learning and Development	Quality of Feedback	Language Modeling
Polite Language (Aggregated)	-0.098	0.042	-0.014	0.017	0.101	-0.063	0.007	-0.047
Good job	-0.099	0.038	-0.017	0.006	0.094	-0.048	0.026	-0.040
Thank you	0.063	0.036	0.030	0.100	0.025	-0.055	-0.114	0.026
Please	-0.085	-0.006	-0.003	0.019	0.036	-0.242*	-0.245*	-0.239*
You're welcome	-0.019	-0.033	0.094	0.162*	0.123	-0.167*	-0.160*	-0.153*

that polite speech is related to several CLASS dimensions. In the future, we will compare the performance of Ramakrishnan et al.'s best model [3] with a new ensemble approach that extends that model with polite language features to determine whether polite language detection can be used to further improve CLASS score prediction.

Chapter 11

Social Implications & Biases

As we ran two crowdsourcing tasks and are designing a tool to be used in classrooms with young children, it is important for us to consider the ethics of our actions and the implications of the tool that we design. First, we must consider how much we pay the workers that completed our two Human Intelligence Tasks (HITs), as this typically affects the quantity of data that can be collected, but it also determines the quality of the work done for the HIT since a worker is more likely to do a thorough job completing a HIT that they feel is worth the money. Then, we also consider how our model will impact teachers and students.

For the speech event labeling task that we submitted to Mechanical Turk, workers were paid $3.50 per hour based on the length of the video that they were asked to label for polite speech events. In retrospect, we observe that the quality of the labels produced by any one worker is relatively low on average. For example, of the 968 speech events that were originally annotated by workers on Mechanical Turk, 154 of these were found not to be polite speech. Therefore, for similar tasks run in the future, we recommend offering workers a higher reward to incentivize them to provide higher quality labels, especially given the time-consuming nature of one such task.

Meanwhile, for our clean speech collection HIT, we paid workers $0.10 per phrase that they recorded and submitted. This task was much less time consuming and required less than a minute for workers to complete, especially if they had already read the instructions and were completing another version of the HIT with a different phrase (as was commonly done). We find that the quality of speech recordings collected from this HIT was very high, and there were very few audio files uploaded that do not contain speech of the desired phrase. Therefore, we believe that this reward was adequate for the type of task that we ran.

Regarding the impact of our proposed model on students, we believe that there are many potential benefits and very little risk. By designing our own specialized speech recognition model, we ensure that the students' interactions with their teacher are not shared over the internet, as they would be if we were to use a commercial speech recognition application such as Google Cloud Speech. Further, students' desire to learn and attend class could improve if their teacher receives more feedback on their teaching and replicates interactions that they see

worked well in the past.

Similarly, this research is designed for teachers to have more access to feedback on their teaching and unbiased self-evaluation. Although CLASS could be used by a school administration to evaluate teachers and use this evaluation for hiring or firing employees, this is not the purpose for which we conducted this research. Moreover, we recommend that our model not be used in this manner because it is prone to error, and the produced polite language probabilities are only weakly correlated with CLASS scores.

Lastly, we acknowledge that there are some key biases in our detector of polite language. To begin with, our detector is trained primarily on examples of polite speech that are spoken by adults, so the model may possibly detect polite speech from adults more accurately than polite speech from children. However, for polite speech detection in toddler classrooms, this is not generally a problem since we observe that toddlers do not generally use a lot of polite language.

Finally, the concept of politeness is culturally dependent. Our study is about toddlers in American preschools and measuring the CLASS scores for those classrooms. The CLASS manual defines polite language as a behavioral indicator of positive climate. We agreed with our collaborator, who is an expert in classroom observation, on a set of key phrases that we define as *polite speech*. Our research question is whether our detectors can predict CLASS scores. If so, this may have implications for videos that are *similar to those in our dataset*. The predictive power of our system may differ for videos sampled from a different population of schools.

Chapter 12

Conclusion

In our research, we make several contributions that extend beyond the highly specialized application of detecting polite language in classroom videos. To begin with, **(1)** we design a custom crowdsourcing task for efficiently labeling videos for speech events. This task can be repurposed to gather labels from YouTube videos for any desired speech or contextual events. Secondly, **(2)** we design a custom crowdsourcing task that facilitates the simple collection of speech recordings. From our experience with these crowdsourcing tasks, **(3)** we offer insight into the reliability of crowdsourcing labels for speech events and the crowdsourcing of short speech recordings, showing that the latter tends to be reliable, as it is a quick and easy task, while the former is more time-consuming and should offer a higher reward for more reliable speech event labels. Next, **(4)** we present strategies for successfully training convolutional models to detect spoken key phrases robust to noisy environments even with high data scarcity and class imbalance. We find that, at least in these preliminary experiments, further temporal pooling, transfer learning, hierarchical multitask learning, and upsampling all seemed to improve the performance of our speech detection task when class imbalance was present and data was scarce. Additionally, we find that warping the speed of audio and adding background noise are useful methods for expanding the training dataset. Lastly, **(5)** we find that polite speech is significantly correlated with multiple dimensions of CLASS, demonstrating its potential as a feature for more accurate CLASS score prediction.

The research conducted thus far leaves many avenues open for exploration. For instance, several of the model variations that we experiment with on our dataset of clean speech are not yet fully understood. In the future, we would like to conduct more rigorous experiments to determine whether the changes we made in model architecture, especially our approaches to multitask learning and our further temporal pooling, are significant and generalizable to other datasets.

Moreover, from a pedagogical standpoint, with this detector of polite language, we really only capture one component of a larger interaction between teacher and student. If we were able to understand more about the entire teacher-student interaction at these moments when polite language is detected, then we might be able to find

a stronger correlation with CLASS score and help to automatically interpret and identify even more interesting interactions.

Additionally, we currently do not explicitly take into account the tone of voice or context of the polite speech. Therefore, a potential area of exploration is incorporating the tone or sincerity of speech into the model, as this might enable the model to better discriminate between polite language phrases used in a polite context and polite language phrases used outside of a polite context. This too could potentially improve the quality of the feedback signals that we provide teachers and CLASS coders.

In conclusion, we have demonstrated that polite language is useful as a predictor of several dimensions of CLASS, and we propose several architectural changes to convolutional speech recognition models that warrant further investigation.

Bibliography

[1] K. M. La Paro, A. C. Williamson, and B. Hatfield, "Assessing Quality in Toddler Classrooms Using the CLASS-Toddler and the ITERS-R," *Early Education and Development*, vol. 25, pp. 875–893, 8 2014.

[2] A. M. Aung, A. Ramakrishnan, and J. Whitehill, "Who are they looking at? Automatic Eye Gaze Following for Classroom Observation Video Analysis," *EDM*, 2018.

[3] A. Ramakrishnan, E. Ottmar, J. Locasale-crouch, and J. Whitehill, "Toward Automated Classroom Observation : Predicting Positive and Negative Climate," *IEEE Conference on Automatic Face and Gesture Recognition*, 2019.

[4] R. O'conner, J. De Feyter, A. Carr, J. L. Luo, and H. Romm, "A review of the literature on social and emotional learning for students ages 38: Teacher and classroom strategies that contribute to social and emotional learning (part 3 of 4)," tech. rep., Institute of Education Sciences.

[5] K. M. La Paro, B. K. Hamre, and R. C. Pianta, *Classroom Assessment Scoring System (CLASS) Manual, Toddler*. 2012.

[6] Z. Zhang, J. Geiger, J. Pohjalainen, A. E.-D. Mousa, W. Jin, and B. Schuller, "Deep Learning for Environmentally Robust Speech Recognition," *ACM Transactions on Intelligent Systems and Technology*, vol. 9, pp. 1–28, 4 2018.

[7] G. Trigeorgis, F. Ringeval, R. Brueckner, E. Marchi, M. A. Nicolaou, B. Schuller, and S. Zafeiriou, "Adieu features? End-to-end speech emotion recognition using a deep convolutional recurrent network," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5200–5204, IEEE, 3 2016.

[8] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, Long Short-Term Memory, Fully Connected Deep Neural Networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP15)*, pp. 4580–4584, 2015.

[9] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, G. Chen, J. Chen, J. Chen, Z. Chen, M. Chrzanowski, A. Coates, G. Diamos, K. Ding, N. Du,

E. Elsen, J. Engel, W. Fang, L. Fan, C. Fougner, L. Gao, C. Gong, A. Hannun, T. Han, L. Vaino Johannes, B. Jiang, C. Ju, B. Jun, P. LeGresley, L. Lin, J. Liu, Y. Liu, W. Li, X. Li, D. Ma, S. Narang, A. Ng, S. Ozair, Y. Peng, R. Prenger, S. Qian, Z. Quan, J. Raiman, V. Rao, S. Satheesh, D. Seetapun, S. Sengupta, K. Srinet, A. Sriram, H. Tang, L. Tang, C. Wang, J. Wang, K. Wang, Y. Wang, Z. Wang, Z. Wang, S. Wu, L. Wei, B. Xiao, W. Xie, Y. Xie, D. Yogatama, B. Yuan, J. Zhan, and Z. Zhu, "Deep Speech 2 : End-to-End Speech Recognition in English and Mandarin," in *International Conference on Machine Learning (ICML16)*, pp. 173–182, 2016.

[10] Y. Qian, M. Bi, T. Tan, and K. Yu, "Very Deep Convolutional Neural Networks for Noise Robust Speech Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, pp. 2263–2276, 12 2016.

[11] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *IEEE Signal Processing Magazine*, vol. 29, pp. 82–97, 11 2012.

[12] B. Logan, "Mel Frequency Cepstral Coefficients for Music Modeling," *ISMIR*, 2000.

[13] L. R. Rabiner and B. H. B.-H. Juang, *Fundamentals of speech recognition*. PTR Prentice Hall, 1993.

[14] A.-r. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic Modeling Using Deep Belief Networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, pp. 14–22, 1 2012.

[15] T. N. Sainath, A.-r. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8614–8618, IEEE, 5 2013.

[16] M. Bi, Y. Qian, and K. Yu, "Very Deep Convolutional Neural Networks for LVCSR," in *Interspeech*, pp. 3259–3263, 2015.

[17] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *ICLR*, 2015.

[18] T. Sercu, C. Puhrsch, B. Kingsbury, and Y. LeCun, "Very Deep Multilingual Convolutional Neural Networks for LVCSR," *arXiv:1509.08967*, 9 2015.

[19] Y. Wang, X. Deng, S. Pu, and Z. Huang, "Residual Convolutional CTC Networks for Automatic Speech Recognition," *arXiv:1702.07793*, 2 2017.

[20] T. Tan, Y. Qian, H. Hu, Y. Zhou, W. Ding, and K. Yu, "Adaptive Very Deep Convolutional Residual Network for Noise Robust Speech Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, pp. 1393–1405, 8 2018.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *arXiv:1512.03385*, 12 2015.

[22] W. Li, L. Wang, Y. Zhou, J. Dines, M. Magimai.-Doss, H. Bourlard, and Q. Liao, "Feature Mapping of Multiple Beamformed Sources for Robust Overlapping Speech Recognition Using a Microphone Array," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, pp. 2244–2255, 12 2014.

[23] C. Liu, N. Inoue, and K. Shinoda, "A unified network for multi-speaker speech recognition with multi-channel recordings," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1304–1307, IEEE, 12 2017.

[24] M. Suzuki, G. Kurata, T. Nagano, and R. Tachibana, "Speech recognition robust against speech overlapping in monaural recordings of telephone conversations," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5685–5689, IEEE, 3 2016.

[25] Z. Chen and J. Droppo, "Sequence Modeling in Unsupervised Single-Channel Overlapped Speech Recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4809–4813, IEEE, 4 2018.

[26] R. Caruana, "Multitask Learning," *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.

[27] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-Stitch Networks for Multi-task Learning," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3994–4003, IEEE, 6 2016.

[28] A. Jain, M. Upreti, and P. Jyothi, "Improved Accented Speech Recognition Using Accent Embeddings and Multi-task Learning," in *Interspeech*, pp. 2454–2458, 2018.

[29] N. Kyun Kim, J. Lee, H. Kyu Ha, G. Woo Lee, J. Hyuk Lee, and H. Kook Kim, "Speech Emotion Recognition Based on Multi-Task Learning Using a Convolutional Neural Network," in *APSIPA*, 2017.

[30] K. Krishna, S. Toshniwal, and K. Livescu, "Hierarchical Multitask Learning for CTC-based Speech Recognition," *arXiv:1807.06234*, 7 2018.

[31] N. Jaitly and G. E. Hinton, "Vocal Tract Length Perturbation (VTLP) improves speech recognition," in *International Conference on Machine Learning (ICML) Workshop on Deep Learning for Audio, Speech, and Language Processing*, vol. 28, 2013.

[32] X. Cui, V. Goel, and B. Kingsbury, "Data Augmentation for Deep Neural Network Acoustic Modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, pp. 1469–1477, 9 2015.

[33] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Interspeech*, 2015.

[34] R. Gomez and K. Nakamura, "Exploring data augmentation methods in reverberant human-robot voice communication," in *RO-MAN 2017 - 26th IEEE International Symposium on Robot and Human Interactive Communication*, vol. 2017-Janua, pp. 1154–1158, IEEE, 8 2017.

[35] R. Scheibler, E. Bezzam, and I. Dokmanic, "Pyroomacoustics: A Python Package for Audio Room Simulation and Array Processing Algorithms," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 351–355, IEEE, 4 2018.

[36] M. Ferras, S. Madikeri, P. Motlicek, S. Dey, and H. Bourlard, "A Large-Scale Open-Source Acoustic Simulator for Speaker Recognition," *IEEE Signal Processing Letters*, vol. 23, pp. 527–531, 4 2016.

[37] L. Lee and R. Rose, "A Frequency Warping Approach to Speaker Normalization," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 1, p. 49, 1998.

[38] Z. Wang, X. Pan, K. F. Miller, and K. S. Cortina, "Automatic classification of activities in classroom discourse," *Computers & Education*, vol. 78, pp. 115–123, 9 2014.

[39] P. J. Donnelly, N. Blanchard, B. Samei, A. M. Olney, X. Sun, B. Ward, S. Kelly, M. Nystrand, and S. K. D'Mello, "Multi-sensor modeling of teacher instructional segments in live classrooms," in *Proceedings of the 18th ACM International Conference on Multimodal Interaction - ICMI 2016*, (New York, New York, USA), pp. 177–184, ACM Press, 2016.

[40] S. K. D'Mello, A. M. Olney, N. Blanchard, B. Samei, X. Sun, B. Ward, and S. Kelly, "Multimodal Capture of Teacher-Student Interactions for Automated Dialogic Analysis in Live Classrooms," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction - ICMI '15*, (New York, New York, USA), pp. 557–566, ACM Press, 2015.

[41] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *arXiv:1502.03167*, 2 2015.

[42] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, Association for Computing Machinery, 2010.

[43] M. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, and G. Hinton, "On rectified linear units for speech processing," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3517–3521, IEEE, 5 2013.

[44] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[45] A. Vaswani, G. Brain, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, . Kaiser, and I. Polosukhin, "Attention Is All You Need," in *31st Conference on Neural Information Processing Systems*, 2017.