NOISE-ROBUST KEY-PHRASE DETECTORS FOR AUTOMATED CLASSROOM FEEDBACK

*Brian Zylich and Jacob Whitehill**

Department of Computer Science, Worcester Polytechnic Institute, MA, USA

ABSTRACT

With the goal of giving teachers automated feedback about their classrooms, we investigate how to train automatic speech detectors of key phrases such as *good job*, *thank you*, *please*, and *you're welcome*. This kind of language conveys support and respect from teacher to student and is one of the behavioral markers used in the established CLASS [1] classroom observation protocol. School classrooms are noisy and contain overlapping speech, presenting a highly challenging environment for automatic speech recognition (ASR), even for state-of-the-art approaches. We train deep neural networks using hierarchical multitask learning (MTL) on a modest-sized but highly-tailored dataset of classroom speech. Compared to 2 state-of-the-art ASR systems for general-purpose speech recognition (Google [2] and DeepSpeech [3]), our system delivers a substantially improved recall rate (50.4% versus 20.5%) while matching their precision (30%). Moreover, our system's predictions correlate with several dimensions of the CLASS.

***Index Terms*—** education, speech recognition, deep learning, multitask learning

1. INTRODUCTION

In school classrooms, the interpersonal interactions between students and teachers impact students' long-term cognitive and emotional development [4, 5]. It is thus important that teachers receive frequent feedback about how well they are giving children the support they need. One common form of feedback is *classroom observation*, whereby a more experienced colleague or administrator watches live or video-recorded sessions of a classroom and rates it along different dimensions. The Classroom Assessment Scoring System (CLASS) is one of the most commonly used protocols [1, 4].

While classroom observation and CLASS coding in particular has shown to be effective for honing teachers' skills [6], its scalability is limited by the lack of trained coders and the labor involved in coding. By automating CLASS feedback, more teachers could get access to this valuable tool, and feedback could be given more frequently. In this paper, we tackle one specific aspect of this challenge: how to recognize *supportive* and *respectful* language spoken by teachers to students in the specific context of toddler classrooms (2-3 year old children). Utterances such as *good job*, *thank you*, *please*, and *you're welcome* constitute some of the behavioral markers that classroom observation coders attend to during coding [1]. This task is highly challenging due to the nature of classroom environments: multiple children may be crying, singing, shouting, or speaking simultaneously while teachers are attempting to give instructions or retain control. Classroom speech is qualitatively different from typical ASR settings, which often contain minimal background noise or overlapping speech, such as automatically providing subtitles for a commentator. Thus, even state-of-the-art ASR systems such as Google Cloud Speech – which are presumably trained on hundreds of thousands of speakers – often struggle to interpret classroom speech accurately.

In this paper, we explore deep learning approaches to recognize 21 different phrases associated with 4 categories (*Good job*, *Thank you*, *Please*, and *You're welcome*) related to supportive and respectful language in toddler classrooms. In particular, we investigate how to train such models on small-scale but highly-tailored datasets of school classrooms. Moreover, we compare different architectures that take more or less advantage of the hierarchical structure of the labels we are trying to estimate (specific phrase, category, and binary supportive/non-supportive). We compare our custom models to two state-of-the-art speech recognition engines that are presumably trained on massive datasets and find that our model gives higher accuracy despite being trained on far fewer training data. Finally, using a CLASS-coded dataset of toddler classroom observations, we assess the extent that our speech recognition network can extract information automatically that is correlated with different CLASS dimensions.

Related Work: Automated classroom analysis has begun to interest the ASR, affective computing, and educational data-mining communities over the past few years. Prior works have used machine learning for audio-based classroom activity detection [7, 8, 9], vision-based gesture detection [10, 11, 12], and automated classroom feedback for teachers [13, 14]. Ramakrishnan et al. [13], in particular, use deep learning on both visual and auditory data to create ensemble predictors for the Positive Climate and Negative Climate dimensions of the CLASS. Their system predicts CLASS scores directly

*This material is based upon work supported by the National Science Foundation Graduate Research Fellowship under Grant No. 1938059, and also by NSF Cyberlearning grants 1822768, 1551594, and 1822830.

from low-level MFCC features without detecting intermediate higher-level language features. Hence, their model is possibly detecting patterns of noise in the classroom (e.g., loud noise in the classroom may suggest the teacher is not creating a positive climate). In contrast, the system that we propose explicitly detects events of *supportive language*. These events may provide semantically higher-level information than an MFCC-classifier conceivably could. Moreover, the output of our system is more easily interpretable by teachers.

2. DATASETS

We train and evaluate our neural networks for supportive speech detection on two highly-tailored datasets: classroom videos on YouTube, and crowdsourced recordings of supportive speech from Mechanical Turk. We also conduct an exploratory analysis, using our trained detector on a dataset [13] collected at the University of Virginia (UVA), of how supportive speech events are related to CLASS dimensions.

2.1. YouTube Dataset of Classroom Videos

We harvest a dataset of 57 publicly available classroom videos [13] from YouTube as training data for our speech detector. These videos show real classrooms of young children and contain significant overlapping speech and ambient noise. As they are public, we crowdsource labels for supportive speech events using Amazon Mechanical Turk. In particular, we ask workers on Mechanical Turk (3 per video) to watch the 57 YouTube videos and identify *when* a supportive speech event occurs as well as *which category* each event belongs to: **Good job** (e.g. "Good job", "Great", "Awesome", etc.), **Thank you** (e.g. "Thank you", "Thanks", etc.), **Please**, and **You're welcome** (e.g. "You're welcome", "No problem", etc.). Labelers use a custom annotation tool that we build for the task (Fig. 1).

To reconcile the different labels across the different workers for each video, we compute the union of all 3 event sets and then apply the following heuristic: If 2+ consecutive labels (ordered by timestamp) are within 2s of each other and from the same category, then they are merged and their timestamps averaged. In this way, we obtain 703 instances of *Good job*, 62 of *Thank you*, 46 of *Please*, and 3 of *You're welcome*. The dataset is partitioned (over videos) into train (49 videos) and test (8 videos) subsets.

2.2. Crowdsourcing Supportive Speech Recordings

Due to the rarity of some supportive speech phrases in the YouTube videos, we choose to collect a secondary dataset of utterances by asking workers on Mechanical Turk to record their own voice. Using another custom tool, we collect ~60 recordings for each of 21 different supportive speech phrases (16 versions of *Good job*, 2 versions of *Thank you*, 1 version of *Please*, and 2 versions of *You're welcome*). These

Fig. 1: Annotation tool to label supportive speech events.

recordings are made by 62 unique adult speakers, typically with minimal background noise or overlapping speech. As part of the instructions of the task, workers are first given examples of both *what* to say and *how* to say it.

To increase data variability, we apply data augmentation based on pitch and speed [15], as well as background-noise. For pitch augmentation, we adjust the tempo to 0.75x and 1.33x the original, changing the sampling rate accordingly to prevent changing the speed. For speed augmentation, we use tempos of 0.9x and 1.1x the original, keeping the sampling rate constant. For background augmentation, we add the clean speech recordings to randomly selected clips from the YouTube dataset without any supportive speech.

2.3. UVA Dataset of CLASS-coded Videos

As our long-term goal is to partially automate classroom observation coding, we use a CLASS-coded video dataset to assess whether automatically detected supportive speech events are predictive of specific CLASS dimensions (e.g., positive climate, language modeling, etc.). In particular, we evaluate our trained system on a dataset of 131 CLASS-coded videos (~100 hours total) collected at UVA. The videos take place in American preschool classrooms and typically contain 1-2 teachers interacting with 8-10 young children, all of whom may be speaking, yelling, crying, or singing simultaneously. These videos are usually 45-60 minutes in length and are split into 15-minute segments. Each of 238 segments is labeled with a numeric value in the range of 1-7 for each of the eight dimensions defined by the CLASS Toddler protocol.

3. SPEECH DETECTOR NETWORK

Our speech detector (Fig. 2) is based on the network design by [16]. It uses non-overlapping pooling layers and padding in both the temporal and feature dimensions in each convolutional layer. Each convolutional and fully-connected layer (FC) is followed by batch normalization and ReLU activation. Our system detects supportive speech by splitting an audio clip into frames and classifying each frame independently.

The detector is run on only those portions of the audio in which speech is detected using the WebRTC Voice Activity Detector (https://github.com/wiseman/py-webrtcvad). The frame length is set to 1.6s, which is the longest utterance length of a random sample of 25 supportive speech utterances from the YouTube data. Due to the longer window length compared to [16], we increase the network depth from 10 to 12 convolutional layers with an extra temporal pooling layer.

Feature extraction: Each audio clip is converted to mono-channel and re-sampled at 48KHz. Then, 64-dimension FBANK features are extracted using a time window of 25ms and time step of 10ms. Feature vectors are concatenated over 158 time windows to yield a frame length of 1.6s, and frames are extracted every 100ms. Each 158x64 feature vector is fed to the network and classified independently.

Attention: We add an attention layer after the last max-pool and before the first FC layer. This layer computes attention weights that sum to 1 (using softmax), which are then multiplied by the output of the max-pool.

Hierarchical MTL: Finally, we use hierarchical multi-task learning (MTL) to capture the relationships between phrases, categories, and supportive speech: The model first predicts whether the audio contains supportive speech or not. Then, a second prediction determines the category (including one extra category for "non-supportive"). Last, the model predicts the specific phrase (including one extra option for "non-supportive"). This architecture is inspired by prior use of MTL for ASR [17, 18] and emotion recognition [19]. Kyun Kim et al. suggest that MTL is beneficial when data is scarce [19], and Krishna et al. suggest that hierarchical MTL outperforms standard MTL with large datasets [18]. Thus, we compare our model, seen in Fig. 2 (a), to several other designs: (1) Singletask: Category-level prediction is the only objective. (2) MTL-SL: The category- and phrase-level objectives are at the *same level*. (3) MTL-H: Category-level predictions are made before phrase-level predictions, imposing a general-to-specific *hierarchy*. (4) MTL-S-SL and (5) MTL-S-H: A binary objective is added to MTL-SL and MTL-H, determining whether the speech is a *supportive* phrase.

Training: We pretrain our network (Adam optimizer, 25 epochs, lr=0.001, batch size=64) on all the crowdsourced recordings (Sec. 2.2). Since all these recordings are examples of "supportive speech", no loss is backpropagated through the "supportive" network output at this stage. Next, we fine-tune the network on the YouTube training data (Adam optimizer, 15 epochs, lr=0.001, batch size=64). Negative examples (i.e., non-supportive speech) are harvested in each clip by finding moments with speech (according to WebRTC detector) but with no labeled *supportive* speech.

4. EXPERIMENTS

We conduct experiments to answer 4 research questions (**RQs**): (1) Can supplementary audio data of human speakers mimicking school classrooms be used to improve recognition accuracy in real classrooms? (2) Does hierarchical MTL improve classification accuracy? (3) How does a neural network for supportive speech classification that was trained on a small but highly tailored dataset compare, in terms of accuracy, with general-purpose ASR engines trained on huge datasets? (4) How do the trained network's outputs relate to human-coded CLASS scores on toddler classroom videos?

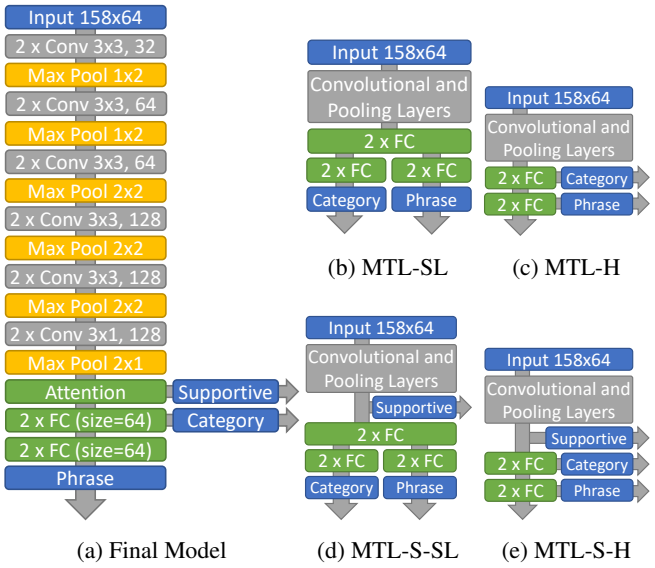

Fig. 2: **(a)** Our final network to detect *supportive speech* categories. **MTL Variants: (b)** Category & phrase predictions occur at same level. **(c)** Category predicted before phrase, hierarchically. **(d)** Binary *supportive* prediction before MTL-SL. **(e)** Binary *supportive* prediction before MTL-H.

Training: We train our networks as described in Sec. 3. Each network is trained 5x from random restarts, and the accuracies are then averaged. For RQ #1, pre-training on crowdsourced recordings is skipped, and the network is trained only on the YouTube dataset.

Metrics: Classification accuracy is always assessed on the YouTube test set. For MTL experiments, we compute Area Under the ROC Curve (AUC) for each of the 5 categories of speech (*Good job* (GJ), *Thank you* (TY), *Please* (PLS), *You're welcome* (YW), and non-supportive) in a 1-v-other manner. For comparison to existing ASR engines (see below), which do not provide probabilistic predictions for all classes, we use precision, recall, and F1 instead of AUC.

ASR baselines: For RQ #3, we compare our system to two state-of-the-art ASR engines: Google Cloud Speech [2] (in "video" mode for better performance), and Project DeepSpeech [3]. Using these tools, we generate transcriptions with word-level timestamps for each video in the test set and compute accuracy between transcripts and YouTube test labels.

Prediction of CLASS scores: For RQ #4, we compute probabilistic predictions of the 4 supportive speech categories

Model	GJ	TY	PLS	YW	Other	Avg
Singletask	0.763	0.547	0.624	0.395	0.760	0.618
MTL-SL	0.779	0.639	0.506	0.477	0.739	0.628
MTL-H	**0.809**	**0.757**	0.511	0.381	0.778	0.6471
MTL-S-SL	0.806	0.682	0.470	0.552	0.779	0.658
MTL-S-H	0.793	0.710	**0.630**	**0.828**	**0.789**	**0.750**

Model	Precision	Recall	F1
Google Cloud Speech [2]	0.304	0.205	0.245
Project DeepSpeech [3]	0.295	0.070	0.114
Our Approach	**0.307**	**0.504**	**0.382**

Table 1: (Left): Comparison of multitask learning formulations on the *test* set of our YouTube dataset. (Right): Comparison with two baselines for supportive speech prediction on the *test* set of our YouTube dataset.

every 100ms for the 95 CLASS-coded UVA videos. We then average the probabilities across the entire video for each category. Finally, we calculate the Pearson correlations r and associated p-values between the 4 category averages and the human-labeled scores for each of the 8 CLASS dimensions.

4.1. Results

Transfer from Simulated to Authentic (RQ #1): Pretraining on the crowdsourced speech recordings of people mimicking classroom environments is crucial: it delivers an accuracy boost, as assessed by the *average* of the 5 separate AUC scores (one for each class), from 0.448 to 0.750. In other words, without access to a larger pool of simulated-by-realistic data, the network is just guessing.

Multitask Learning (RQ #2): Table 1 (left) shows the AUC, separate for each speech category and also averaged over all categories. Results indicate that MTL increases accuracy substantially (from 0.618 to 0.750 average AUC). Moreover, hierarchical MTL is more effective than flat MTL. When we add the binary supportive predictor as another level in the hierarchy, performance improves again. Our findings differ from Krishna et al. [18], as we find that hierarchical MTL outperforms flat MTL even with a small dataset.

ASR Baseline Comparison (RQ #3): Table 1 (right) shows that both Google Cloud Speech and DeepSpeech have low recall rates, perhaps due to overlapping speech and ambient noise that is inherent to classrooms. In contrast, our system gives a higher recall (50.4%) for the same precision (~30%). This result is noteworthy because both Google Cloud Speech and DeepSpeech are likely trained on massive datasets. Hence, for application-specific tasks with specialized recording conditions, it can be beneficial to collect a highly tailored dataset, even if it is modest in size.

Prediction of CLASS scores (RQ #4): Fig. 3 shows an example of supportive speech classifications on one CLASS-coded video. The moments with the 5 highest probabilities are manually labeled with the actual phrase spoken, or an X if no supportive phrase was present (false positive). The *Good job* category is dominant, and the other categories are predicted with lower probability. Of the 5 labeled moments, 3 contained actual instances of *Good job* phrases ("very good", "right", and "yes").

Over all 131 videos in the UVA dataset, we find correlations with $p < 0.05$ for these CLASS dimensions: (1) *Facilitation of Learning and Development* with GJ (r=0.163, p=0.012), TY (r=-0.237, p=0.0002), and PLS (r=-0.170, p=0.009). (2) *Quality of Feedback* with GJ (r=0.152, p=0.019), TY (r=-0.194, p=0.003), and PLS (r=-0.135, p=0.037). (3) *Language Modeling* with PLS (r=-0.214, p=0.001).

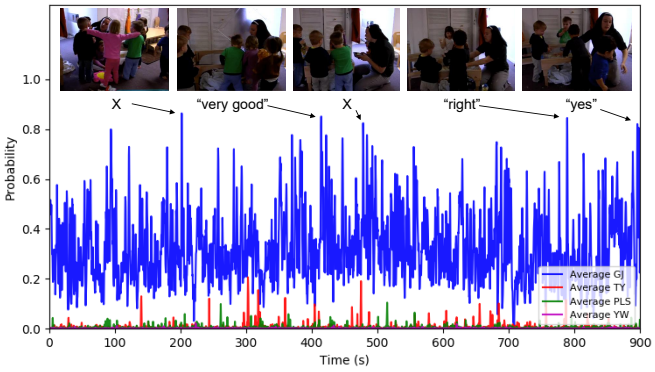

Fig. 3: The smoothed average supportive language probabilities over a 15min segment of a UVA CLASS-coded video.

Despite the simplicity of the aggregation method (simple averaging), we find non-trivial correlations between supportive language categories and multiple CLASS dimensions. This suggests that supportive speech detection would be a useful addition to an ensemble predictor of CLASS scores.

5. CONCLUSION

We develop a deep learning methodology for detecting supportive speech phrases in classroom videos, a challenging setting for ASR given the many simultaneous speakers and abundant noise. We find that (1) training on small but highly tailored datasets gives better accuracy for our application domain than a general-purpose ASR system (e.g., Google Cloud Speech) trained on huge datasets; (2) hierarchical MTL improves accuracy over a flat prediction architecture; (3) speech recordings by humans who are asked to mimic classroom speech is useful for training; and (4) the network's probabilistic predictions predict human-labeled CLASS scores on real classroom videos. **Future work** will incorporate the dialog context and emotion associated with a spoken phrase to potentially increase CLASS score prediction accuracy.

6. REFERENCES

[1] Robert C. Pianta, Karen M. La Paro, and Bridget K. Hamre, *Classroom Assessment Scoring System (CLASS) Manual: Pre-K*, Brookes Publishing, Baltimore, MD, 2008.

[2] "Cloud Speech-to-Text - Speech Recognition — Cloud Speech-to-Text — Google Cloud," 2019.

[3] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, and Andrew Y. Ng, "Deep Speech: Scaling up end-to-end speech recognition," *arXiv:1412.5567*, 2014.

[4] Karen M. La Paro, Amy C. Williamson, and Bridget Hatfield, "Assessing Quality in Toddler Classrooms Using the CLASS-Toddler and the ITERS-R," *Early Education and Development*, pp. 875–893, 2014.

[5] Rosemarie O'conner, Jessica De Feyter, Alyssa Carr, Jia Lisa Luo, and Helen 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.

[6] B K Hamre, Jason T Downer, Faiza M Jamil, and Robert C Pianta, "Enhancing teachers intentional use of effective interactions with children: Designing and testing professional development interventions," *Handbook of early childhood education*, pp. 507–532, 2012.

[7] Robin Cosbey, Allison Wusterbarth, and Brian Hutchinson, "Deep Learning for Classroom Activity Detection from Audio," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 3727–3731.

[8] Patrick J. Donnelly, Sean Kelly, Nathaniel Blanchard, Martin Nystrand, Andrew M. Olney, and Sidney K. D'Mello, "Words matter: Automatic detection of teacher questions in live classroom discourse using linguistics, acoustics, and context," in *International Learning Analytics & Knowledge Conference*, 2017, pp. 218–227.

[9] Zuowei Wang, Xingyu Pan, Kevin F. Miller, and Kai S. Cortina, "Automatic classification of activities in classroom discourse," *Computers & Education*, pp. 115–123, 2014.

[10] Wen Li, Fei Jiang, and Ruimin Shen, "Sleep Gesture Detection in Classroom Monitor System," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 7640–7644.

[11] Nigel Bosch and Sidney Dmello, "Automatic Detection of Mind Wandering from Video in the Lab and in the Classroom," *IEEE Transactions on Affective Computing*, pp. 1–16, 2019.

[12] Jiaojiao Lin, Fei Jiang, and Ruimin Shen, "Hand-Raising Gesture Detection in Real Classroom," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2018, pp. 6453–6457.

[13] Anand Ramakrishnan, Erin Ottmar, Jennifer LoCasale-Crouch, and Jacob Whitehill, "Toward Automated Classroom Observation: Predicting Positive and Negative Climate," in *IEEE International Conference on Automatic Face & Gesture Recognition*, 2019.

[14] Arkar Min Aung, Anand Ramakrishnan, and Jacob Whitehill, "Who are they looking at? Automatic Eye Gaze Following for Classroom Observation Video Analysis," in *International Conference on Educational Data Mining*, 2018, pp. 252–258.

[15] Tom Ko, Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "Audio augmentation for speech recognition," in *INTERSPEECH*, 2015, pp. 3586–3589.

[16] Yanmin Qian, Mengxiao Bi, Tian Tan, and Kai Yu, "Very Deep Convolutional Neural Networks for Noise Robust Speech Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 2263–2276, 2016.

[17] Abhinav Jain, Minali Upreti, and Preethi Jyothi, "Improved Accented Speech Recognition Using Accent Embeddings and Multi-task Learning," in *INTERSPEECH*, 2018, pp. 2454–2458.

[18] Kalpesh Krishna, Shubham Toshniwal, and Karen Livescu, "Hierarchical Multitask Learning for CTC-based Speech Recognition," *arXiv:1807.06234*, 2018.

[19] Nam Kyun Kim, Jiwon Lee, Hun Kyu Ha, Geon Woo Lee, Jung Hyuk Lee, and Hong Kook Kim, "Speech Emotion Recognition Based on Multi-Task Learning Using a Convolutional Neural Network," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2017, pp. 704–707.