WHOSECOUGH: IN-THE-WILD COUGHER VERIFICATION USING MULTITASK LEARNING

Matt Whitehill[⋆] *Jake Garrison*[†] *Shwetak Patel*[⋆]

[⋆]University of Washington
[†]Google Inc.

ABSTRACT

Current automatic cough counting systems can determine how many coughs are present in an audio recording. However, they cannot determine *who* produced the cough. This limits their usefulness as most systems are deployed in locations with multiple people (i.e., a smart home device in a four-person home). Previous models trained solely on speech performed reasonably well on forced coughs [1]. By incorporating coughs into the training data, the model performance should improve. However, since limited natural cough data exists, training on coughs can lead to model overfitting. In this work, we overcome this problem by using multitask learning, where the second task is speaker verification. Our model achieves 82.15% classification accuracy amongst four users on a natural, in-the-wild cough dataset, outperforming human evaluators on average by 9.82%.

***Index Terms*—** Cough, Health Sensing, Multitask Learning, Speaker Verification, Deep Neural Networks

1. INTRODUCTION

1.1. Cough Detection

Coughing is a symptom of many respiratory ailments such as asthma, tuberculosis, and cystic fibrosis. Thus, counting and analyzing coughs can serve as an important diagnostic tool for these conditions. Automated cough detection systems count the number of coughs in an audio file by distinguishing them from other sounds such as speech, background noise, and music. They usually begin by converting the audio waveform to a frequency representation, then use machine learning to identify the coughs [2, 3]. Recent work has achieved greater than 90% sensitivity using this approach [4, 5].

1.2. Cougher Verification

Cough counting algorithms face one important limitation—they cannot identify who produced the cough. This means whenever multiple people inhabit a common space, cough counting algorithms cannot attribute a cough to the right person. Thus, identifying who produced a cough sample could dramatically increase the utility of these systems.

Zhang et al. trained a system for speaker verification that also happened to perform well on cougher verification [1]. However, this system was tested on forced coughs, where a user is instructed to cough at the study coordinator's command. Because forced coughs are produced while the participant is consciously thinking about coughing, the resulting coughs usually sound very similar. Natural coughs pose a more difficult challenge. Because they are produced unintentionally, coughs from the same person can present in many unique styles and lengths.

1.3. Speaker Verification

Speaker verification involves determining whether a speech segment (utterance) comes from a specific speaker. First, an utterance is processed through an encoder to produce an embedding. Next, speakers are enrolled by aggregating known utterances from that speaker. Test utterances are then compared to the known utterances. If the test utterance is similar enough to the enrollment of a speaker, the speaker is verified.

Recently, deep neural networks have become the state-of-the-art for producing speaker embeddings. Wan et al. used a 3-layer LSTM and an end-to-end trainable cosine similarity metric to compare utterance embeddings [6]. Chung et al. also used cosine similarity, but with siamese training and a resnet-inspired convolutional neural network to create the embeddings [7]. Snyder et al. presented a time-delay neural network and statistical pooling layer to create embeddings called x-vectors, then a probabilistic linear discriminant analysis (PLDA) classifier to compare x-vectors [8].

1.4. Overview

In this paper, we introduce a cougher verification model to predict if a cough came from a given enrolled user. Our main contribution is a multitask learning training scheme that creates a general model when trained on both coughs and speech. We show this model can outperform a baseline model trained solely on speech. To the best of our knowledge, this is the first such model to be tested on natural, in-the-wild coughs.

2. MULTITASK LEARNING

Inspired by the work by Zhang et al. [1], we began our investigation by training a speaker verification model and testing it on our natural cough dataset (see Section 4.1 for details on this dataset). Figure 1 shows the t-SNE clustering [9] of cough embeddings for four different users. As shown in the figure, the speaker verification model groups together most coughs of the same user. However, it creates multiple clusters within each user instead of one cluster per user. This can lead to poor performance since the enrollment samples may all come from the same sub-cluster. When we listen to samples within these sub-clusters, we find the samples sound similar. For example, the samples in one sub-cluster may all be low amplitude or contain more pronounced vocal chord vibrations. This presents an interesting finding—the speaker verification model performs well as a sound detector. It is able to bring together audio samples that sound similar to humans, even if they are from a different domain (coughs).

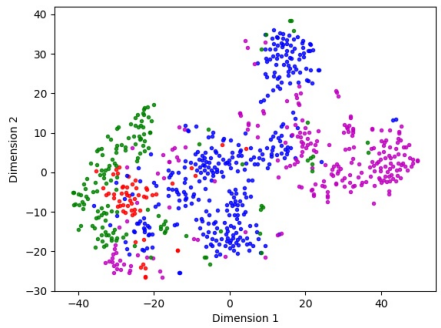


Fig. 1. t-SNE clustering of cough embeddings for four users using a speaker verification model.

The main obstacle to cougher verification is the lack of large-scale, publicly available, natural cough datasets. With only a small training set, creating a general model for unseen coughers is a significant challenge. Our goal in this work is to leverage the findings about the speaker verification model's performance on coughs to combat this issue. We utilize multitask learning [10] where the second task is speaker verification. This second task helps the model learn from the sound detector and create generalizable features for the cougher verification task. Figure 2 shows the training procedure.

3. MODEL ARCHITECTURE

In this section, we describe the model architecture to be used with the multitask learning training scheme. Samples are first converted to a mel-frequency spectrogram, framed, then processed by an encoder to create an embedding. Embeddings are then compared by cosine distance to predict whether the sample is from an enrolled user.

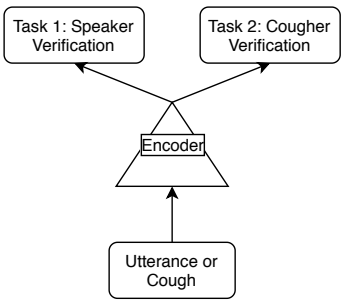


Fig. 2. Multitask training for model.

3.1. Framed Inputs

Cough episodes are variable length, ranging from as short as 150 ms to up to 3 s. Because the same user can produce coughs that are both long and short, we utilize a frame-based approach to prevent the model from using sample length as a feature. We convert the audio sample to its mel-frequency representation, segment the spectrogram into shorter frames, then process each frame through an encoder to create an embedding. We then average the embeddings from each frame to get the global embedding.

We use a Hamming window of size 25 ms with 10 ms step size and 40 mel-filterbanks. Each sample is framed by stacking non-overlapping windows of 19 frames in time by the 40 mel-filterbanks. We select 19 frames since it is the smallest symmetric window that provided sufficient temporal dimensionality at the final encoder convolutional output.

3.2. ResNet Encoder

Similar to the work by Chung et al. [7], we use a residual-network (ResNet) architecture for the encoder [11]. We use 3 residual network blocks where each block has the same structure. First is a convolutional layer with a 2x2 kernel and a stride of 2, followed by 2 layers with a 3x3 kernel and a stride of 1. The block's layers use 64, 128, 256 filters respectively. Batch normalization (batch-norm) and a rectified linear unit (ReLu) follow each convolutional layer. A skip connection links the output of the first convolutional layer's batch-norm to the output of the final layer's batch-norm. After the residual blocks, a channel-wise average pooling layer is applied, followed by a fully-connected layer to create an embedding for each frame. These embeddings are then averaged to create a global embedding.

3.3. End-to-End Model Loss

In each task, we use the generalized end-to-end loss proposed by Wan et al. [6]. Let e_{ij} be the output embedding where i is the speaker or cougher ID and j is the utterance or cough ID. Let c_k be the centroid of all embeddings for the speaker or cougher k; however, where $i = k$, e_{ij} is removed from

the centroid calculation of c_k. The similarity matrix S_{ij} is the cosine similarity from the embedding e_{ij} to each centroid c_k:

$$S_{ij,k} = w \cdot \cos(e_{ij}, c_k) + b$$

where cos is the cosine similatiry function and w and b are learnable parameters. We then use the softmax to calculate the loss:

$$L_n = \sum_{ij}(-S_{ij,i} + \log \sum_{k=1}^{N} \exp(S_{ij,k}))$$

where N is the number of speakers or coughers and n is the task number. The full loss is

$$L = L_1 + \alpha L_2$$

where L_1 is the speaker verification task and L_2 is the cougher verification task. Because there are less coughers than speakers in our dataset, we use .05 for α to encourage progress on the speaker task before focusing on the cougher task.

4. EXPERIMENTAL SETUP

4.1. Cough Dataset

In this work, we use the in-the-wild, natural cough dataset from our lab's previous work on cough counting [3]. To collect the data, participants with a frequent cough carried a smartphone in their shirt pocket or on a lanyard around their neck for 3 to 6.5 hours. The dataset contains 2,445 individual coughs within 1,331 cough episodes from 16 participants (8 male, 8 female). See Table 1 for coughs per user.

User	Coughs	User	Coughs	User	Coughs
1	32	7	79	13	142
2	41	8	81	14	230
3	53	9	85	15	261
4	64	10	98	16	904
5	67	11	102		
6	77	12	129		

Table 1. Coughs per user sorted by count.

Each individual cough is manually segmented to the beginning and end of the cough. During training, each individual cough is taken as a separate sample. At inference time, we convert each individual cough to its mel-frequency spectrogram, then stack all spectrogram frames for individual coughs in the same episode to create one sample; we call this a *combined cough*. We define an individual cough as being included in the same episode if it occurs within 500 ms of the end of the previous individual cough. Figure 3 details this further.

Because the cough dataset was collected in-the-wild, the samples have significant background noise. To address this issue, we use data augmentation. We first produce 4 extra copies of each cough with varying amplitude. We then use the MUSAN dataset [12] to apply background noise, music, and babble at 5db to produce 15 total copies.

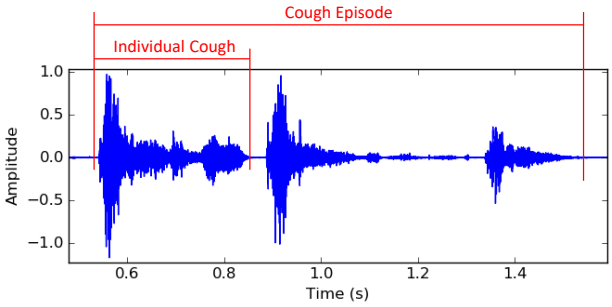


Fig. 3. Time domain plot of a cough episode.

4.2. Speaker Dataset

To further reduce the impact of background noise, we train the speaker verification task on the Voxceleb dataset [13]. Voxceleb is a large-scale speaker verification dataset compiled from YouTube videos. Because many of the videos contain background noise, training the model on this dataset helps produce noise-robust features.

4.3. Training

Of the 16 users in the cough dataset, we use 12 for training and leave out 4 for test. Because the dataset is already small, we always use the 3 users with more than 200 coughs for training, not test. We also only train on user 4's samples as they contain an abnormally high level of noise. Of the remaining 12 users to be used for test, we use 3-fold cross-validation, holding out 4 users (2 male, 2 female) at a time.

For training, we use batches of size N users $\times$ M utterances, where $N = 12$ and $M = 10$, for both coughers and speakers. We use the same hyperparameters as included in the work by Wan et al. [6], with the exception of also adding an $L2$ regularization loss of .001 to prevent overfitting. We train for 10,000 training steps and decay the learning rate after 2,000 and 3,500 steps.

5. RESULTS

5.1. Human Evaluation

To quantify the challenge of in-the-wild, natural cougher verification, we begin by obtaining a human baseline. First, a human evaluator is permitted 5 minutes to listen to 5 random samples from each of the 12 test users so they can get familiar with listening to coughs. We then select 4 users from one

of the cross-validation sets and perform verification and classification tests. For the verification test, we iterate through each of the 4 users, selecting 10 random combined coughs as enrollment. The evaluator is able to listen to these 10 samples as often as they like while evaluating the test samples. We then randomly present 10 test samples, 5 from the same user, 5 from different random users, and ask the evaluator to determine whether the cough is from the same user. For the classification test, we present 20 random samples of coughs, 5 from each of the 4 users, and ask the evaluator to determine which of the 4 users each sample came from. We use 8 human evaluators and each performs both tests for the 3 folds. The results are listed in Table 2.

Metric	FAR (%)	FRR (%)	Verif Acc (%)	Class Acc (%)
Average	15.73	21.77	81.25	74.77
Median	12.07	16.38	83.89	77.69
Std Dev	7.34	3.95	7.59	13.72
Best	8.62	10.34	88.89	90.00
Worst	27.59	43.10	68.61	43.70

Table 2. Statistics for the human evaluation including false accept rate (FAR), false reject rate (FRR), verification accuracy, and classification accuracy.

The results show evaluators performed better on the verification task. Evaluators commented that it was easier to make a binary decision (same speaker or different speaker) than a four-way classification. It may have also been easier for evaluators to use enrollment samples during the verification task when there were only 10 samples versus 40 for classification.

As demonstrated by the verification (81.25%) and classification (74.77%) accuracies, in-the-wild, natural cougher verification is a difficult task. Natural coughs are influenced by both environmental conditions and physiological changes that can produce dissimilar-sounding coughs. For example, a user may intentionally reduce the cough amplitude to not disturb a quiet setting. Or if they have a particularly challenging contaminant in their respiratory system, they may cough more harshly than usual. In-the-wild data collection also increases the difficulty as channel effects and background noise are more pronounced than in a controlled setting.

This is most easily viewed in Figure 4b which shows the t-SNE clustering of four users' cough embeddings using our model. While most coughs by the same user are clustered together, there are outliers. Upon listening to these samples, we notice the outliers sound very different from the rest of the cougher's samples. We would not expect a human or model to be able to classify these correctly.

5.2. Model Results

Our baseline model has the same architecture and hyperparameters as our model, but is trained only for speaker verification on the Voxceleb dataset. To evaluate both the baseline and our model, we use 10 random samples per user as enrollment. See Table 3 for the results.

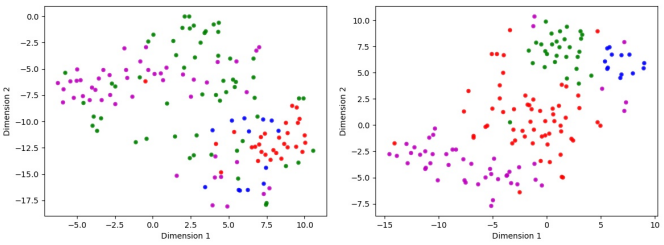
Model	FAR (%)	FRR (%)	EER (%)	Class Acc (%)
Human Evaluation	15.73	**21.77**	N/A	74.77
Baseline Model	16.49	38.13	30.04	73.05
Our Model	16.25	23.41	**22.69**	**82.15**

Table 3. Results for the human evaluation, baseline model, and our model. EER refers to the equal error rate. For the baseline and our model, we find the model's similarity threshold that approximately matches the human evaluation FAR, then report the FRR at that threshold.

The results show our model provides a 24.47% decrease in EER and a 12.46% increase in classification accuracy over the baseline. It also outperforms the human evaluators in the classification task on average by 9.87%, although it yields a lower specificity in the verification task.

5.3. Embedding Visualization

Figure 4 shows the t-SNE clustering of the embeddings from one cross-validation fold. As shown in the figure, our model produces improved clusters of embeddings over the baseline model.



(a) Baseline Model (b) Our Model

Fig. 4. t-SNE clustering of cough embeddings for one cross-validation test set of 4 users.

6. CONCLUSION

In this work, we present a novel multitask learning approach for in-the-wild, natural cougher verification. Training with a secondary task of speaker verification helps overcome the small dataset problem to create a more general model. We show that our model can, on average, outperform human evaluators at a 4-way classification task using 10 enrollment samples. Using 3-fold cross-validation, we achieve a 22.69% EER and 82.15% classification accuracy.

7. REFERENCES

[1] Miao Zhang, Yixiang Chen, Lantian Li, and Dong Wang, "Speaker recognition with cough, laugh and" wei"," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2017, pp. 497–501.

[2] Pablo Casaseca-de-la Higuera, Paul Lesso, Brian McKinstry, Hilary Pinnock, Roberto Rabinovich, Lucy McCloughan, and Jesús Monge-Álvarez, "Effect of downsampling and compressive sensing on audio-based continuous cough monitoring," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2015, pp. 6231–6235.

[3] Eric C Larson, TienJui Lee, Sean Liu, Margaret Rosenfeld, and Shwetak N Patel, "Accurate and privacy preserving cough sensing using a low-cost microphone," in *Proceedings of the 13th international conference on Ubiquitous computing*. ACM, 2011, pp. 375–384.

[4] Justice Amoh and Kofi Odame, "Deep neural networks for identifying cough sounds," *IEEE transactions on biomedical circuits and systems*, vol. 10, no. 5, pp. 1003–1011, 2016.

[5] Filipe Barata, Kevin Kipfer, Maurice Weber, Peter Tinschert, Elgar Fleisch, and Tobias Kowatsch, "Towards device-agnostic mobile cough detection with convolutional neural networks," in *7th IEEE International Conference on Healthcare Informatics (ICHI 2019)*, 2019.

[6] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno, "Generalized end-to-end loss for speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4879–4883.

[7] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.

[8] David Snyder, Daniel Garcia-Romero, Gregory Sell, Alan McCree, Daniel Povey, and Sanjeev Khudanpur, "Speaker recognition for multi-speaker conversations using x-vectors," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 5796–5800.

[9] Laurens van der Maaten and Geoffrey Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[10] Rich Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[12] David Snyder, Guoguo Chen, and Daniel Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[13] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.