EmoNet: Audio-only Emotion Detection using Federated Learning

Adar Arnon
Harvard University
Boston, MA
aarnon@g.harvard.edu

John Keck
Harvard University
Boston, MA
rok805@g.harvard.edu

ABSTRACT

Audio-only emotion detection is growing in popularity as more businesses look to leverage audio streams from customer service calls and internal meetings for insights into behavior. While there are many robust NLP models for sentiment analysis on text, models trained on audio features are less common. As audio data is larger and more complex than text data, it has historically been more computationally intensive and time consuming to create labelled audio feature datasets and run inference tasks on these data. With the increased development of deep neural networks and audio machine learning frameworks, these audio-only inference tasks have become much easier to perform while the data collection has remained difficult. To this end we present EmoNet, a convolutional neural network trained through federated learning, allowing for a distributed solution to the lack-of-data problem in existing models. EmoNet consists of a client-server infrastructure with a centralized server orchestrating federated model weight updates and multiple clients manually labeling new data, performing local training, and relaying updated model weights back to the central server. By hosting the client at a publicly accessible URL, EmoNet is able to bootstrap an improved audio-only emotion detection model without creating or storing a centralized dataset.

1 Introduction

Sentiment detection has become more common in recent years as businesses, governments, and individuals aim to pull insights out of an ever-growing stream of text, audio, and video data. In telemedicine, customer support, sales, and other conversational scenarios, companies have begun to use automated sentiment detection to improve patient outcomes, triage angry customers, and close more deals. While the rise of easy-to-use machine learning frameworks has increased the accessibility of these sentiment models, their implementations have not changed much over time. Many companies use large corpora of text data to train NLP models to infer sentiment from things like tweets or chat messages.This text-based system is helpful for many online activities but is lacking when it comes to first-class audio data. Our system, EmoNet, allows individuals to easily run audio-first sentiment detection on streaming audio data. Audio-only sentiment detection gives the user the ability to skip transcription steps normally required by NLP sentiment models and preserves end-user privacy. Additionally, this method requires much less computational power than a traditional pipeline of speech-to-text and NLP classification. By using embeddings of the audio data itself, EmoNet can also be applied regardless of language. EmoNet is trained using a federated system as well, allowing the model to improve through a democratized portal over time.

In this paper, we will describe the key issues and opportunities with current audio-only sentiment detection systems, give background on the space, describe the EmoNet system, and give preliminary results from testing the system.

2 Key Problems & Opportunities

EmoNet addresses two fundamental factors: the lack of high fidelity, audio-only emotion detection models and the lack of a large dataset of labelled audio sentiment data. The lack of data leads to the lack of a highly performant and usable model which leads to fewer people focused on creating improved datasets. This negative feedback loop has prevented audio-only models from emerging into the mainstream.

The lack of data comes from the difficulty in creating labelled datasets for audio-only learning. The standard data creation method is to use voice actors reading

scripts and then having a group of bystanders rate the emotion on a scale. For example, the RAVDESS dataset used in training many audio-only neural networks has multiple actors speaking two sentences with a range of emotions. While quality data, this dataset has fewer than 5000 recordings. The effort required to increase the number of data points in the dataset is high enough that it prevents others from attempting to create more. SAVEE, another common audio-only detection dataset, has even fewer data points.

Another issue with these datasets is the lack of diversity in data. SAVEE, for example, only has 4 different actors and they are all men from the U.K. All the major datasets are also English speakers and many of them also skew towards men. Many of the datasets come from the same age group, ranging from 20s - 50s. This lack of variation in the training data leads to biased models that do not accurately predict emotion for those outside of that group.

Building a federated system allows us to address these problems by allowing people of all characteristics to help train the model from their web browser. This distributed data collection system allows the user to record a sentence with a certain emotion, training a local model, and using the individual data points to bootstrap a high-fidelity centralized model.

3 Audio-only Emotion Detection

Audio-only emotion detection is poised to augment or replace text-based sentiment models due to the accuracy improvement when using multi-modal models as well as the increased need for privacy-protection for end-users. Emotion detection as a whole has increased in popularity in recent years and research into the space has increased as edge device inference has become more possible with current processors.The transition from text-based NLP models to audio processing for sentiment analysis has opened up many opportunities for improvement in the space which EmoNet aims to solve.

3.1 Background

As NLP models became more robust and text corpora grew in size, the default for much of this sentiment detection work was done by running text data like emails, instant messages, or social media posts through basic tokenization and subsequently classification models that inferred sentiment of the message. Often, this text data is simply rated on a positive, neutral, negative scale and is used for high-level classification of positivity. In a customer service context for example, an angry customer chatting with a support representative would classify as negative and the rep could upgrade the level of the inquiry to a manager or higher tier. These basic models have been improved upon by including more granular labels, delineating anger and sadness for example.

As the NLP models became the standard for sentiment detection tasks, transcription services and speech-to-text APIs allowed for audio streams of spoken work to be converted to a sentiment as well. The pipeline of audio input to transcription service to NLP sentiment model has since become the de facto method of analyzing sentiment from audio.

3.2 Motivation

Although these models are useful, there are many downsides to a transcription step. User privacy is often a concern in business contexts and the transcription step creates a log of personal information in the process. Additionally, transcription services are notoriously expensive, both computationally and financially. This makes running emotion detection on audio streams in real-time much less attractive.

By transitioning to an audio-only model for emotion detection, we are able to bypass the text-base modality used by many of the most common methods which improves the accuracy and decreases cost.

3.3 Related Work

There has been initial work done in the audio-only emotion detection space, broadly falling into the following categories:

- Sentiment Analysis on Speaker Specific Speech Data
  - Models are often based on speech transcripts, similar to the traditional speech-to-text to NLP model pipeline for sentiment detection.
- Sentiment Extraction from Natural Audio Streams
  - Models often use Youtube videos, speeches, or songs as input data. These natural audio streams are good for diverse data inputs but are costly and time-consuming to label.
- Audio-only Sentiment Analysis
  - Models are based purely on acoustic features. Similarly to the Mel-frequency cepstrum coefficient method used in EmoNet, others have used other audio features for inference.
- Multimodal Audio-Text Sentiment
  - Models combine audio and text sentiment data to improve accuracy. While these are useful, it does not meet our criteria of leveraging only anonymous audio data.
- Datasets
  - Some work has been done in attempting to collect applicable datasets for audio-only training. The most common of these datasets are RAVDESS and SAVEE

4 Success Criteria

With a federated system like EmoNet, the success criteria we analyzed were the inference time and the improvement in accuracy with the additional data. Our goal was to retain an inference time at the client of under 1 second and an improvement increase of 10%. By keeping inference speed under 1 second, we would allow the end-user to perform inference in near real-time instead of post-processing. Additionally, increasing the accuracy of the model is important as the biggest problem with current audio-only models is overfitting on existing, small datasets. Being able to improve accuracy while using additional user-generated data will show a positive improvement to the existing technology.

4.1 Project Goals & Metrics

We set out to work on this project with two goals in mind.

- Create an audio-only emotion detection model based on publicly available datasets. The model should be performant enough to run on edge devices.
- Create a federated learning infrastructure to improve this model over time based on “crowdsourced” contributions. Run the model with multiple rounds of contributions and verify the accuracy improves.

5 Approach

Separate from text-based or multi-modal approach, our audio-only approach uses Mel-frequency cepstrum coefficients (MFCCs) to train a convolutional neural network. Many existing emotion analysis models have been LSTM or MLP and using a CNN allows us to improve upon these models using the MFCCs.

In EmoNet, Mel-frequency cepstrum coefficients are used as input to a multi-layer convolutional network to analyze pure audio characteristics. MFCCs are a commonly used tool in audio processing and audio compression and can be used to analyze the waveform in a discreet manner. These coefficients may be used as a 1-dimensional input to a multi-layer CNN with fully connected layers to classify emotion within a range.

With an initially trained CNN using MFCCs from the RAVDESS dataset, our approach then uses a real-time, on-device inference and labeling interface to bootstrap the model in a distributed manner. With a small footprint and fast inference speed, the CNN may be used on an end user’s device where labeling can be quality controlled by the user. By hosting the interface on a public-facing website (https://emonet.xyz), we are able to allow many different users the ability to add their own voice and emotion label, update the model locally on their own device, and improve the central model. These validated inferences may be used to continue to train central mode across different countries and groups of people.

With newly self-labeled data by multiple users, the existing CNN is improved with real-world data. Raw audio data is not stored or transmitted, only MFCCs are used for inference, and all training is done at the edge device. Weights are sent from the client device anonymously to the central server where updates can be made. At no time is any personally identifiable information transferred along with the model.

6 Key Differentiators & Intellectual Points

As discussed, many of the systems currently used for emotion detection use text-based or multi-modal methods for classifying sentiment. These models are large and slow and require complicated pipelines, making on-device inference difficult. Leveraging Mel-frequency cepstrum coefficients and the characteristics of the audio itself, we are able to create a CNN that accurately and quickly infers emotion from a short audio snippet.

WIth audio and sentiment data, privacy is a major concern. Many current implementations require sending data to a central server for transcription and inference. With a federated learning approach, we are able to allow the user to label data and improve the model at scale without disclosing the content of the audio recording or sending any data other than the updated weights back to the central server.

7 Work Performed

EmoNet consists of two main components: the EmoNet Client and the EmoNet Server. The EmoNet Client contains the frontend functionality of the public, user-facing real-time inference and data labelling tool. The EmoNet Server contains the business logic and the federation process. These two in concert make up the EmoNet audio-only emotion detection system.

Before creating the EmoNet Client/Server, the initial CNN was created using keras and tensorflow. As mentioned in the approach, the model was built with 2 1-dimensional convolutions with dropout, max pooling, and a ReLU activation. The network was then flattened and densed and used a softmax to classify the emotion in the audio sample.

Additionally, in order to train the model, two datasets were cleaned and prepped for EmoNet. Both RAVDESS and SAVEE were used and utility functions were created to ingest the data for use in the EmoNet model.

All of the initial work collecting and cleaning the data, building and testing the EmoNet model, and creating the scaffolding for the federation process between the device and central server was done in a Google Colab notebook for each of collaboration and testing. Once

comfortable with the initial designs, the Client and Server were written for production.

The EmoNet Client was designed using a Flask application that renders a simple HTML/CSS/JS single page website. This site uses a third-party audio recording JavaScript file to allow the user the ability to record audio on the browser. Once recorded, the user has the ability to replay or re-record at any time. EmoNet prompts the user to record a sentence with a certain emotion (neutral, calmness, happiness, sadness, anger, fear, disgust, or surprise). Once the user records the audio, they then have the ability to use the current model to predict the emotion or submit the model to be used in the federated training.

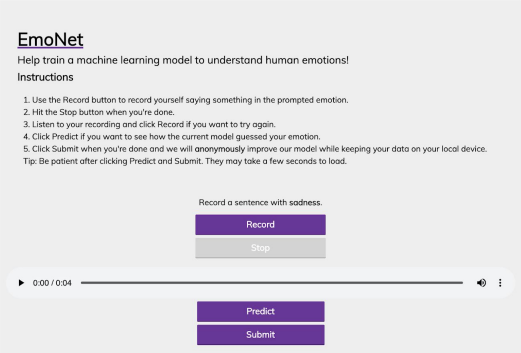

Fig. 1 - EmoNet user interface

After the user records the audio, clicking predict calls the /predict endpoint on the backend API which pulls the most up-to-date model from the centralized server, runs an inference, and then returns the response via an alert on the page.

Clicking submit uses the local version of the model to train and then sends the updated model weights to the centralized server. The user can continue to submit and predict on the EmoNet site as much or as little as possible.

The EmoNet Server was also built as a Flask application with no frontend functionality. This API, hosted at https://api.emonet.xyz, acts as the orchestrator for the federated system.

The API has four endpoints: /receive-update, /send-model, /test-model, and /predict.

The /receive-update endpoint receives the model updates from each client, stores them temporarily in memory, and then updates the base model every 10 updates from the client. This is the core logic for the federation of the model.

The /send-model endpoint sends the most up-to-date base model to the requesting client. This allows the client to train on the most recent model when training at the edge.

The /test-model endpoint uses a dataset to test the base model as it stands at the moment. Users can use RAVDESS or SAVEE to test the current base model.

Finally, the /predict endpoint uses the base model and an input of MFCCs to run an inference.

Both EmoNet Client and EmoNet Server were built using Flask to leverage the Python machine learning community. Once both applications were completed, they were deployed to Google Cloud Platform using Google App Engine Flexible. This, along with a custom domain, allowed us to test the end-to-end system in the real-world.

Once deployed, we used the /predict endpoint on the EmoNet Server to capture the accuracy improvement over time. As more clients continued to use the EmoNet Client inference and labeling interface, the accuracy of the model when compared to RAVDESS or SAVEE improved.

Finally, we ran additional ad hoc tests using the predict functionality on the EmoNet Client. While analyzing test accuracy of the model against RAVDESS or SAVEE datasets is a helpful proxy for success, testing in real-time is very important as we did not want to be restrained by existing data. By running multiple tests with different people, we could see how the accuracy improved as more people added their data to the trained model.

8 Results

After initially training the EmoNet model locally, we saw an accuracy of 95% when tested against the RAVDESS dataset used to train the model. Test accuracy against the SAVEE dataset was around 30%. This showed that the model was very sensitive to the data input. Therefore, we were confident that our federated method with a simple or empty base model would be more effective than building on top of an existing dataset. Even though we implemented dropout in the CNN, the model seemed to overfit to the RAVDESS dataset when trained.

Once trained for 1000 data labels submitted via the EmoNet Client, the test accuracy against the RAVDESS dataset increased from 95% to 98%, signifying that the federated system was an improvement in the existing model itself and accuracy increased with more anonymous data. When tested for the SAVEE dataset, the accuracy increased even further to 48%. This showed that the improvement against SAVEE was even higher due to the increased diversity in the data.

In addition to accuracy results against existing datasets, we performed ad hoc testing that resulted in very interesting results. While this was very manual, we ran 100 inference tests on the EmoNet Client using the predict button. Out of the 100 tests run by two different people, we saw a 72% accuracy with the prompted

emotion being correctly inferred. While this is a rough estimate of overall model accuracy, we could eventually add a tracker to count and log the number of correctly predicted emotions from EmoNet Client.

On the other hand, the inference speed significantly slowed when running on a browser. While our initial goal was under 1 second inference time, the inference speed hovers between 1 and 1.5 seconds in the current version of EmoNet Client. There are most likely optimization opportunities in the system to decrease that time to below the 1 second desired.

Overall, EmoNet showed significant promise as a distributed method for improving the accuracy of audio-only emotion detection models using federated learning.

9 Conclusion

In this paper we presented a novel way to collect audio data for training emotion detection models. By using this approach, we significantly improved the accuracy of models built on publicly available datasets.

Labeled audio data collection has been challenging historically due to privacy concerns, but federated learning has alleviated this fear. It makes the approach both more scalable and more adoptable - computation is done on edge devices instead of a central cloud, and users will be more willing to use a model that protects their privacy. This has been proven with text models used in Google's keyboard (Gboard), and we believe that audio adoption is the next logical step.

The infrastructure used for EmoNet is open-source, and can be used for any kind of learning that requires simple end-user interaction. A centralized dataset of real-world audio samples is beyond our reach, but we imagine a user-friendly, privacy-preserving tool for quick model building and iteration.

10 Future Work

For our research, we implemented a web service that allows users to submit labeled audio samples. This approach is limited by our ability to reach willing participants. There are multiple ways to expand quickly:

- Use tools like Mechanical Turk to encourage participation and collect samples
- Build an SDK that can be integrated into existing applications. For example, a customer service center might want its agents to label recorded calls according to the user's perceived emotion to improve the quality of their service.

Any tool that uses our API can improve the model, and collectively these approaches will reach a more diverse and balanced user base. Reaching a diverse user base is critical if this model is to be used in real-world applications. The speakers in datasets like RAVDESS and SAVEE are professional actors who are super-playing emotional intensity. The data is missing a variety of accents, origins, and languages. It is comparable to creating a text-messaging prediction model based on classic English literature.

Another aspect that could be further researched is the model itself. Our work revolved around improving a specific model we built; another approach could A/B test different kinds of models to see which model trains better and achieves a better test accuracy. This will be most efficient at scale with many users, so we defer this research to a later point in time.

REFERENCES

[1] Livingstone SR, Russo FA (2018) The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. PLoS ONE 13(5): e0196391. https://doi.org/10.1371/journal.pone.0196391.

[2] S. Haq and P.J.B. Jackson, "Multimodal Emotion Recognition", In W. Wang (ed), Machine Audition: Principles, Algorithms and Systems, IGI Global Press, ISBN 978-1615209194, chapter 17, pp. 398-423, 2010.

[3] S. Haq and P.J.B. Jackson. "Speaker-Dependent Audio-Visual Emotion Recognition", In Proc. Int'l Conf. on Auditory-Visual Speech Processing, pages 53-58, 2009.

[4] S. Haq, P.J.B. Jackson, and J.D. Edge. Audio-Visual Feature Selection and Reduction for Emotion Classification. In Proc. Int'l Conf. on Auditory-Visual Speech Processing, pages 185-190, 2008

[5] C. Busso, M. Bulut, C.C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J.N. Chang, S. Lee, and S.S. Narayanan, "IEMOCAP: Interactive emotional dyadic motion capture database," Journal of Language Resources and Evaluation, vol. 42, no. 4, pp. 335-359, December 2008.