*p*WAVE: A Novel Dataset for Emotional Confidence Detection

Sanjay Kannan
Stanford University
skalon@stanford.edu

Rooz Mahdavian
Stanford University
rooz@stanford.edu

Abstract

We present a novel hand-labeled dataset for learning emotional confidence detection on American English speech. Our dataset consists of 9 quarterly earnings calls, each labeled by one to two raters. In this paper, we describe some useful preprocessing on the collected data, develop some metrics of inter-rater reliability, explain a number of modeling features from the raw call audio, and introduce baseline models for the task at hand.

1 Problem Statement

Our modeling task is as follows. Given a speech utterance in American English, output a measurement of the utterance's projected confidence.

Here, we use the term *confidence* to denote the perceived emotion of confidence, rather than any notion of statistical confidence. This is further operationalized in Data Collection.

Specifically, we focused on *acoustic* indicators of perceived confidence, which means we will not be using any word-level features. This helps to remove semantic indicators of confidence (like the "strength" of a particular word) from their underlying acoustic features. In other words, we hoped to test the adage that it's not *what* you say, it's *how* you say it.

Our output is a discrete confidence score within some range, where the maximum value in the range represents the highest possible perceived confidence, and the minimum value in the range represents zero perceived confidence. This is discussed further in Methodology.

2 Motivation

A quantified measure of perceived confidence would be useful in a number of situations. For instance, tools could be built to provide analytical confidence metrics. People might rehearse a speech and receive real-time feedback on how confident they sound, adapting their speaking to improve those scores.

And as with all affect detection systems, these tools could help those with disabilities like autism perceive confidence, which is crucial in high-stakes situations (political debates and business meetings, among other things).

3 Literature

Unfortunately, we were unable to find any previous research focusing specifically on confidence detection in speech. Statistics and machine learning papers have overloaded the meaning of *confidence*, and this complicated our bibliographic review via search engines.

However, we were able to find existing work on traits like assertiveness, which is closely related to confidence in the colloquial sense. One notable paper (Ranganath et al., 2013) describes a speed-dating corpus consisting of over 1000 four-minute speed dates; the speakers in this study were variously rated for their perceived friendliness, flirtatiousness, awkwardness, and assertiveness.

Other relevant studies are concerned with stress detection in speech. Intuitively speaking, stress is a factor in determining confidence, as high stress in an utterance might be an indicator of low confidence.

Work done at Cornell (Lu et al., 2012) achieved very promising results on this task, with over 80% indoor detection accuracy, utilizing an adaptive GMM-based model and input from an off-the-shelf Android device microphone. Among other things, this study suggests that highly controlled data collection is not necessary for interesting results.

Of course, we have reviewed the papers involving the most general form of our task: emotional affect detection. Confidence is an emotional affect, and there is plenty of existing work that demonstrates the tractability of spoken affect detection. In particular, we point to the work of (Scherer, 2003), (Schuller et al., 2011), and (Liscombe et al., 2003).

4 Data Collection

To train a confidence detection model, we required human-labeled confidence data on speech utterances. Gridspace Incorporated provided us with nearly 200 recorded quarterly earnings calls from various companies. The calls were delivered as raw `WAV` files, and we were also given their word-level transcripts as `TXT` files.

For two of the calls, we were provided with labeled confidence data for every millisecond of audio, each reading consisting of a discrete integer from 0 to 100. For a given reading, higher numbers indicated higher confidence relative to other readings.

Unfortunately, confidentiality issues prevented us from receiving the fully-labeled dataset. As a result, we collaborated with another research group to hand-label further data.

In addition to the two earnings calls we were provided labels for, 10 earnings calls were selected randomly from the full set. Given a sign-up spreadsheet for these calls, participants were free to choose their calls. However, participants were encouraged to rate calls without existing raters before tackling those calls with existing raters.

It was communicated to participants that we do not care about confidence in a semantic sense. If a speaker communicated uncertainty about their company's future, that would be immaterial to our task. On the other hand, if they communicated this uncertainty in a very tentative manner, that is something we wanted a lower confidence rating for.

For rating purposes, participants used a proprietary software tool from Gridspace. Named SecurePlay, the tool lets users play audio files while dragging a slider component, allowing them to rate the current perceived confidence at every timestep in a recording. (See Figure 1 for a visual.) The span of the slider is a range from zero to one hundred. After playback finishes, the recorded confidence data is written to disk on a per-millisecond basis.

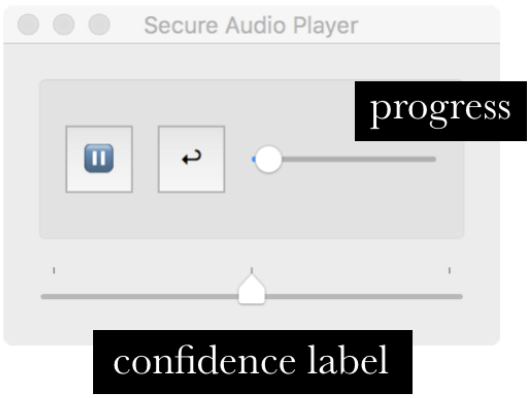

Figure 1: The SecurePlay rating tool.

When rating was complete, a total of 7 calls were rated by a total of 5 participants. Three of the seven calls received a rating by at least two participants, for an average of two calls rated per participant.

5 Methodology

5.1 Preprocessing

Our collected data immediately lends itself to a number of supervised modeling tools. Model inputs become the raw audio files for each call, while model outputs are zero-to-hundred confidence ratings for each millisecond of a call.

While this is valid in theory, it is intractable to directly learn from and predict so many timesteps of confidence, and especially from audio files up to two hours long.

Moreover, participants received little to no direction on their use of the confidence slider. Indeed, some participants used the full width of the slider to capture small shifts in confidence, while other participants kept the majority of their ratings near the median value. In a similar vein, participants were also likely to have a reaction time between hearing an utterance and assessing its confidence.

Finally, we noted that confidence ratings were noisy, displaying significant amounts of jitter. More meaningful than every tiny slider change were longer-term trends in rating.

To address these issues, we applied a number of preprocessing steps to our raw confidence labels. For the remainder of this section, we will be describing this procedure on a single earnings call.

Consider the vector of confidence ratings given by a single participant, rating at every millisecond on a given call. We first shift the vector of ratings t milliseconds in time ($t = 300$ was our empirical choice), using the assumption that participants had a typical reaction time of about t milliseconds. For the lost ratings at the end of this vector, we pad the vector with mean values to compensate, although the loss of t ratings is typically inconsequential.

Next, to mitigate issues of rating scale and bias, we standardize this vector by subtracting its mean and dividing its entries by their standard deviation.

Third, we compute an exponentially-weighted moving average on each rater's vector. To avoid skew, we do this in both the forward and backward directions, averaging the two moving averages as a smoothed form of the original vector. We use the Pandas library (McKinney, 2010) to compute the moving average, and we apply an empirically-validated `span` parameter of 30000, which we found to be a good compromise between discarding noise and losing meaningful data.

This smoothing appears to work well, but may suffer from a theoretical issue. Because we are smoothing across the entire rating vector for a given participant and call, our method is extending confidence trends across speaker boundaries.

However, given that the speakers in earnings calls talk for extended periods of time, we find this issue important to note, but by no means devastating to the legitimacy of our work. In the future, a speaker diarization step would allow us to do more granular, intra-speaker smoothing.

Examples of the smoothing procedure are given in Figure 2 for two calls with multiple raters.

Last, we condense the smoothed rating vectors to intervals of 0.1 seconds by averaging every 100 ratings. Using the word-level transcripts of the calls (which are timestamped to a precision of 0.01 seconds), we then split the condensed ratings into sentence-level chunks. Intuitively, words are the smallest coherent unit we assign confidence to, and within a sentence it is informative to assess each word's contribution to the overall confidence of the sentence.

Therefore, a single training *example* to our model can be described as follows. Its target output is a sequence of confidence ratings, at every 0.1 second interval in a given sentence. The input that should produce that target output is given by a sequence of audio feature vectors, which corre-

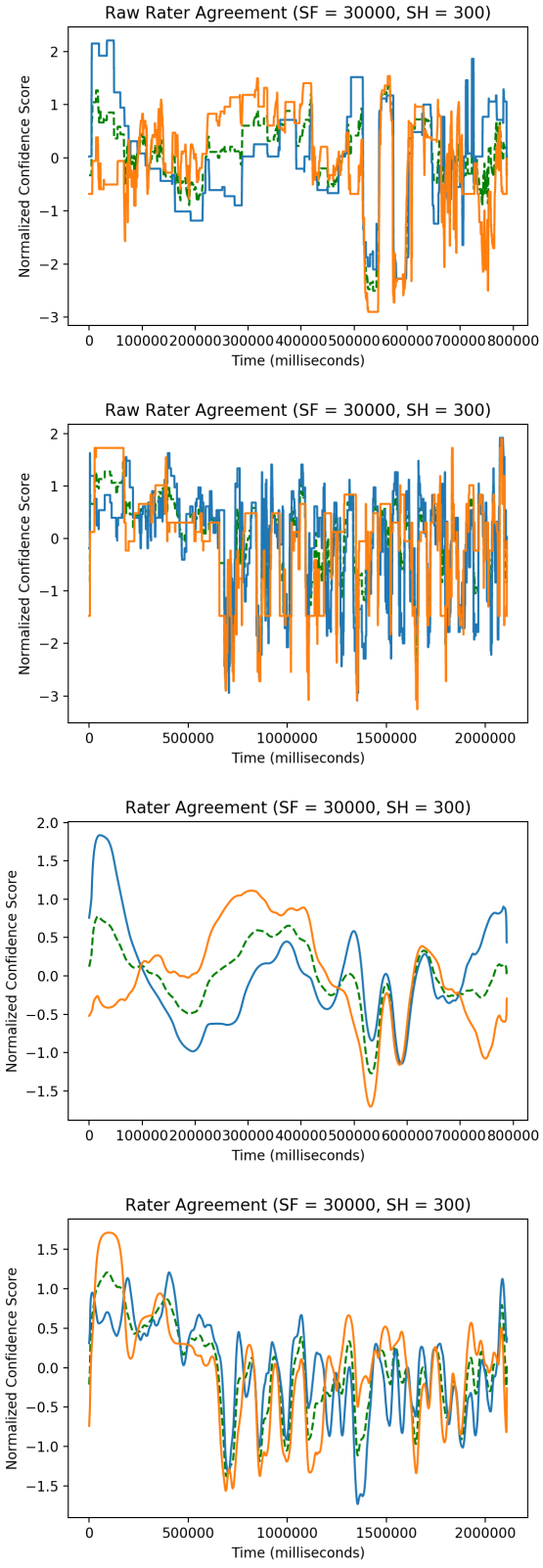

Figure 2: The top two graphs show confidence ratings for two calls, each call having two raters. In each graph, one rater's scores are in blue, the other rater's scores are in orange, and their averaged score is given in green. The bottom two graphs display the smoothed versions of these curves.

spond to each of these intervals in a sentence. In Feature Extraction, we describe the computation of these feature vectors.

5.1.1 Inter-Rater Reliability

For calls with multiple raters, the final preprocessed scores for each rater were averaged to produce target scores for the call.

In addition to visually inspecting the rater similarity on a call, we briefly define a metric of inter-rater reliability. Given the preprocessed rating vectors for two raters r and s, where each vector has n elements and $r[i]$ denotes the ith index of vector r, we have

$$\sqrt{\frac{1}{n}\sum_{i=0}^{n}(s[i]-r[i])^2}. \quad (1)$$

For more than two raters $r_1, r_2, \ldots, r_m$, we average the above value for all unique pairs of raters, and analogously write

$$\frac{2}{m^2-m}\sum_{r_j,r_k}\sqrt{\frac{1}{n}\sum_{i=0}^{n}(r_j[i]-r_k[i])^2}. \quad (2)$$

We term this metric the root-mean-squared rater disagreement (RMSRD). Higher RMSRD values are clearly worse from a consistency standpoint. With the limited dataset we have (only three calls with two raters each), comparing these values was not very useful. However, when undertaking a large data collection effort, this metric might inform the handling of outliers or the codification of rating guidelines.

5.2 Feature Extraction

As we noted before, features must be extracted at 0.1 second intervals, and then collated into examples for each sentence using the provided transcripts.

We made canonical use of Mel-frequency cepstral coefficients (MFCCs) as a natural speech feature (Hasan et al., 2004), extracting each of 13 cepstral coefficients per 0.25 second interval. (We averaged 4 of these sub-intervals to get the MFCCs for a full interval). We tried extracting delta and double-delta MFCCs, but their benefit was obviated by using recurrent model architectures.

We also extracted the first formant as the acoustic correlate of speaker pitch. At the same time, we avoided word-level features for the reasons discussed in Problem Statement.

Below, we describe our experiments in further detail, but it is instructive to provide some ablative results on a particular instance of the confidence task (3-class INTRA-FILE).

MFCC vectors alone achieved 74.14% validation accuracy on this task, while a one-dimensional pitch vector alone achieved 58.62% validation accuracy. Together, the features achieved 63.79% validation accuracy.

Theoretically, using the feature vectors together should do better than using either of the feature vectors individually, but optimizing along this higher-dimensional manifold is more difficult and may require different hyperparameters for training.

5.3 Modeling

Our dataset suggests a sequence model, and in particular one we might classify as *many-to-many*. For each sentence, we feed in several intervals of audio features, and at each interval in the sentence, we expect our confidence detection model to output a confidence score.

For our purposes, we made two changes to this paradigm. First, instead of predicting a confidence score at each interval, we predict a single confidence score representing the average confidence of the intervals in a sentence. By contrast, this would be a *many-to-one* model.

Second, remember that we standardize our confidence scores. For a given confidence score, we can easily interpret it as the number of standard deviations from mean confidence. However, the distribution of labels is not necessarily a bell curve around zero, so a confidence score of 0.5 may be significantly confident if there is a long-tailed distribution. Furthermore, there are empirical benefits to using discrete output labels (as compared to a regression problem on continuous labels).

Therefore, we bucket confidence scores into bins, based on evenly-spaced percentiles of the sorted training data. Consider a hypothetical case where we have 100 sentences in our training set, and the fiftieth sentence by sorted confidence scores has a label of 0.7. We would put all training, test, and validation examples with lower confidence scores into bin 1, then place the remaining examples into bin 2. An analogous procedure holds for more bins.

Architecturally, it is helpful that the intervals in a sentence have temporal structure. We exploit

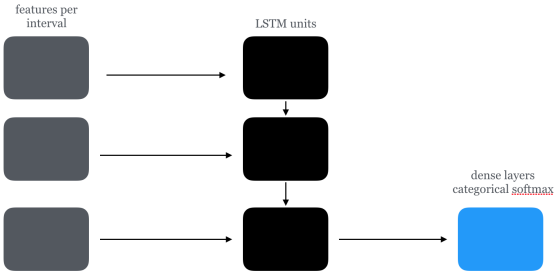

Figure 3: A recurrent architecture for predicting sentence confidence. The left-most set of blocks represents feature vectors from each interval in a sentence.

this structure to build Recurrent Neural Networks (RNNs). Our networks take in feature vectors at each timestep of a sentence, propagate a hidden state across these timesteps, and then output bin scores corresponding to each potential confidence bin. Our model's confidence prediction is simply the arg-max over these bins.

Our model architecture is depicted in Figure 3. Note that we use one or more dense layers on top of the recurrent layer output, which improves the representational capacity of our network. To improve longer-term dependency tracking across sentence intervals, we also use long short-term memory (LSTM) units (Hochreiter and Schmidhuber, 1997) instead of conventional units in our recurrent layer. All models were implemented in Keras with a Tensorflow backend.

Concerning an adequate hyperparameter search, we tried different numbers of bins, different learning rates, and different network layer sizes. A selection of these trials are detailed in the next section.

5.4 Experiments

We ran three experiments on the model, which we will denote as INTRA-FILE, INTER-FILE (Simple), and INTER-FILE (General).

In the INTRA-FILE experiment, we trained the model on a call-by-call basis, where sentences from an individual call were split 70-20-10 into train, validation, and test sets. The model achieved noteworthy levels of performance, achieving 89.66% validation accuracy with 2 confidence classes and 74.14% accuracy with 3 confidence classes.

In the INTER-FILE (Simple) experiment, we trained the model across all calls, where sentences

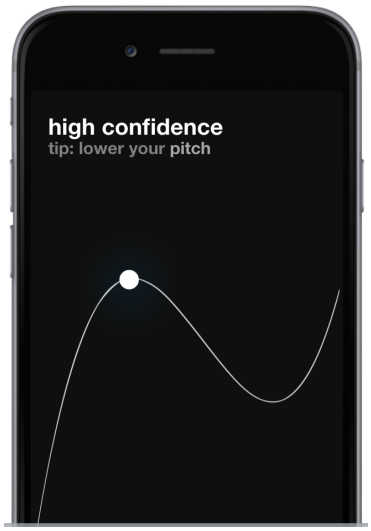

Figure 4: A real-time application for confidence feedback.

from all calls were split randomly 70-20-10 into the train, validation, and test sets. The model performed worse, with validation accuracy dropping to 65.73% with 2 confidence classes and 50.00% with 3 confidence classes. This is disappointing but somewhat expected, since the model must learn to generalize beyond individual calls. This task is objectively more challenging.

In the INTER-FILE (General) experiment, we split entire *calls* 70-20-10 into train, validation, and test sets, then assigned all sentences from these calls to the corresponding sets. The model performed even worse in this setting, with validation accuracy dropping to 45.48% with 2 confidence classes and 39.38% with 3 confidence classes. These numbers are especially paltry; a randomly-guessing oracle would achieve respective accuracies of 50% and 33%.

However, this drop was also expected. In this setting, the model must completely generalize not only to unseen sentences, but sentences from unseen calls. Precisely speaking, the joint distributions of inputs and outputs might be completely different between seen calls and unseen calls, and the model would have no way of knowing this from the examples it trains on.

The INTER-FILE (General) setting is probably the closest thing to a real-world confidence environment. We really want to improve these scores and believe that significantly more data is the solution to doing so.

6 Future Work

We have presented a novel dataset for learning emotional confidence detection, in addition to relevant preprocessing methods, target features, metrics of inter-rater reliability, and baseline models.

Future work would focus on two tasks. First, we hope to scale up the data collection via the described routines. We hope to improve the performance of our baseline model, where these early results with limited data have been promising.

Second, we hope to develop a smart-phone application using this model. This hypothetical application would report a real-time confidence score for incoming microphone data, and suggest adjustments in voice features that might improve that confidence score. This concept is illustrated in Figure 4.

7 Acknowledgements

We would like to thank Andrew and the rest of the CS 224S teaching staff for their effort and dedication this quarter! We would also like to thank Gridspace Incorporated for providing us with the raw audio of earnings calls.

References

Md Rashidul Hasan, Mustafa Jamil, Md Golam Rabbani Md Saifur Rahman, et al. 2004. Speaker identification using mel frequency cepstral coefficients. *variations* 1(4).

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Jackson Liscombe, Jennifer J Venditti, and Julia Hirschberg. 2003. Classifying subject ratings of emotional speech using acoustic features. In *INTERSPEECH*.

Hong Lu, Denise Frauendorfer, Mashfiqui Rabbi, Marianne Schmid Mast, Gokul T Chittaranjan, Andrew T Campbell, Daniel Gatica-Perez, and Tanzeem Choudhury. 2012. Stresssense: Detecting stress in unconstrained acoustic environments using smartphones. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, pages 351–360.

Wes McKinney. 2010. Data structures for statistical computing in python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*. pages 51 – 56.

Rajesh Ranganath, Dan Jurafsky, and Daniel A McFarland. 2013. Detecting friendly, flirtatious, awkward, and assertive speech in speed-dates. *Computer Speech & Language* 27(1):89–115.

Klaus R Scherer. 2003. Vocal communication of emotion: A review of research paradigms. *Speech communication* 40(1):227–256.

Björn Schuller, Anton Batliner, Stefan Steidl, and Dino Seppi. 2011. Recognising realistic emotions and affect in speech: State of the art and lessons learnt from the first challenge. *Speech Communication* 53(9):1062–1087.