INTERSPEECH 2021: Show & Tell Contribution
30 August – 3 September, 2021, Brno, Czechia

Audio Segmentation based Conversational Silence Detection for Contact Center Calls

Krishnachaitanya Gogineni, Tarun Reddy Yadama, Jithendra Vepa

Observe.AI, India

krishna@observe.ai, tarunr@observe.ai, jithendra@observe.ai

Abstract

In a typical contact-center call, more than 35% of the call has neither the contact-center agent nor the customer speaking, we usually refer to such areas in the call as *Conversational Silences. Conversational silences* comprise mostly of hold-music, automatic-recorded-messages, or just silences when the agent or customer is engaged in some off-call work. Most of these conversational silences negatively affect important KPIs for call-centers, like dead-airs affect customer satisfaction, long-holds affect average call handling time and so on. In this paper we showcase how Observe.AI helps contact-centers identify agents who are breaching accepted levels of conversational silences by using an in-house Audio Segmenter system paired with an NLP system to classify the contexts around these *Conversational Silences*. This solution is provided by Observe.AI to hundreds of contact centers who use it to improve their average call handling time and customer satisfaction scores

Index Terms: audio segmentation, spoken language understanding, contact center AI, speech analytics, business intelligence

1. Introduction

Conversational Silence - Area in a contact center call where *neither the agent nor the customer is speaking*. The two most important types of Conversational Silences in a contact center are dead-air and Hold-time-violation.

dead-air - A conversational silence which occurs *without the agent giving any prompt* to the customer to expect a silence.

Hold-time violation (**HTV**) - A conversational silence when the agent *puts the customer on hold* while following the apt protocol *but the duration of the hold exceeds a specified pre-set limit.*

Identification of hold-time violations and dead-air instances in contact center calls are very important as these directly drive important business KPIs of contact center like Average Handle Time and Customer Satisfaction Score

2. Architecture

To detect dead-air and hold-time violation incidents automatically, we process every call in 3 stages.

First, the audio recording uploaded by the customers is sent to an **Audio Segmentation module** for the extraction of the audio profile. This module divides the audio into multiple segments for e.g speech-over-music, silience, noise, etc. and this audio profile is further sent through a post processing logic where multiple segments are merged together based on their label, to produce an audio file divided only into two regions,i.e conversational silence and conversational speech regions. For example, hold-music and and background speech are both put into conversational silence and only the portions of interaction between the customer and agent is tagged as conversational speech.

Second, the call is sent into an **ASR engine** to get the transcription. Transcript combined with the conversational silence and speech regions are used to extract the 40 word context before every conversational silence instance.

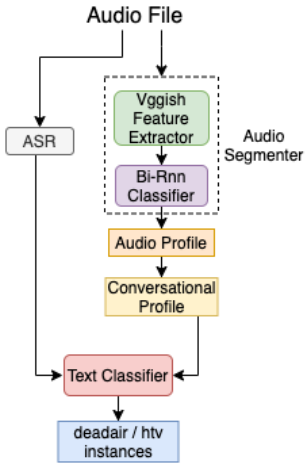

Figure 1: *Architectural Diagram for dead-air/htv detection*

Third, this context of 40 words is fed into a **text classifier**, which classifies every conversational silence into one of two categories: a no prompt category and a hold-time prompt category.

Further based on the *minimum duration of a silence threshold*, *length of maximum hold-time threshold* set by the user combined with the prompt category from the *text classifier*, each conversational silence instance is categorized as a dead-air or a hold-time violation respectively. The timestamps of occurrence of each instance are saved and are shown on the dashboard. Introducing audio segmenter resulted in a 25% absolute improvement in dead-air precision and a 75% increase in the volume of true htv detections when compared to a vanilla ASR offset based silence detection pipeline.

We describe the Audio Segmenter module and text classifier in detail in the following sections. Figure 1 Shows the complete system flow and architecture.

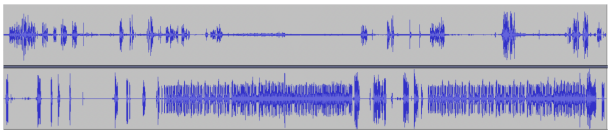

Figure 2: *Dual channel audio file, Customer channel at the top with Agent channel at the bottom*

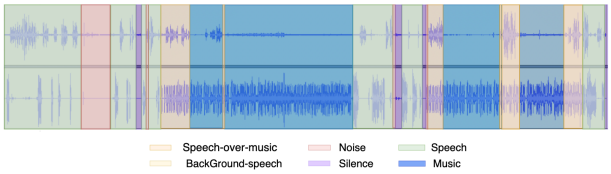

Figure 3: *Audio Profile powered by Audio Segmenter*

2.1. Audio Segmenter

Audio segmentation refers to the class of theories and algorithms designed to segregate audio streams into contiguous scenes. These scenes can be perceived as equivalents of paragraphs in text, and can be associated with various categories. Thus enabling us to peek in the dimension of audio profiling.

We manually analysed hundreds of customer calls from various verticals to identify distinct categories which encompass the audio profile. We zeroed in on Speech, Silence, Music, Speech-over-music, Background speech, Noise and Dial-Tone for our contact center use case.

Historically, audio profiling has been addressed with audio-signal-features such as MFCCs and statistical classifiers based on Gaussian mixture models and Hidden-Markov models, more recent approaches use deep learning inspired models like Convolutional neural networks and Recurrent neural networks. Dividing the problem into two parts, Feature extraction and segment classification. We adopted the Vggish[1] architecture based on CNNs applied on an input of audio-spectrogram for feature extraction, followed by a Bi-directional Recurrent Neural Network[2] based architecture for segment classification. Figure 2 is a visualization of a dual channel audio call with both channels stacked vertically and Figure 3 represents the audio profile of the call.

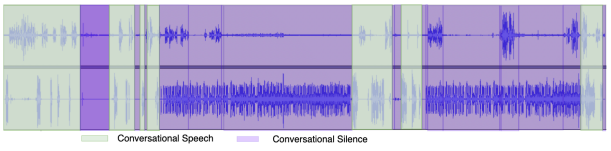

Figure 4: *Conversational Profile*

2.2. Text classifier

Context before a silence is used to predict the type of prompt given for the silence. Through experimentation we have narrowed down to a context length of 40 words spoken before any conversational silence instance. The text classifier takes the input as the 40 word context and classifies it into one of the prompts; Hold-time prompt or a no prompt in which case it can be inferred as a Silence without a prompt. Figure 4 shows the conversational silence and speech profile and Figure 5 shows the prompts detected for each silence instance.

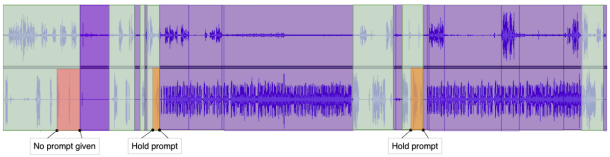

Figure 5: *Prompts identified through text classifier*

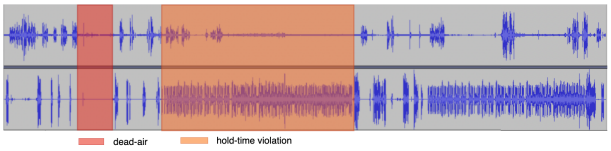

Figure 6: *dead-air and hold-time Violation instances*

Figure 6 shows the detection of a dead-air instance and a hold-time violation instance in the audio call.

3. Dashboard

Dashboard combines all the details about a call to present at a single place optimising the time spent per call by the user. They can listen to the audio, see the transcription of the audio and look at any instances of dead-air and hold-time violations on the same screen.

User can click on any part of the audio to go to the corresponding transcription. All instances of dead-air and hold-time violation are tagged on the visual representation of the audio. dead-air is highlighted in red color for the corresponding duration and hold-time violation is tagged at the start of the hold. One can hover on any such instance to look at the duration and also upon selecting any instances, will be directed to the text spoken before the silence.

Dead-air and hold-time can be configured by the user with parameters, such as *minimum length of silence instance* to be considered for as a valid conversational silence and *maximum length of hold-time* before it is considered as a violation.

4. Conclusions

Audio segmenter helps contact centers to identify silences in the calls effectively and with the audio and text available at a click, quality analysts are able to quickly find and associate a silence to a reason. This empowers them to train agents in avoiding unwanted silences and attend to customer queries in a swift way thereby decreasing customer wait time and consequently, improve the overall customer experience and reduce average customer handling time.

5. References

[1] S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson, "Cnn architectures for large-scale audio classification," 2017.

[2] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," pp. 2392–2396, 2017.