



Real-Time Monitoring of Silences in Contact Center Conversations

Digvijay Ingle, Ayush Kumar, Krishnachaitanya Gogineni, Jithendra Vepa

Observe.AI, India

digvijay.ingle@observe.ai, ayush@observe.ai, krishna@observe.ai, jithendra@observe.ai

Abstract

Contact center conversations often contain segments with hold music, automatic-recorded-messages or pure silences, where neither the customer nor the agent is speaking. We refer to these segments as Conversational Silences [1]. These silences when continued beyond an acceptable level can negatively impact important contact center KPIs, like average handling time, agent efficiency, etc. and may lead to poor customer experience. As a result, it becomes imperative for contact centers to identify silences in conversations and define mechanisms to better handle them. In this paper, we propose a cascaded system consisting of an ASR engine, a silence detector block, a text classification layer and a heuristic engine to surface instances in calls where agents are missing the protocols to handle silences. This system is used to trigger alerts to agents in real time thus enabling them to course correct while being on call with the customer. Moreover, these instances can also be surfaced to their supervisors so as to identify agents who are frequently missing these protocols and thereby design dedicated coaching sessions.

Index Terms: contact center, silence, hold, audio segmentation, speech analytics, real time, hold time

1. Introduction

Contact centers form a central point through which organizations handle customer interactions via various channels like chat, email and phone calls. Of these, phone calls happen to be the most preferred choice by customers to reach out to support teams as it offers a chance to have a live interaction with an agent and thus get their issues resolved immediately. Silences form an integral part of these interactions that appear when either of the parties need to indulge in some off-call work. While these silences are important to provide the best resolution to the customer, the impact they can have on the business is two-fold: 1) Lower operational efficiency due to longer call handling times, 2) Risk of poor customer experience.

Table 1 illustrates scenarios where silences naturally appear in conversations, however, the manner in which the agent handles them determines how they would be perceived by the customer. Before placing the customer on hold, agents often give an estimate on the duration of silence to customers using appropriate prompts. However, while the silence is in progress, they are generally engrossed in finding resolution for the customer and often lose track of the hold-time they promised (Example 2). On the other hand, in the course of the conversation agents often miss to use an appropriate prompt before going on silence (Example 3). As a result, this is perceived as an abrupt silence by customer as opposed to a formal hold. Either of these scenarios are misses by the agent and can potentially lead to poor customer experience. Contact centers often have evaluation mechanisms, wherein agents receive periodic feedback from their supervisors who identify similar instances where a potential breach had occurred and accordingly design coaching sessions for them. However, these evaluations are generally

done in a post call setup where supervisors review a bulk of calls by an agent and identify the most common misses to provide the feedback. This makes the process retrospective rather than preventive. Moreover, the gap between consecutive coaching sessions with the supervisor often ranges from a few days to weeks implying that particular mistakes committed by the agent while handling calls could potentially be repeated all throughout this time. Thus it becomes imperative for contact centers to have a mechanism that helps surface these issues in real time and appropriately prompt agents during the call so that they can take corrective measures to prevent such instances. In this paper, we propose a real time system to:

1. Identify if the agent used appropriate prompt to place the customer on hold and trigger an alert on missing it
2. Identify if the running hold time is within the promised levels and trigger an alert if it continues beyond that
3. Surface the instances flagged by the system for further action by supervisors

Table 1: Illustrative Examples. The color codes used are as follows: a) **Blue** -Actual silent segment in the call, b) **Pink** -Indicative prompt used by agent, c) **Yellow** -Silence Interval

S.No	Example
1	<p>Agent: do you mind if i place you on a two to three minutes hold</p> <p>Customer: sure please go ahead</p> <p>SILENCE [1:37 - 3:15] <hold time adhered></p> <p>Agent: thank you for holding</p>
2	<p>Agent: do you mind if i place you on a two to three minutes hold</p> <p>Customer: sure please go ahead</p> <p>SILENCE [1:37 - 5:15] <hold time not adhered></p> <p>Agent: thank you for holding</p>
3	<p>Agent: your email id please</p> <p>Customer: abc at xyz dot com</p> <p>SILENCE [2:16 - 2:57] <no prompt used></p> <p>Customer: hello are you there</p>

2. Pipeline & Components

The pipeline consists of four components as discussed below:

2.1. Automatic Speech Recognition Engine

For transcribing the audio conversations to text we use a third party ASR Engine. The audio conversations consist of English dyadic conversations that span across multiple lines of business like retail, e-commerce, financial services, healthcare, etc. We evaluated performance of the ASR Engine on calls across these industries and observed a WER of 15-20%.

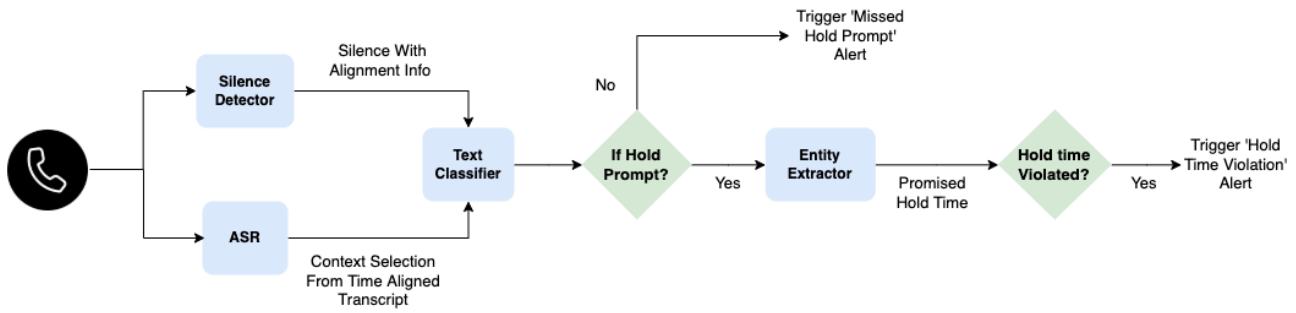


Figure 1: Architecture Diagram

2.2. Silence Detector

Silence detector module consists of an audio segmenter model proposed in our past work [1]. The audio segmenter model uses a VGGish [2] architecture based on CNNs to extract features from an audio-spectrogram, followed by RNN block to classify segments in an audio into one among the following classes: speech, silence, music, speech-over-music, background speech, noise and dial-tone. For the current use-case, we specifically consider segments that are tagged as *silence* by the audio-segmenter model. We apply an additional filter to discard silences that are shorter than 5 seconds since short silences are very common in contact center conversations and have minimal impact on customer experience.

2.3. Text Classifier

Text classifier block uses the output of silence detector module to extract a fixed context of W words before the beginning of the silent segment. Based on our experiments we found that $W=40$ provides best results for our use case and hence we fix this value. This context is fed to a text classification model that predicts whether a preemptive indication or prompt (similar to Examples 1 and 2 in Table 1) was given by the agent to the customer before placing them on hold. The model uses DistilRoBERTa-base¹ architecture and is built by fine-tuning a language model pre-trained on our proprietary conversational dataset [3]. The macro averaged F1 for this model is observed to be 86% on the above classification task.

2.4. Entity Extractor

In order to keep the customers well-informed, agents often give an indication to them to expect silence via prompts like, ‘*may i place on a two to three minutes hold*’, ‘*give me one minute to take a look*’, etc. The entity extractor block uses a regex module to extract the hold time promised by the agent in these prompts and consequently compare it with running time of silence following the prompt.

3. Real Time Monitoring

For every call that the agent is handling, the *ASR Engine* generates real time transcription whereas the *Silence Detector* block continuously monitors for silence. As soon as a silence is identified, the pipeline triggers alerts based on the outputs of *Text Classifier* and *Entity Extractor* blocks. If the *Text Classifier* block detects a prompt, it further calls the *Entity Extractor* block to extract the promised hold time and consequently triggers a

‘*Hold Time Violation*’ alert to the agent when the running time crosses the promised hold time. The agent can then provide latest update to the customer or proactively inform them about any additional time required to find the resolution. In case the *Text Classifier* is unable to detect the prompt, it triggers a ‘*Missing Hold Prompt*’ alert to the agent so that they can course correct and inform the customer.

Furthermore, the instances flagged by the system are streamed to an interactive dashboard post completion of the call where users can access the call recording, transcript and the instances that are flagged by above system. Agent supervisors can use this dashboard to get a holistic view of calls and identify agents who are frequently missing the protocol. Further, they can also create dedicated coaching sessions for them and track progress on the same. Thus our system helps improve customer experience by assisting agents to better handle silences and provides visibility to the supervisors into opportunities of improvement for their agents.

4. Conclusions

In this paper, we put forth an end-to-end real time system that not only helps surface instances in calls that lead to poor customer experience but also trigger alerts to agents that can help them take necessary actions to course correct. This further serves as an instantaneous feedback mechanism to agents and thus helps reduce the time to action from a few days or weeks to a few minutes as compared to traditional one-on-one feedback from supervisors. Furthermore, the instances flagged by the system can also be viewed on the dashboard which can be used by supervisors to generate periodic reports to identify agents who are frequently missing the protocol and thereby design dedicated coaching sessions for them.

5. References

- [1] K. Gogineni, T. R. Yadama, and J. Vepa, “Audio segmentation based conversational silence detection for contact center calls,” in *Interspeech 2021, 22nd Annual Conference of the International Speech Communication Association, Brno, Czechia, 30 August - 3 September 2021*, pp. 2349–2350.
- [2] 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. W. Wilson, “CNN architectures for large-scale audio classification,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017, New Orleans, LA, USA, March 5-9, 2017*, pp. 131–135.
- [3] A. Kumar, M. N. Sundararaman, and J. Vepa, “What bert based language model learns in spoken transcripts: An empirical study,” in *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP, 2021*, pp. 322–336.

¹<https://huggingface.co/distilroberta-base>