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

Addressing Compliance in Call Centers with Entity Extraction

Sai Guruju, Jithendra Vepa

Observe.AI, India

sai@observe.ai, jithendra@observe.ai

Abstract

Call centers record and store customer-agent conversations for the purpose of coaching, quality assurance and to comply with *Industry Regulations*. Good amount of these audio recordings contain sensitive information pertaining to their customers' financial or personal details. To ensure data security, compliance and to reduce the risk of abuse/theft, it becomes important to identify such instances in audio recordings and mask these segments. To automate this process, we propose a cascaded system; first, Automatic Speech Recognition (ASR) generates transcript and text-to-audio alignment information for an audio recording. Then, Entity Extraction is performed on generated transcripts to identify and locate sensitive information, and the corresponding sensitive segments are masked in audio recordings using alignment information. We introduce a novel system for selective masking of sensitive information in both audio and transcript.

Index Terms: entity extraction, audio masking, call centers, customer-agent interactions, compliance

1. Introduction

With ubiquitous proliferation of call centers across the geographies, massive volume of customer calls are flowing in and out of call center telephony systems. These recorded audio interactions represent the voice of customer, as they hold the customer's feedback, suggestions, queries and problems faced with the products and services offered.

A significant number of these interactions contains sensitive/critical information ranging from Payment Card Industry (PCI) data (example: *Credit/Debit/Prepaid Card details*), financial information (example: *Bank Account Numbers, Routing Numbers, Card details*) to Personally Identifiable Information (PII) (example: *Contact numbers, Mailing Addresses, Email Addresses, Social Security Numbers, Date of Birth, Names*), raising the risk of misuse, even when accessed internally. To reduce this risk, without limiting the data availability for training, masking such sensitive information becomes critical. It is important to note that the definition of sensitive varies from company to company. For a debt collection agency, it is important to mask financial information but personal information should be retained to ensure correct verification process. For a retailer, only the PCI data should be masked. Given the large amounts of data, it becomes an arduous task to manually annotate and mask sensitive information, introducing an opportunity for automated tools to perform audio masking. We focus on the problem of masking sensitive information from a customer-agent audio recording. In an audio masking task, given an audio recording, segments containing sensitive information are identified, and audio recording is modified by masking only such segments.

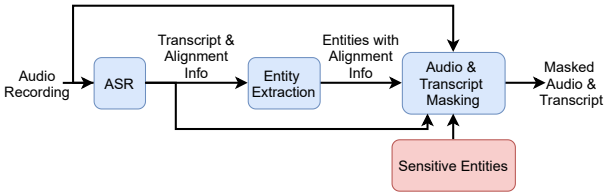

Figure 1: *Architectural diagram for Selective Masking pipeline*

2. Pipeline & Components

Proposed pipeline (Figure 1) consists of three key components:

1. *Automatic Speech Recognition:* an ASR is used to convert the audio recording into a transcript
2. *Entity Extraction:* transcript is processed to locate and identify entities
3. *Audio & Transcript Masking:* both audio and transcript are modified by masking the sensitive entities identified

2.1. Automatic Speech Recognition

For transcribing the audio conversations, we use a third-party ASR system. The audio conversations usually contain two speakers and a majority of them are dual channel, with a speaker on each channel. All the conversations are in English and majority of them have North American accent. We have evaluated ASR's performance on several (10+) customers, which cover a wide range of industries (*collections, retail, automotive, banking* etc) and observed a WER score of 15% - 20% for them.

For an audio stream $A(t)$, transcription is a sequence $\{(w_j, T_j, c_j, s_j)\}$ units, where w_j is the word uttered by the speaker s_j in the time interval $T_j = [t_j^{start}, t_j^{end})$ with ASR assigned confidence c_j. We use the alignments generated for masking the audio/transcript in the final step. We do not use speaker information and confidence scores generated in the downstream tasks.

Table 1: *Entities considered with description*

Entity	Description
address	Postal Addresses, ZIP Codes
bank-account-number	Bank Account Number
card-details	Credit/Debit Card Number/CVV
date	Date of Birth, Delivery date
email	Email Address
internal-number	Order/Reference Numbers
money	Monetary Mentions
phone-number	Contact Number
routing-number	Routing Number
time-mentions	Time Mentions
unique-identification-number	Social Security/License Numbers

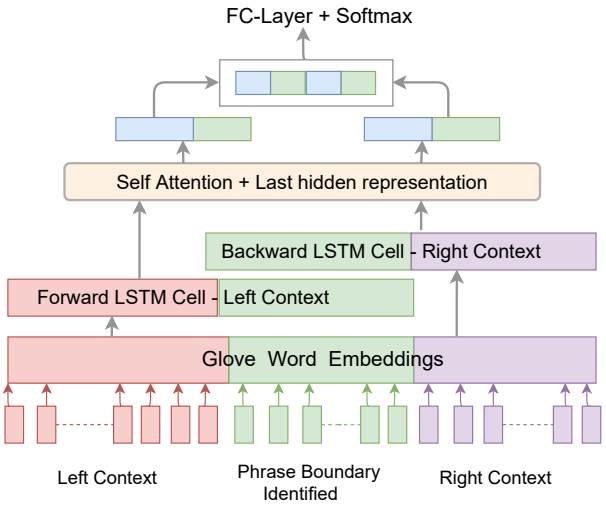

Figure 2: *Architectural diagram for Classification network*

2.2. Entity Extraction

Given a ASR transcript, we identify the entities present along with their boundaries in the transcript. Currently, eleven entities (Table 1) are supported. This approach consists of two modules: (i) heuristic engine for figuring out phrase boundaries (ii) a deep learning classifier over identified boundaries.

2.2.1. Heuristic Engine:

Since most entities mentioned are numeric and keyword oriented, we determine possible entity boundaries using heuristics. This includes rules identifying contiguous blocks of keywords (*cardinal numbers, ordinal numbers, month names, monetary terms etc*) and a regex module to match email mentions in ASR transcript. The phrase boundaries that do not belong to any entity type but are sampled by heuristics engine, are mapped to *rest*.

We compare boundaries identified in manual transcripts with that of ASR transcripts by the heuristic engine, found a significant match (around 95%) between them. This could be attributed to the robustness of heuristics. Also, ASR system has a WER score in the range of 10.3% - 12% in recognising the keywords used.

2.2.2. Deep Learning Classifier:

We use a *deep bidirectional LSTM-Attention network* (Figure 2) with GloVe [1] word embeddings for initialization, consuming both the identified boundary and supporting context to find the entity type. Few reasons for choosing this architecture:

Context: An individual phrase does not provide sufficient information about what it represents. For example: *"two thousand thirteen"* could refer to a Date or last four digits of a Social Security Number. We take into consideration both left and right context to disambiguate between competing entities.

Long time difference: During the exchange of information, customer/agent may take time to obtain the information asked, and the customer is engaged through small talks in this mean time. This increases the gap between useful context (agent's ask for information) and identified boundary (customer's response). *Long Short Term Memory (LSTM)* [2] networks are used to learn these long-ranging dependencies.

Attending to correct context: Consider the snippet, *"and the zip code associated with this bank account is five three seven nine nine four"*. Due to proximity of the phrase *"bank account"* to the entity *"five three ..."*, a simple classifier over LSTM outputs would categorize this as a *bank-account-number*. *Attention mechanism* [3] helps in such cases by associating entity with the correct set of words by paying more attention to them. Past works [4] have shown that Attention-based LSTM Networks improve performance on Classification tasks.

2.3. Audio & Transcript Masking

An audio recording is modified by masking the sensitive segments, identified using the text-to-audio alignment and extracted sensitive entities. The objective is two fold, to mask the sensitive information from both the transcript and the audio stream, and to ensure that rest of the information in the call remains intact as erroneous masking could limit the contribution of a call for the purpose of training and drawing insights.

3. Conclusions & Key Application

Heuristic Engine helped convert the sequence tagging nature of entity extraction into a classification task. Inference stage latency was twice faster compared to sequence tagging, since sequence tagging involves predicting entity label for every token in the transcript, but with heuristic engine, we only predict entity labels for identified boundaries using the classification model. We demonstrate that a combination of heuristics generated boundaries and a classification model is effective for listed entities in masking sensitive information in practical applications. Additionally from operational point of view, classification models are also much easier to integrate in production pipelines.

Observe.AI allows call centers to search, analyse and check historical calls. Compliance is ensured for all the calls on the platform by masking sensitive information from both the audio and transcript.

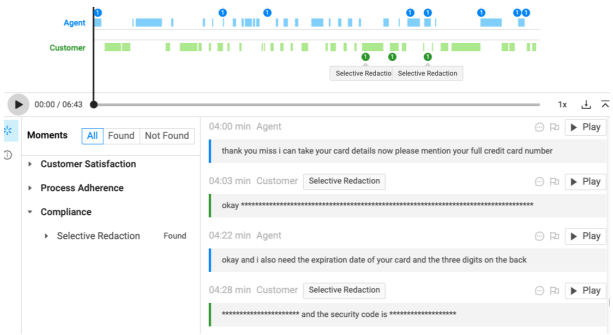

Figure 3: *Observe.AI app shows a call masked for card details*

4. References

[1] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *EMNLP*, 2014, pp. 1532–1543.

[2] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation, 9(8)*, pp. 1735–1780, 1997.

[3] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *NAACL*, 2016, pp. 1480–1489.

[4] Y. Wang, M. Huang, L. Zhao, and X. Zhu, "Attention-based lstm for aspect-level sentiment classification," in *EMNLP*, 2016, pp. 606–615.