RELI Group

White Paper

Center for Medicare and Medicaid Services Quality Assurance Monitoring

Autonomous Identification of Call Topics, Based on Machine Learning and Artificial Intelligence Topic Clustering of Audit Call Logs from Medicare Administrative Contractors

By Vj Dwarapudi, Vice President of Technology

Published August 1, 2022

Center for Medicare and Medicaid Services (CMS) Quality Assurance Monitoring (QAM)

Anonymous Identification of Call Topics, Based on Machine Learning (ML) and Artificial Intelligence (AI) Topic Clustering of Audit Call Logs from Medicare Administrative Contractors

Background:

The current management at system contact centers is seeking to achieve better results and audit volume, improving operational productivity and effectiveness. Currently the program is administered manually, and the wide range of issues such as appeals, enrollment, claims, fraud and abuse, and medical necessity are monitored on a manual oversight basis.

RELI Group's objective is to both help identify the issues at play and introduce the corresponding ML and AI functionality as a solution to help enhance the audit processing capability within CMS.

Problem Statement:

In the industry, today's contact/call centers are awash and abound with call logs and workforce integration on call log content. Much of their voice content is in in the nature of claims, complaint processing platforms and automated voice response systems. Thankfully this need has created an opportunity to extract value. There are several innovations in the opensource arena like DeepSpeech, Kaldi, SranfordSpeech, Wave2Letter, PyTorch SpeechBrain, Google, Azure, AWS, and Oracle OCI that are part of this solution using ML and AI to address various unique segments/requirements. Our intent is to bring that capability to help enhance the CMS QAM.

In CMS' specific case, there is a large amount of manual review. The contact centers are seeking opportunities to leverage manual reviews to help improve productivity and effectiveness within the program by understanding the categories of call segments which limits the issue to the single call category issue.

Currently in the QAM program, as part of the oversight, the review focus process involves manually scoring quality of interactions for channels including adherence to privacy, knowledge skills and customer skills. The QAM’s metrics include privacy, knowledge skills, accuracy, and completeness around the quality of customer service rendered during the process of addressing complaints at the MAC.

However, more can be done via ML/AI to extract trends, since call knowledge is not captured. First, the core problem is that the majority of the calls are grouped in one or two primary topic areas, while in reality there over 200 categories/topics. Second, a key issue in contact centers is that some information is lost/gets stuck in routines of manual firefighting, preventing process upgrades to handle the needs of modern customers. Third, as some of today’s customers tend to call in with increasingly complex queries, it isn’t always possible to provide an immediate answer.

As identified by CMS and QAM management, we believe this is a good first step to enhance the call center audit process. This whitepaper outlines an approach to classify topics in audit calls using ML/AI capabilities, allowing CMS to understand and identify clusters of topics and predict trends in customer concerns.

Approach and Methods

For CMS, there is an opportunity to apply automation on audit calls for topic identification and topic clustering, and possibly predicting trends in call issues, supplemented by readily available open-source platforms. As such, well-configured ML and AI capabilities will boost the audit volume processing and automation through gathering information, actions taken, and the responses. CMS can leverage ML and AI to create more insightful models to use in the call center. This ability is made possible by natural language processing (NLP) the speech-focused segment of the AI industry, as well a, sentiment analysis and speech-to-text conversions through ML and AI algorithms. This will allow CMS to potentially uncover issues sooner or intercede before there is a significant negative business impact. Below we describe our approach and the iterative process in building these insights.

To keep these solutions open and portable, we present two options to implement this capability in a custom or turn-key build, while discussing NLP components, data flows, consolidation under CMS AWS/ Oracle OCI Cloud, Actors/Stakeholders, and process.

Our approach involves deriving context from the call logs, enabling preprocessing to provide a general idea of the context of the conversation. Using this data, the ML pipeline can be constructed to have a view of the customer’s area of concern. Historically. the context of the call log should stay constant, showing the customer’s general area of concern.

Once we derive the context, we will data mine the context to create content “density maps” or content “centroids,” which are open and freely available in various forms in the open-source domain. This step allows us to establish the frequency algorithm, in which we can data mine the various categories of interest for the CMS call groups.

Between the two, we can now map the context to the categories and produce a larger subset of calls and call groupings, and apply the following across the logs with the cooperation of the MACs:

- Clustering: A machine learning technique that groups general log data points together.
- Reason Classifiers: Classification is the process of predicting the class of given data points. The reason classifiers are those data points that help the machine learning model output recommendations.
- Grouping Codes: Groupings are codes that describe the reasons for a customer calling customer support.

Since this is not a single shot model and to prevent vendor lock-in while extending portability of the solution, RELI has provided both a custom and turn-key option.

For the custom option, a small team of developers and a current analyst on the program will build this custom iterative model to grow and improve the accuracy over a 30- to 60-day period to deliver the models as listed in the roadmap below. We ask for the 30- to 60-day period to create and optimize the model, because if left as an unoptimized model, the mappings could become divergent or irrelevant. We achieve this via a supervised learning model. While the AWS/OCI or other AI-powered assistants are leveraging the context of the conversation to create better groupings, we will review and reiterate through any corrections as needed.

To build the model, the data logs flow through the collective set of algorithms, including state machines, rules to extract semantics, and words & discourse in NLP.

The process to extract implicit metadata and caller topics differs between supervised and unsupervised models. To assign implicit metadata, variables or topics to each sentence or corpus (text from the call), each sentence is categorized according to the predetermined schema of modality and topic or subject.

In the supervised model, the first text classification task involves manually classifying a set of training documents in preparation for feeding the automatic system. The next step is to take these manually classified documents and process them through the trainable text classification system. During the process it builds a set or vector of terms, phrases or sentences, and entities extracted from the text.

Fortunately, as a fall back, there’s also option 2 where there are several other readily turnkey alternatives provided by third parties who deal with similar problems and have completed several of these steps.

Some pre trained models, bearing just the model (without content) are also available in the industry on a plug-in/turnkey basis. These models have been pre trained and can be plugged in, though it is essential to choose the correct fit and expend some effort to configure and set up the environment vs. coding it from scratch. Using one of these options and the learning corrections (see roadmap below), RELI Group will repeat the entire grouping, centroid or density frequency determination and mapping process in the roadmap via the unsupervised learning process.

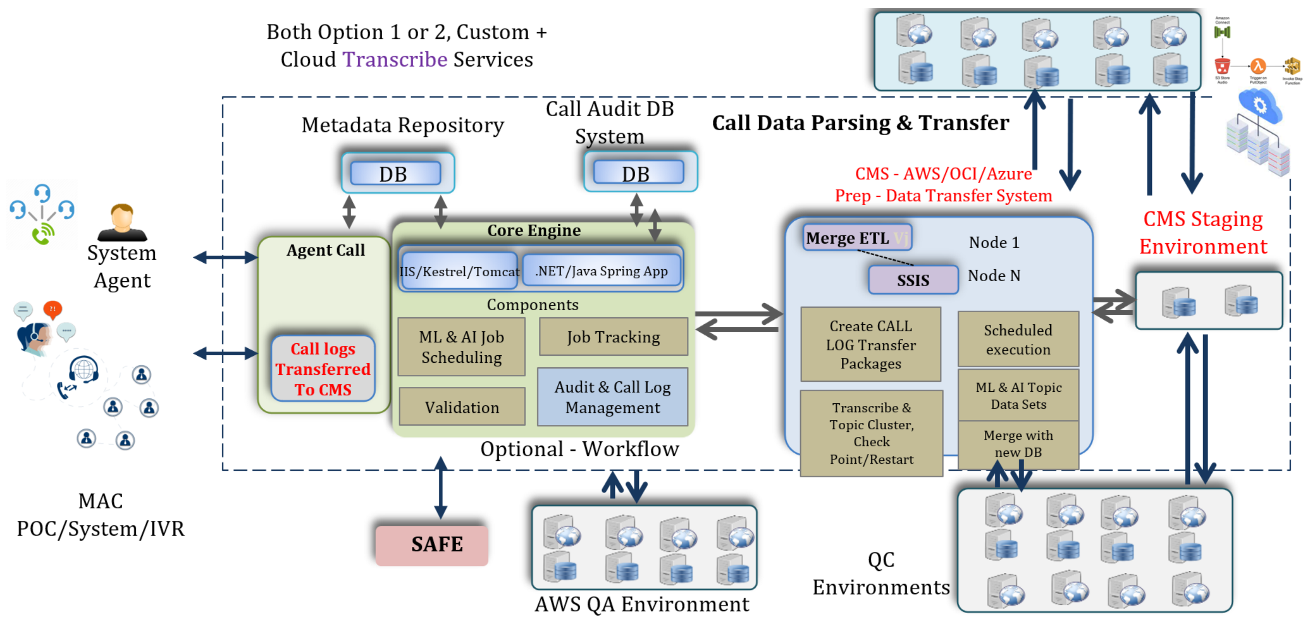
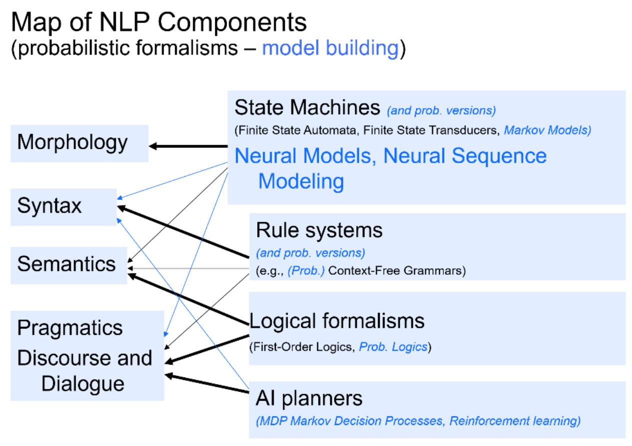


Figure 1: Proposed draft system architecture


Map of NLP Components
(probabilistic formalisms – model building)
Morphology
Syntax
Semantics
Pragmatics Discourse and Dialogue
State Machines (and prob. versions)
(Finite State Automata, Finite State Transducers, Markov Models)
Neural Models, Neural Sequence Modeling
Rule systems
(and prob. versions)
(e.g., (Prob.) Context-Free Grammars)
Logical formalisms
(First-Order Logics, Prob. Logics)
AI planners
(MDP Markov Decision Processes, Reinforcement learning)

Below is a full sample of the functional code on tokenization. The design would typically be rolled up as a microservice in Lambda.

```
from nltk import word_tokenize
from collections import defaultdict

def QAMS_Audit_High_Frequency_Topics(corpus, top_x,
skip_top_n):
 QamsCallWordsCount = defaultdict(lambda: 0)
 for c in corpus:
 for w in word_tokenize(c):
 QamsCallWordsCount[w] += 1
 QamsCallWordsCount_tuples = sorted([(w, c) for w, c in
QamsCallWordsCount.items()], key=lambda x: x[1],
reverse=True)
 return [i[0] for i in QamsCallWordsCount_tuples[skip_top_n:
skip_top_n + top_x]]


def Qams_Convert_Top_Audit_Words_To_Vectors(corpus,
top_x):
 topx_dict = {top_x[i]: i for i in range(len(top_x))}

 return [
 [topx_dict[w] for w in word_tokenize(s) if w in topx_dict]
 for s in corpus
 ], topx_dict


def filter_to_top_x(corpus, n_top, skip_n_top=0):
 top_x = QAMS_Audit_High_Frequency_Topics(corpus, n_top,
skip_n_top)
 return Qams_Convert_Top_Audit_Words_To_Vectors(corpus,
top_x)
```

Table 1: Example Audit Tokenization

The approach at its core hosts a Probabilistic Language Model and a trained ML model that represents the QAMS CMS cluster topics.

The model has to be developed using Tokenization, Vector Semantics and Embedded Word (Topic) calculations based on information/data sources that come in from the primary actors or stakeholders as identified below. We may also ask for a refined topic definition list in the process.

Primary Actors and Data Sources

- CMS
- MACs
- Caller
- CSR
- CMS Standardized Provider Inquiry Chart
- Data Collection Tool (Selected Technology)

- Data Analysis and Reporting Tool (Selected Technology)

This beneficial approach allows CMS to centralize all the various MAC silos/logs under the same solution, while leaving options to upgrade future enhancements in a portable way.

The current audit process flow from our solutions understanding is listed below.

Pre-Conditions/Triggers

- Precondition – The caller must be routed to a CSR. A CSR must be logged in and in Ready mode.
- Trigger –The caller is connected with the CSR and a discussion is started.

Basic Flow

- The caller dials into the MAC toll-free number.
- The system sends the caller to the IVR.
- The caller enters the authentication elements in the IVR.
- The IVR provides options for the caller to select.
- The caller selects an option.
- The IVR provides the information to the caller.
- The caller has questions about the information received from the IVR and selects to be routed to a CSR.
- The system routes the caller to a CSR.
- The caller is connected to a CSR.
- The data collection tool starts to capture the call.
- The CSR speaks with the caller, checks, uses, other databases and systems, logs the call in the CRM.
- The caller or the CSR disconnects the call.
- The data collection tool completes the capture.
- The data collected is analyzed and a report showing the inquiry category, subcategory, and type is created.
- The report is sent to CMS.

Alternate/Exception Flows

- The caller is satisfied and does not need to be routed to a CSR. The use case does not start.
- The caller disconnects the call before being routed to a CSR. The use case does not start.

Post-Conditions

- The data is analyzed and a report showing the inquiry category, subcategory, and type for each call is created.
- The report is sent to CMS.

- The data and reports are stored.

Execution Roadmap

Core program functionally is performed in the following sections. To ensure CMS has oversight, RELI Group will also create and share a project plan with timelines at kickoff based on the deliverables

Supervised Learning of Call Topic Clustering

Using matching algorithms, this section consists of a target/outcome variable (or dependent variable), which would be a CMS topic density. Using the understanding of the topic, we try to deduce or predict the topic variable based on a given set of predictors (independent variables); this is currently a gap and not well understood. Collectively these variables allow us to generate a function that maps the inputs to desired output topic of CMS.

Our supervised learning and model training process continues until the model output provides us reliable or comfortable accuracy rates based on the training data.

Some open-source algorithms we can apply for include Centroid, Regression, Decision Tree, Random Forest, KNN, Logistic Regression and more.

Unsupervised Learning of Call Topic Clustering:

This rapid approach involves selecting algorithms to blitz through the data and allow unsupervised deductions. Here we do not prescribe any target topics or outcome variables to predict or estimate; rather, the model based on the data profile spits out trends it discovers. It is often used for clustering populations in different groups and leveraged for insights into customer segmentation into different aggregates. Some open-source algorithms we can apply for include Apriori algorithm and K-means.

Reinforced Learning of Call Topic Clustering

This approach and algorithm allow a lot more control, where we can focus the learning of the machine and the machine is trained to make specific deductions and decisions. We repeatedly reexposd the data to the ML models in an environment where it trains in a focused manner, with self-driven feedback and continuous trial and error outcomes.

We typically intervene to guide and pick the models we feel is the right outcome, allowing the machine to understand the desired outcome as it learns from past experience and uses the decision trees and logic that allow it to reach a certain outcome. It then reinforces, or gives more weight to, co-related models to capture the best possible knowledge to make accurate business decisions. Some open-source algorithms we can apply are the Reinforcement Learning use Markov Decision Process.