Paper 3141-2015

Unstructured Data Mining to Improve Customer Experience in Interactive Voice Response Systems

Dmitriy Khots, Ph.D. West Corporation

ABSTRACT

Interactive Voice Response (IVR) systems are likely one of the best and worst gifts to the world of communication, depending on who you ask. Businesses love IVR systems because they take out hundreds of millions of dollars of call center costs in automation of routine tasks, while consumers hate IVRs because they want to talk to an agent! It is a delicate balancing act to manage an IVR system that saves money for the business, yet is smart enough to minimize consumer abrasion by knowing who they are, why they are calling, and providing an easy automated solution or a quick route to an agent. There are many aspects to designing such IVR systems, including engineering, application development, omni-channel integration, user interface design, and data analytics. For larger call volume businesses, IVRs generate terabytes of data per year, with hundreds of millions of rows per day that track all system and customer- facing events. The data is stored in various formats and is often unstructured (lengthy character fields that store API return information or text fields containing consumer utterances). The focus of this paper is the development of a data mining framework based on SAS® that is used to parse and analyze IVR data in order to provide insights into usability of the application across various customer segments. Certain use cases are also provided.

INTRODUCTION

Contrary to common opinion, the voice channel is not dying. While it is true that other channels such as mobile and web are growing in usage, even the die-hard DIY'ers need to speak with a customer service representative when they're signing up for service or when a product is not working, and even more so when they have questions about charges on their bill. And, of course, there is that demographic that still likes to pay their bills using the phone or even find out where the closest store location is. Fact of the matter is – the voice channel is here to stay, while the "omni-channel" environment is continuing to grow. Enterprises adapt to this by rebranding their call centers into "contact centers" and establishing best-in-class customer interaction platforms in order to acquire, care for, and retain their customers.

The front end to a customer interaction is typically an IVR system. The purpose of an IVR system is to automate as many routine tasks as possible and allow the customer to reach an agent if needed. West Corporation develops and supports state-of-the-art IVR systems that not only serve customers within the voice channel but also act as pivot points for other channels (e.g. SMS). West IVR is also integrated with necessary customer systems that in real time help the IVR application to identify callers and treat them appropriately based on their unique service settings, their intent or reason for the call, and past interactions.

All said interactions generate a sizeable amount of data that by today's standards is easily classified as "Big". An enterprise with 20 million subscribers generates around 500,000 calls per day, with up to 300 events (both system and consumer generated) per call. Some of those events are API calls that retrieve information from the enterprise Customer Relationship Management (CRM) system and can have 1000+ key-value pairs (KVP's). Thus, IVR data is both wide and deep. Such data is a mecca for data miners and a nightmare for the warehousing teams who have to manage such volumes in real time conditions and support historical reporting.

Another big challenge with IVR data is that most IVR processes are highly dynamic – KVP's change on the fly, new prompts are added and other prompts removed. Engineering teams at West have created event logging mechanisms that are invariant to changes and can record data very quickly, however, the tradeoff there is most of it is now unstructured.

The purpose of this paper is to provide an overview of a Data Mining Framework that has been created by the advanced analytics team to deal with the above challenge in order to provide data-driven actionable insights for optimizing performance of IVR applications.

IVR SYSTEM OVERVIEW

IVR systems typically consist of the following components: welcome messaging, caller authentication, intelligent decisioning (such as predictive intent – knowing why the customer is calling, VIP treatment, collections, etc), main menu, and functional areas (such as billing or repair). West IVR is equipped with Natural Language processing that allows the system to leapfrog the customer from the main menu to the right spot within the flow thereby avoiding unnecessary questions and reducing customer effort. The customer can also opt to speak with an agent at every step throughout the flow. Figure 1 shows a simplified IVR call flow.

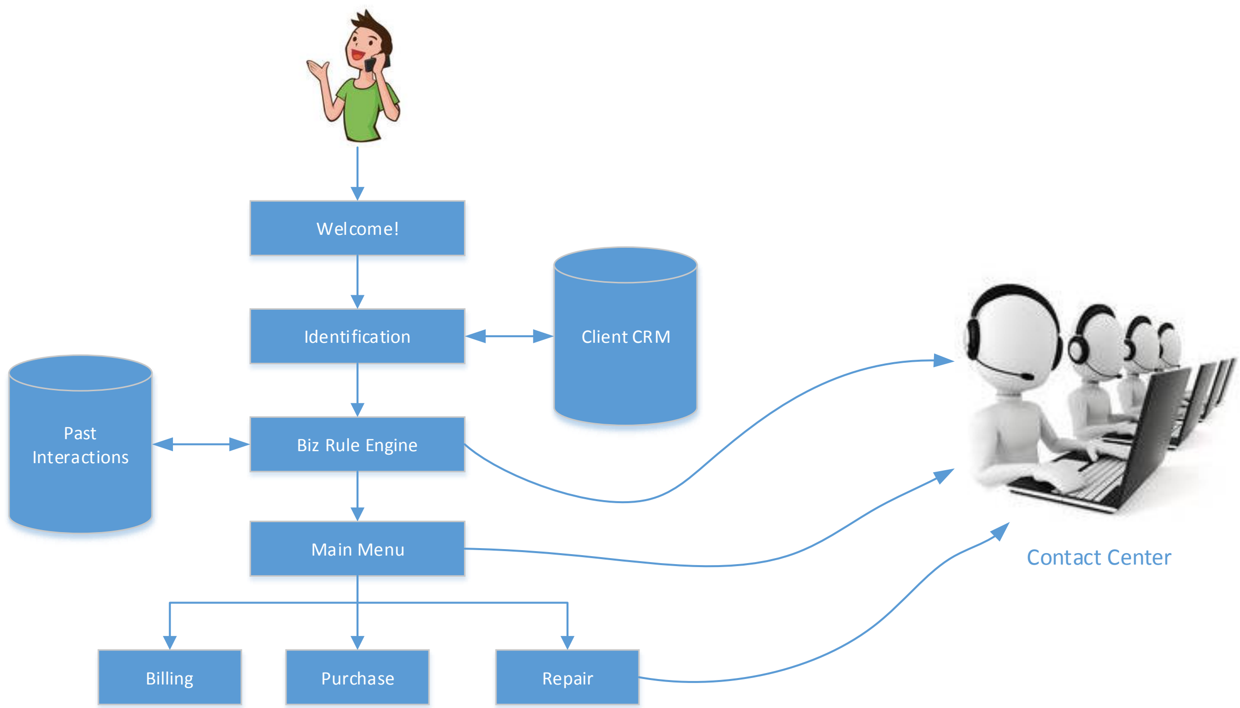

Figure 1. Simplified IVR Call Flow

Customer authentication is a critical component of IVR flow because most self-service (or automation) options are available only when the system knows who the customer is. In most cases the carrier (phone company) will present the customer's phone number to the IVR system. The system then performs a dip into client CRM using a particular API to see if the phone number is associated with an account. If an account is returned, the system asks the customer to confirm certain pieces of their identity to fully authenticate them and the call proceeds. If there is not an account found, which does happen occasionally, the IVR will ask the caller to provide their account number and if still unsuccessful it will use other tactics such as zip code lookups or data aggregator hits to ensure the number of authenticated callers is maximized.

Once the caller is identified various processes take place to intelligently decision the caller. Collectively these processes are referred to as the "Business Rules Engine". This engine uses CRM data retrieved by the API along with past interaction database information to provide the customer with the best experience possible. For example, VIP customers will immediately be routed to a high touch agent group, or a concierge, while customers who had contacted the brand previously or recently will be asked if they are calling for the same reason in order to be expeditiously routed to appropriate service area. On the other hand, customers who are in collections may be asked to pay for their services upfront.

All of the above takes mere seconds of processing time. If no special rules are triggered by the caller, the call proceeds to Main Menu, where the customer is asked why they are calling. Most West IVR systems are equipped with a Natural Language engine with a touch tone fall back. This means that the customer can speak freely into the system and it will recognize what the customer is saying and will place this caller into the appropriate branch within the flow. If the customer is in a noisy area or is not comfortable speaking their account information, the system is also equipped to receive touch tone input by letting the customer simply press buttons on their phone.

Depending on the brand and integration levels, the IVR provides self-service for various routine tasks such as updating one's credit card number for automatic payments or paying a bill or ordering a product or even some routine troubleshooting like resetting a modem. About 40% to 50% of customers end up successfully completing transactions inside the IVR or receiving useful content about their account.

Cost per IVR call is negligible compared to agent handled calls and thus it is always a goal for IVR to maximize *meaningful* containment (number of calls resolving inside IVR / total number of calls) while allowing customers to transfer to agents to provide for high customer satisfaction and loyalty ratings (such as Net Promoter Scores). It is this fine balancing act that drives the need for data mining of IVR data in order to prescribe actions that will maximize meaningful containment by identifying more callers, predicting why they are calling, and offering a personalized experience that will create customer loyalty, resolve issues, avoid repeat calls, and even generate more sales.

DATA OVERVIEW

The following data categories are captured on each interaction (or call) by the event loggers:

1. **Who** interacted or called?
2. **When** they interacted?
3. **Why** they interacted?
4. **What** happened during the interaction?
5. What was the **channel** of interaction?
6. What was the **direction** of the interaction?

Who interacted is a more loaded question than it seems. It is not just the phone number or account number and name and address of the customer. It is potentially the 1,000+ KVP's that are retrieved back regarding this customer. This includes customer start dates, what types of services or devices they have, billing history, customer segmentation scores, etc.

When the customer interacted is simply the date and time stamp of each interaction.

Why the customer interacted has explicit and implicit data points. Explicit data includes options pressed by the customer at main menu or what they have said during the Natural Language prompt, which is another source of unstructured data. Implicit data includes inferred reasons for the call, such as the fact that their bill is due today or that they had called twice within last hour about a technical problem. Identifying these is also part of the data mining exercises undertaken by the advanced analytics team.

What happened during the interaction is the entire stream of events that had occurred during the interaction. For example, all of the prompts that the customer heard and what options they pressed, how long was the interaction, whether they had transferred to an agent, etc.

Channel of the interaction is Voice/IVR and potentially SMS, while **Direction** is either inbound or outbound. The IVR can perform proactive outbound outreach to customers based on various triggers and then interact with them should they pick up the phone.

The data is captured as the interaction progresses and events are written into a logger which then feeds a variety of databases. One is an operational data store used for real time reporting and another one is a warehouse used for historical data storage. The SAS system at West uses SAS/ACCESS TO ODBC to connect to the storage and the analytics team uses pass – through SQL (using SQL Procedure or PROC SQL) to retrieve the data while outsourcing major queries to the database.

DATA MINING FRAMEWORK

The data mining framework to support enhancements to IVR systems consists of two layers: development and production.

DEVELOPMENT LAYER

The data mining framework development layer is shown in Figure 2.

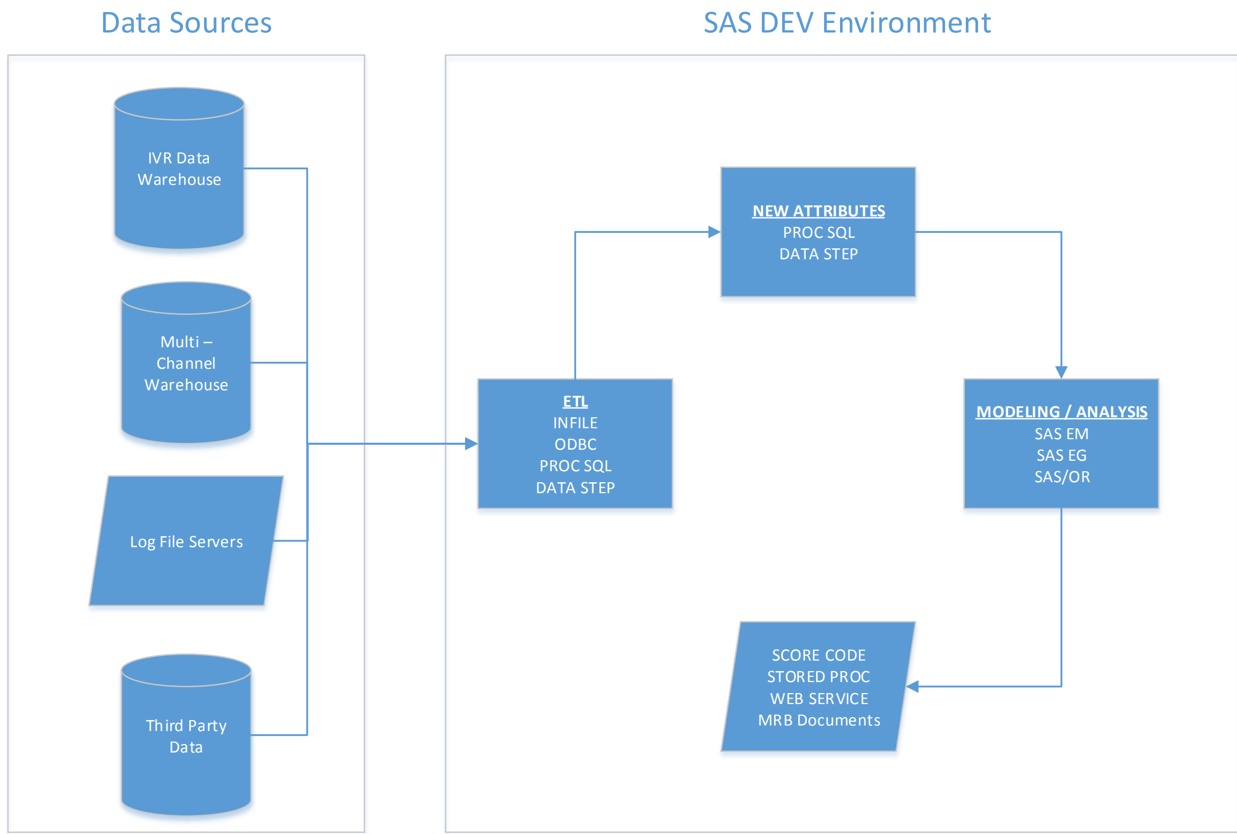

Figure 2. SAS Data Mining Framework – Development Layer

There are four major inputs of data into the development environment. First is the IVR Data Warehouse that contains all IVR related data mentioned in the previous section in third normal form (thus significant SQL skills are required to put it into a format acceptable for data mining purposes). Second data source is the Multi-Channel Data Warehouse that contains non-voice interactions such as SMS and keys that can tie each non-voice interaction to the IVR interaction (which are available since IVR is a pivot point for the SMS interaction). In certain cases, a third IVR data source is required. It is a log file server which contains data in raw text form which either has not yet been incorporated into the IVR Data Warehouse or has been created for special nonstandard purposes. Finally, the fourth data source is Third Party Data, which typically includes various demographics datasets (at ZIP, ZIP+4, or even household levels) or other auxiliary data such as product and serviceability lists. For development purposes, historical data may be analyzed for as little as just a couple of hours ago to as long as years, depending on the application.

The data is ingested into Windows based SAS environment either via PROC SQL using pass-through SQL facility and SAS/ACCESS TO ODBC or using a DATA STEP with an INFILE statement. It is imperative that proper and updated versions of ODBC drivers are installed within the Windows environment. The INFILE option is used to load raw data logs over the secured network. One of the biggest advantages of using pass-through SQL is the ability to outsource processing such as joins and calculations into a typically much more powerful database, rather than performing the same tasks after the fact working with SAS datasets. For example, consider the need to understand impact of repeat

callers (total call volumes along with transfers to agents) across time, different clients and different regions within those clients. Caller is typically identified by their ANI (automated number identification or simply the phone number where they called from). The following code provides the best solution for this task:

```
PROC SQL;
CONNECT TO ODBC as myODBC (NOPROMPT= "DSN=ODS");
CREATE TABLE report_input AS
SELECT * FROM CONNECTION TO myODBC
(SELECT

ivr.start_date,
cli.client_number,
srm.division_identifier,
ivr.ani,
count(distinct ivr.wic_ivr_key_identifier) as calls,
count(distinct case when tt.transfer_type_identifier = 1 then ivr.wic_ivr_key_identifier end) as xfrs

from txl.Transaction_call_fact ivr
left join txl.transaction_client_dimension srm on (ivr.wic_ivr_key_identifier=srm.wic_ivr_key_identifier)
left join txl.client_division cd ON (cd.division_identifier = srm.division_identifier)
left join ods.program_apn apn on (ivr.program_apn_identifier = apn.program_apn_identifier)
left join ods.ods_client cli on (apn.client_identifier = cli.client_identifier)
left join txl.call_exit_type cet on (ivr.call_exit_identifier = cet.call_exit_identifier)
left join txl.call_type ct on (ivr.call_type_identifier = ct.call_type_identifier)
left join txl.transaction_transfer_fact ttf on (ivr.wic_ivr_key_identifier = ttf.wic_ivr_key_identifier)
left join txl.transfer_type tt on (ttf.transfer_type_identifier=tt.transfer_type_identifier)
where cli.client_number in (111111,222222,3333333)
and ivr.start_date >= '10-mar-2015' and ivr.start_date <= '31-mar-2015'
group by ivr.start_date, cli.client_number, srm.division_identifier, ivr.ani) ;
QUIT;
```

This is a join across nine tables necessary due to the third normal form of this data. The other option would have been to retrieve each table separately and then join everything in SAS, however, that would have taken a much longer time due to the fact that all database benefits such as speed and indexing would have been lost.

Once the data is ingested into SAS, the analytics team performs typical CRISP-DM steps. Data is explored for abnormalities and outliers, new attributes are created from existing attributes, and data is loaded into SAS Enterprise Miner for predictive modeling activities. Upon completion of all modeling tasks, the model is taken through a Model Review Board, which is a peer reviewed internal QA process that eliminates any potential issues and tests it for statistical rigor. The analyst responsible for the model then creates either a score code, which depending on the production schema can be in SAS or in another language such as JAVA, or a SAS stored process which is then exposed to the IVR system via a web service.

PRODUCTION LAYER

There are three options for a production layer of the SAS Data Mining Framework. The first is real time scoring / processing, where the production system communicates with the SAS Scoring Engine via a web service or API. The second is batch processing, where the production system and the SAS Scoring Engine are exchanging files via a batch staging process using SFTP or ODBC protocols. The final option is when the SAS generated score code is embedded into the existing Business Rules Engine (discussed above) and may not be in SAS format, but can be in JAVA or another language that's interpretable by the Business Rules Engine. The three options for the data mining framework production layer are shown in Figure 3.

Figure 3. SAS Data Mining Framework – Production Layer

In the first two cases, once the development of a data mining solution (such as a predictive model) is complete, the score code generated by SAS Enterprise Miner is added into the SAS Scoring Engine, which is typically equipped with model management processes from basic production execution reporting to more advanced early warning detection systems that track deviations of important variables and model statistics such as KS or R^2 and alert the user of abnormal changes. Real time scoring is possible using SAS Integration Technologies which exposes the SAS Scoring Engine to the outside world via a Web Service and using procedures such as PROC SOAP.

Batch scoring is accomplished using a work stream scheduler that executes a series of operations and ensures smooth transition from step to step. In one embodiment, the IVR system writes order events into a database table which are then ingested into the SAS Scoring Engine via ODBC on a daily basis and executes batch score code within the SAS Scoring Engine which regenerates association rules and writes the recommendations back into the database via ODBC. Additional information on this and other use cases are presented in more detail in the next section.

Embedded scoring is accomplished without the use of a SAS Scoring Engine. Once the model is built in SAS Enterprise Miner, the SCORE node generates JAVA score code and since the existing Business Rules Engine is equipped with a JAVA interpreter, the score code is simply embedded into the Business Rules Engine. Usage of either of the three methods described above depends on each specific use case. If the model logic is fairly simple and it does not rely on a lot of historical data to make a decision, embedded scoring method is preferred. For use cases requiring more complex processing, batch or in

some cases real time scoring methodology is preferred. The next section discusses some use cases that utilize these methodologies and provides more details on the data mining aspects of unstructured data analysis.

USE CASES

There is a vast amount of applications of data mining and advanced analytics to optimize IVR and multi-channel performance and customer experience. This section provides a cursory overview of some of the recent developments in the works by the Advanced Analytics team at West Corp.

USE CASE I: IMPROVING PATIENT HEALTH BY OPTIMIZING TECHNOLOGY ENABLED CARE FOR HYPERTENSIVE PATIENTS

Affordable Care Act also known as Obamacare has brought forth a tremendous amount of changes within the US healthcare system. There are now significant financial and goodwill benefits to health related organizations to ensure that populations under their coverage are healthy. One of the main practices to ensure sustained patient health is health management. Traditionally, this is accomplished by doctors and nursing staff by providing recommendations to their patients on best practices related to good health (e.g. don't smoke, check your eyes and feet if you have diabetes, measure your blood pressure regularly if you are hypertensive, etc.). West Corporation helps organizations responsible for patient health by making it easier for nurses to reach their patients, by using automated outbound notifications, and by allowing patients to submit their metrics via the IVR system. This process holistically is called Technology Enabled Care or TEC.

This unstructured data mining case is specifically for patients with hypertension whose blood pressure readings need to be evaluated by the medical staff on a regular basis. Patients are automatically notified by the system when the blood pressure readings have not been captured. When the patient picks up their phone they can enter their systolic and diastolic readings along with pulse into the automated system. If it is not available immediately, they can measure it and call back later to enter the readings. Once a reading is captured (either on outbound notification itself or on a consequent inbound call), these readings are sent to a patient health management database via an API.

The benefits of such a system are clear. Nursing staff can now reach a lot more patients than they could manually and can act faster if there are abnormally high readings submitted. Since this product is relatively new, it is critical to understand adoption, how many patients respond to the notifications, and how many enter their blood pressure readings on the notification itself or within a certain amount of time (e.g. 4 days or 14 days).

The data involved in this process is semi-structured. It has structured call results and unstructured API return (a set of KVP's). More specifically, the data has the following fields:

- Call ID,
- Call Direction,
- Call Result,
- Patient ID,
- Call Datetime,
- XML Payload.

The last field contains the API return data, which is a flattened out XML string. Here is an example (with sensitive data removed):

{"dataItemMainMenuSelected":"blood_pressure","dataItemCallType":"Inbound",
"dataItemCallPath":"Unknown caller", "ccName":"", "callDuration":"189",
"callPath=":null, "dataItemMadeMedChange":"", "dataItemAniSource":"Telco",
"ccPhoneNumber":"", "dataItemAPN":"8117472326","callDisposition":"",
"ccId":"","dataItemFinalAttempt":null, "dataItemTransfer":"N","dataItemDOB":"NNNN-NN-NN",
"dataItemShortCall":"N",
"dataItemWeight":"2014-05-26,AM,143|2014-05-26,PM,143",

"dataItemDateTime":"2014-05-27 13:02:11",
"dataItemExperiencedSideEffectsFromMedChange":"",
"callId":"e3412-20140527130211-629","patientName":"LASTNAME, FIRSTNAME",
"ccSiteName":"SITENAME",
"dataItemPulse":"2014-05-26,AM,56|2014-05-26,PM,56",
"dataItemDataFor":"1", "dataItemANI":"5555551111",
"dataItemBP":"2014-05-26,AM,120,67|2014-05-26,PM,119,76"}

A cursory overview of this string reveals that this is a varchar string, meaning that depending on how many readings the patient decided to submit, the length of this string will change. Now, if the goal is to measure how many morning blood pressure readings this patient submitted over time, then the following code accomplishes this task:

```
PROC SQL;
   CREATE TABLE patient_summary AS SELECT
   Patient_ID,
   count(SUBSTR(xmlpayload,index(xmlpayload,'dataItemBP'),100), 'AM' ) as
   AM_BP_Count
   FROM CALLDATA
   WHERE Call_Datetime > "&startdate" AND Call_Datetime < "&enddate"
   GROUP BY Patient_ID
   ;
QUIT;
```

This is a good example that manual inspection of the data, also known as PROC EYEBALL, is an important step in creating the code. It can be observed that the blood pressure readings are given as the last KVP in this XML, thus we can use the INDEX function to find the first occurrence of keyword *dataItemBP*, then apply the SUBSTR function to all content after the appearance of this keyword and finally use the COUNT function to count the number occurrences of keyword *AM*. Note that this code would be considerably more complex if the *dataItemBP* was not the last KVP. Using powerful methods embedded in SAS, parsing out strings becomes easy and all the business questions regarding adoption and effectiveness of this system can now be answered.

Now that this process has been measured, the text step for this use case is to optimize the outreach success rate by finding optimal time of day and frequency of contact. This is included in future work for this use case.

USE CASE II: IDENTIFY MISROUTES BY EXPLORING CHANGES IN IVR ROUTING DECISIONS ISSUED BY THE AUTOMATED CALL DISTRIBUTION SYSTEM

One of the key performance indicators in the call center is the percent of agent-to-agent transfers. It is typically used as a proxy for customer satisfaction – the more the customer gets bounced around between agents the worse the experience. There are typically two main drivers for agent-to-agent transfers. The first one is agent skill and poor training, while the second is improper routing by the automated phone system. Routing errors (or misroutes) generated by the automated phone system have two broad categories – an untuned IVR system that is not making accurate routing decisions and the inaccurate routing decisions issued by the automated call distribution system (ACD).

When agent-to-agent transfer rates are high, troubleshooting usually starts by exploring connections between IVR and ACD and whether ACD overrides the IVR routing decisions. This use case is about exploring these associations by parsing through unstructured data. The production process works as follows:

1. Customer opts to speak with an agent in IVR
2. IVR sends user information and location within the flow to a Transfer Database

3. Transfer Database returns the appropriate Transfer Location for the specifics of this call (e.g. if the customer attempted to pay their bill, but their credit card got declined, the Transfer Location may be “Billing – Unsuccessful Payment”).
4. IVR sends the Transfer Location to the ACD via an API hit.
5. ACD returns “OK” or “CHANGE ROUTING” decision back to IVR within the same API return.

From a database perspective, all the API send and receive data is stored as unstructured KVP’s in a single varchar field. Below is an example of the mentioned steps using data elements.

1. API request to get appropriate transfer location:

<send>param0=EIVR|param1=RES|param2=2|param3=10|param4=29|param5=3|param6=21:09|param7=en-us|param8=|param9=SalesRetention|param10=Sales
</send>

2. API return with the appropriate transfer location (given by a 12 digit number):

<receive>col0=FOUND|col1=0|col2=0|col3=100|col4=OFF|col5=ON|col6=Array|col7=OFF|col8=795028071136|col9=SIP|col10=18|col11=4|col0=FOUND|col1=0|col2=0|col3=100|col4=OFF|col5=ON|col6=Array|col7=OFF
</receive>

3. API request from IVR to ACD to notify about the upcoming transfer:

<send>timestamp=2015-03-25T21:09:47-04:00| sourceSystemId=1| sourceSystemUserId=1 |sourceServerId=1|trackingId=035034f0b3054250bb0f41105bf5b3aa|applicationName=WESTEIVR|telephonyUID=035034f0b3054250bb0f41105bf5b3aa|ani=5555551111|dnis=3036901|language=ENG|enteredZipCode=12345|hierarchyLevel=REGION|hierarchyName=REG_NAME|accountNumber=|accountType=RESIDENTIAL|browserSessionID=|browserHostName=|callFlowPoint=ROUTING_LOOKUP|key=Module|value=SalesRetention| term=795028071136
</send>

4. API return from ACD to IVR indicating agreement with the routing decision:

<receive>extAttrRtgAction=NO_ACTION|dialString=795028071136|transferType=SIPREF_UUI|attributeRoutingUID=316786844|key=BROWSER_GROUP_NAME|value=DEFAULT_BROWSER_GROUP_NAME|key=EXT_ROUTING_ENGINE_ACTION|value=NO_ACTION|key=EXT_ROUTING_SERVER_NAME|value=NED_ATTR_ROUTER_B|key=EXT_ROUTING_SERVER_UID|value=316786844
</receive>

Note that send and receive data are typically stored in different tables. In order to measure how often ACD agrees or disagrees with IVR routing decisions, SAS coding with string parsing similar to the first use case is required. In one particular example, it was found that ACD disagreed with IVR 60% of the time. This was a very high ratio which led to an audit of both systems to understand which systems decisions were accurate.

USE CASE III: IMPROVING TITLE RECOGNITION IN AUTOMATED PAY-PER-VIEW ORDERING SYSTEM

West offers customers of various entertainment companies the flexibility to order Pay Per View (PPV) movies and sporting events using IVR and SMS channels. In particular, when a customer wishes to order a PPV, they can text keywords such as “MOVIES” or “PPV” to a specific short code using their mobile phone and the system will start interacting with the customer who can order a movie or even browse different titles while on the go. The system is able to recognize movie titles or some variations of these titles (e.g. Expendables Two vs Expendables 2) which are pre-programed in its database, however,

occasionally titles may not get recognized, especially if they are misspelled by the customer. Since each order is valuable to the enterprise, it is imperative to grow the database with common title misspellings so that the number of successful orders is maximized. Currently, there is a manual effort to do so.

This use case proposes a text mining / fuzzy matching approach to improve title recognition accuracy. One way is to use clustering using SAS Text Miner, however, the problem is formidable due to the fact that there are thousands of movie titles available, which means the classification problem has thousands of classes and requires considerable sampling and resource to get retrained. This could be a viable solution if the number of classes is decreased considerably.

The way to reduce the number of classes is to only consider those titles that are not recognized the most – but how can this be done before a classification model is built? Because if a title is not recognized – how would the machine know what title would have been the correct one? This problem is solved by analyzing the sequence of text messages between the system and the customer. It turns out that when a customer enters a movie title and a match is not returned from the database, the system sends the customer the following message "IM SRY, I DID NOT FIND *customer-entered-text*. PLS TRY AGN". The solution is then to count the number of messages that occurred right before this message and to group these messages by the message content itself.

Next, SAS implementation of the Levenshtein distance using function COMPLEV is used to measure the top N (where N is parametrically defined) errors against all known titles and associate those to the titles that have the smallest distance. Matches can also be manually validated. Next, SAS Text Miner and clustering is used to split all the failed attempts into N classes. The process is repeated until a parametrically defined confidence level is reached.

USE CASE IV: PREDICTIVE MODELS

Predictive Intent

"I've called you three times already because my internet is down! You're making me repeat everything I've said!!" Some of the readers may be familiar with this situation. Customers now expect the IVR system to know who they are and why they are calling. Predictive intent is a determination of why someone is calling before the customer indicates in within the main menu. Today, there are rudimentary business rules that say "this customer called twice in the last hour about technical reason – they are likely calling about the same problem again". Tomorrow, the goal is to use advanced statistical techniques like predictive models to actually predict the propensity for a customer to call about each specific reason and present them with dynamic menus that are driven by those predictions.

Everyone is familiar with statements such as "this call may be recorded for quality" which typically occurs right at the beginning of the interaction with the IVR. What everyone may not know is that they are being recorded even when they are on hold or not interacting with the system. A lot of folks actually speak during those times and can give clues to the reasons for their call. Coupling actual caller behaviors with the transcribed utterances of what's being said results in much more accurate predictive intent models that ultimately optimize IVR performance and improve customer experience.

Customer Churn

Customer churn is a business challenge that enterprises have been tackling for decades. Typical embodiment is to build a propensity to churn model and then treat customers differently (either by proactive outreach or during an inbound interaction) and entice them with offers or perhaps rightsizing their services if they are about to roll off a contract and their discount will stop. The algorithm wars have been fought and won (or lost) long ago and now everyone uses the same procedures. The competitive advantage really comes from the data that can be used in predictions. West IVR gathers critical data about customer behaviors that is unparalleled in the industry. West Advanced Analytics uses this data coupled with customer attributes and third party demographic data to build more accurate churn models. Moreover, being the front end of most customer interactions, West can act upon these models and actually prevent churn as well.

It was mentioned earlier that West IVR is equipped with a Natural Language engine that is able to understand what the customer is saying and place him or her in the right area of the flow. Most of the

time, there is an interesting paradigm that is observed where consumers have been trained to speak in just a few words instead of large phrases (possibly there is an inherent mistrust in the ability of the system to recognize the phrase so customers have adapted to say “BILLING” instead of “I’D LIKE TO PAY MY BILL PLEASE”), however, there are certain enterprises who have customers in particular demographic who actually literally talk to the system. Since all the customer utterances are captured and transcribed by the system, it makes it possible to use SAS Text Miner to categorize those utterances and use the resulting categories (moving from unstructured data to structured data) to ascertain a customer’s agitation and then enhance the churn models with these new variables that add significant lift to purely structured data driven models.

The deployment of such churn models would be accomplished in real time as presented in Figure 3 above. For more examples on this topic, see Dmitriy Khots talk on Voice Analytics at the SAS Analytics Conference 2013.

CONCLUSION AND NEXT STEPS

West IVR is a customer interaction portal. It is the face of some of the most well-known enterprises and their brands – it is something that first greets potential customers when they want to learn more or sign up or existing customers when they have a question. As such, it plays a critical role not only in saving West clients millions of dollars in avoiding agent calls via automation of routine tasks and generation of revenue through automation of ordering, but also in driving customer satisfaction, which is key for customer retention as well as revenue generation. Understanding what exactly happens inside the IVR (especially during launches and outages – in near real time), how consumers interact with it, what paths they traverse, how they complete automation tasks, what is their effort (time taken to complete transactions), and where they fall out (hang-up or transfer to agents) are all questions that can be answered with data mining methodologies, which in turn can be used to optimize the performance of IVR both for the users and for West clients (managing user experience vs cost savings). This paper presents a high level data mining framework for accomplishing these tasks along with a few use cases of what is under development today. The next steps are to measure impact of the changes advanced analytics team is recommending through the use of A/B and multivariate testing using design of experiment best practices and measure ROI on such initiatives as well.

ACKNOWLEDGMENTS

The author would like to thank Professor Goutam Chakraborty of Oklahoma State University for the invitation to discuss applications of unstructured data mining in IVR at the SAS Global Forum 2015.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Dmitriy Khots, Ph.D.
West Corp
dkhots@west.com