Documentation

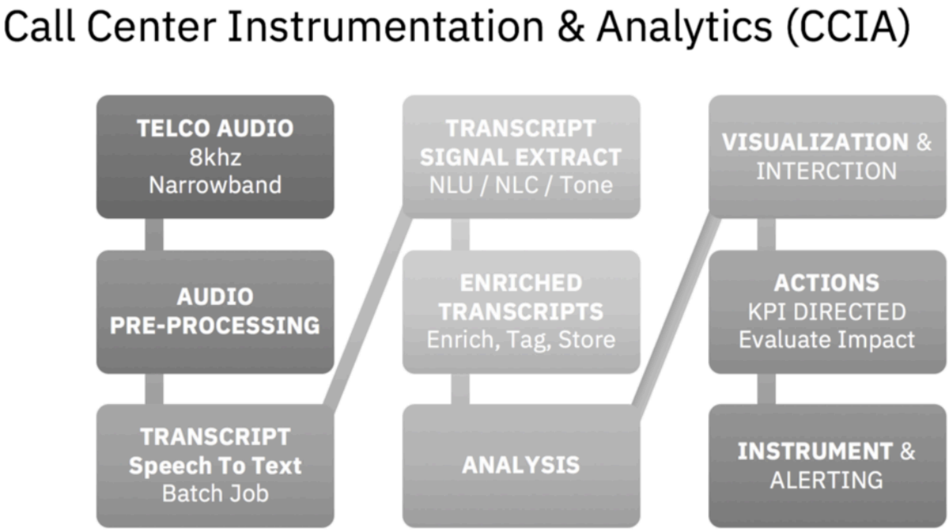
Call Center Instrumentation & Analytics

Watson-Call-Center-Think18 Lab – March 2018

Overview

This document provides guidance and background for a hands-on Python + IBM Watson lab being presented at an IBM Think2018 conference in March 2018, and for an IPython / Jupyter notebook and python code available for open source after the event.

The focus is Call Center Instrumentation and Analytics (CCIA) pattern. The Notebook and information here seek to help organizations beginning to explore how to better understand the unstructured "dark data" that arises from phone calls to call centers.


Call Center Instrumentation & Analytics (CCIA)
TELCO AUDIO 8khz Narrowband
AUDIO PRE-PROCESSING
TRANSCRIPT Speech To Text Batch Job
TRANSCRIPT SIGNAL EXTRACT NLU / NLC / Tone
ENRICHED TRANSCRIPTS Enrich, Tag, Store
ANALYSIS
VISUALIZATION & INTERCTION
ACTIONS KPI DIRECTED Evaluate Impact
INSTRUMENT & ALERTING

In short, a beginner's guide for starting to analyze call center data:

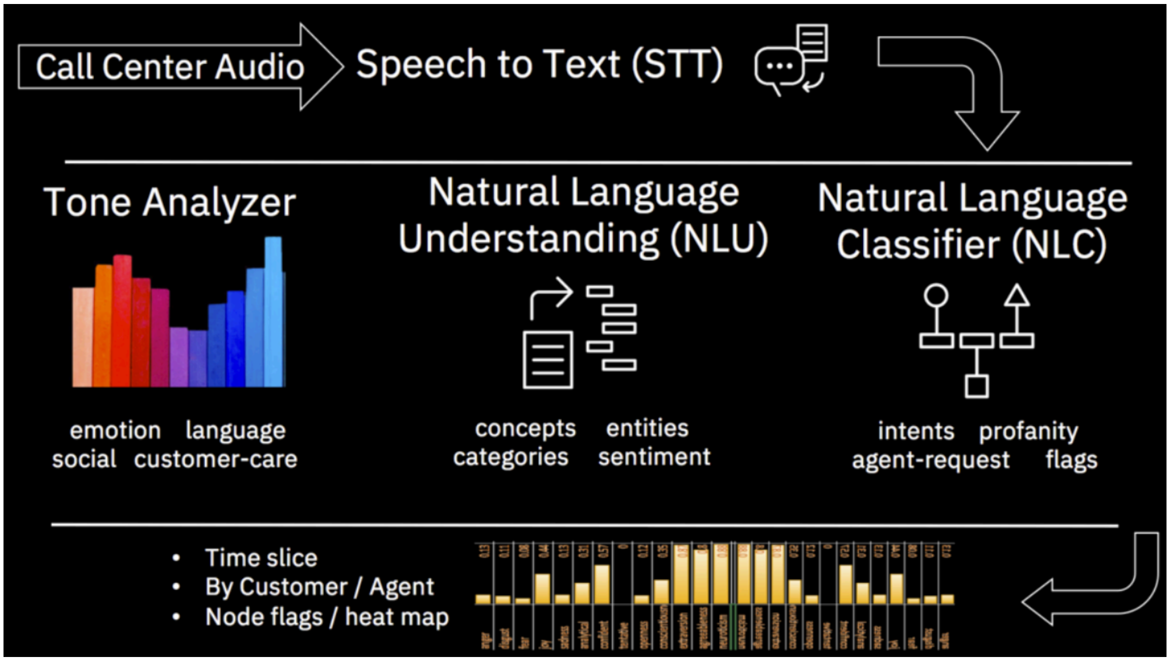
- DSX / Watson Studio Notebook Name: Watson-Call-Center-Think18 Lab
- *GH:* https://github.com/mamoonraja/call-center-think18/tree/master/notebooks

Why is this useful?

Enterprises spend more than $1 trillion on 250 billion customer service calls each year. By using multiple IBM Watson "signal services" to extract signal from raw audio data; perform data analytics, clustering, unsupervised and machine learning, and visualizations, technical teams can use data understand patterns in call centers. KPI and ROI positive.

What is the process? And what Watson services are used?

1. Speech to Text (STT) – Converts Raw Audio to Transcripts
2. Natural Language Understanding (NLU) - extracts features concepts, entities, keywords, categories/topics, sentiment and emotion
3. Natural Language Classifier (NLC) - is a user trained classification service, with user defined "ground truth" that classifies text chunks
4. Tone Analyzer (Tone) – uses linguistic analysis to detect emotional and language tones in written text
5. Call Center Analytics – analyzes and visualizes the data signal to allow for interpretation of data and in cases, actionable insights.



Beginner Audience & Focus on Basics

- This is a beginner lab intended to educate on the fundamentals of getting from data to insights with IBM Watson and open source tools
- Audience may include IT and operations teams curious about enriching unstructured data – the lab is NOT intended for sophisticated call center technologists
- Lab/code does NOT purport to compete with expensive and sophisticated solutions already in market
- The lab and code cover the basics – to educate on the fundamental plumbing and steps, to provide base for instrumentation

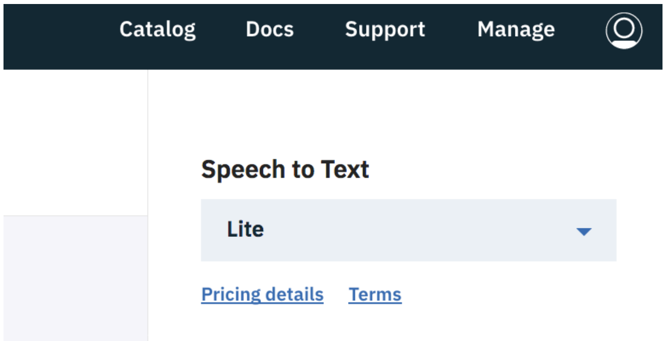
Success Metrics

If successful – the lab participants or notebook users will

1. Gain experience in using an IPython / Jupyter notebook https://ipython.org/notebook.html
2. Connect to four **Watson Developer Cloud** 'signal service' APIs https://www.ibm.com/watson/developer/
3. Connect to IBM **Cloud Object storage** for data read and write https://www.ibm.com/cloud/object-storage
4. Understand whether/how the tools and methods might benefit org https://github.com/mamoonraja/call-center-think18/tree/master/notebooks

Notebook #1 – Speech to Text (STT) & First Contact

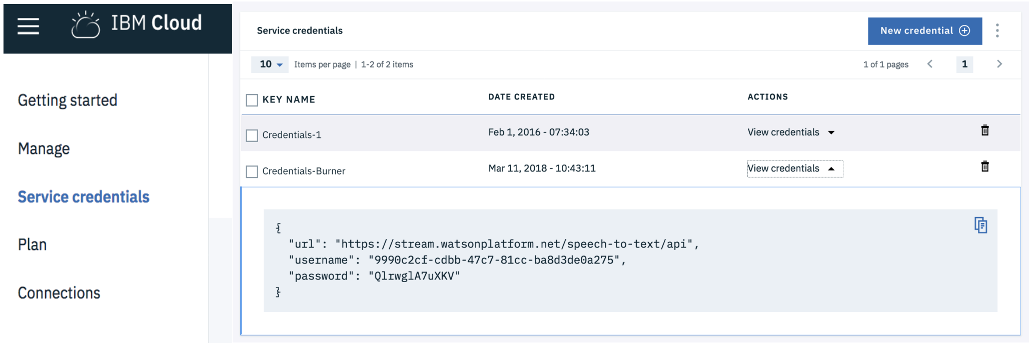
- 1.1 – Installing Dependencies
- 1.2 – Importing Credentials – Each Watson signal service (STT, NLC, NLU and Tone) will require credentials and configuration.
    - An API endpoint (URL)
    - Username & Password
    - If you already have an IBM Cloud / Bluemix account
        - login here https://console.bluemix.net/
    - **If you have not registered for IBM Cloud - you will need to Register for a Free account here https://www.ibm.com/watson/developer/**
- At the https://console.bluemix.net/developer/watson/dashboard dashboard -
        - Speech to Text
        - Natural Language Understanding (NLU)
        - Natural Language Classifier (NLC)
        - Tone Analyzer (Tone)
          More info on services here: https://www.ibm.com/watson/developer-resources/
    - LITE Plan for STT "gets you started with 100 minutes per month at no cost"



Speech to Text

Lite

- The Username and Password (and URL) are found by clicking on service credentials, and then “view credential”



Service credentials

KEY NAME	DATE CREATED	ACTIONS
□ Credentials-1	Feb 1, 2016 - 07:34:03	View credentials
□ Credentials-Burner	Mar 11, 2018 - 10:43:11	View credentials

```
{
  "url": "https://stream.watsonplatform.net/speech-to-text/api",
  "username": "9990c2cf-cdbb-47c7-81cc-ba8d3de0a275",
  "password": "QlrwglA7uXKV"
}
```

- 1.3 – For the lab – some short audio files (OGG format) will be available for transcription – this step establishes credentials for connecting to the Cloud Object Storage resource – which will be BOTH READ (Audio) and WRITE (Signal returned from Watson APIs)
- 1.4 – Two Methods are used for translating Speech To Text:
  - get_transcript() calls speech to text enpoint and generates a text transcript for you for a sample audio file.
  - analyze_sample() gets the sample object from cloud storage, calls get_transcript to fetch the tranccript, and saves your transcript in cloud storage as <file_name>_text.json.
  - OGG, WAV FLAC, L16, MP3, MPEG formats are options for the STT service – with Narrowband (generaly telco) and Broadband (e.g. higher quality USB mic) audio. For the LAB – OGG format was used for sample files.
- 1.5 – TRANSCRIBE – this is where STT receives the OGG files provided and returns text
- 1.6 – FILE LIST – in this notebook, each OGG file produces its own transcript. These ones are quite short so it happens quickly. Longer (e.g. 1 hour) audio files may justify using asynchronous method, and a real time a sessions method (both defined below)

https://www.ibm.com/watson/developercloud/speech-to-text/api/v1/

  - **WebSockets** includes a single method that establishes a persistent connection with the service over the WebSocket protocol.
  - **Sessionless** includes a method that provides a simple means of transcribing audio without the overhead of establishing and maintaining a session.
  - **Sessions** provides methods that allow a client to maintain a long, multi-turn exchange, or session, with the service or to establish multiple parallel conversations with a particular instance of the service.
  - **Asynchronous** provides a non-blocking interface for transcribing audio. You can register a callback URL to be notified of job status and, optionally, results, or you can poll the service to learn job status and retrieve results manually.
- 1.7 Analyze – the returned text is chunked up in this case – so later we will concatenate each piece, and then chunk it up into segments defined (for example X number of words; or Y seconds in transcripts) which will permit measuring of each bucket in time domain.

Notebook #2 – Natural Language Understanding (NLU)

- https://www.ibm.com/watson/services/natural-language-understanding/
- https://www.ibm.com/watson/developercloud/natural-language-understanding/api/v1/
- 2.1 –Importing Credentials – Each Watson signal service (STT, NLC, NLU and Tone) will require credentials and configuration.
  - An API endpoint (URL)
  - Username & Password
  - If you already have an IBM Cloud / Bluemix account login here https://console.bluemix.net/
  - **If you have not registered for IBM Cloud - you will need to Register for a Free account here https://www.ibm.com/watson/developer/**
- https://console.bluemix.net/developer/watson/dashboard
  LITE Plan for NLU is free

Notebook #3 – Natural Language Classifier (NLC)

- https://www.ibm.com/watson/services/natural-language-classifier/
- https://www.ibm.com/watson/developercloud/natural-language-classifier/api/v1
- 3.1 –Importing Credentials – Each Watson signal service (STT, NLC, NLU and Tone) will require credentials and configuration.
    - An API endpoint (URL)
    - Username & Password
    - If you already have an IBM Cloud / Bluemix account login here https://console.bluemix.net/
    - **If you have not registered for IBM Cloud - you will need to Register for a Free account here https://www.ibm.com/watson/developer/**
- https://console.bluemix.net/developer/watson/dashboard
  LITE Plan for NLC is free

- **3.2 - NATURAL LANGUAGE CLASSIFIER must be trained** – unlike the other Watson APIs in this lab – which work out of the box – the NLC needs an Ground Truth (a CSV file with examples of utterance and classes) to 'show Watson' the desired behavior
    - NLC Training from GT's can take time – depending on time and complexity – e.g. a 300 row, 5 class example may take 12 minutes be available. More complex data takes longer to train –sometimes taking more than an hour to complete training.
    - NLC for this lab will use a PRE TRAINED – Ground Truth examples will be provided
- 3.3 At the moment, the lab is using a pre-trained NLC that classifies based on the Harry Potter books – the four houses of Hogwarts – so that Evil or Mean language should go to Slytherin; smart talk to Ravenclaw; bold brave noble to Gryffindor and middling to Hufflepuff (this may change)

Notebook #4 – Tone Analyzer

- https://www.ibm.com/watson/developercloud/tone-analyzer.html
- https://www.ibm.com/watson/developercloud/tone-analyzer/api/v3/
- Tone Analyzer service uses linguistic analysis to detect emotional and language tones in written text.

- 4.1 –Importing Credentials – Each Watson signal service (STT, NLC, NLU and Tone) will require credentials and configuration.
    - An API endpoint (URL)
    - Username & Password
    - If you already have an IBM Cloud / Bluemix account login here https://console.bluemix.net/
    - **If you have not registered for IBM Cloud - you will need to Register for a Free account here https://www.ibm.com/watson/developer/**
- https://console.bluemix.net/developer/watson/dashboard
  LITE Plan for Tone is free
- 4.2

Notebook #5 – Call Center Analytics & Visualizations

This section is about the "What next?" – For the lab it is basic visualization of data for interpretation, but other real-world signal paths may include:

- Feed information into an existing business intelligence system
- Develop a new dashboard or visualization
- Instrument for alerts that map to KPIs or business rules
- Develop experiments to test ideas or patterns – e.g. https://hbr.org/2017/01/kick-ass-customer-service
- Feed information to data science, analytics or operations team for them to leverage
    - 5.1 - Install and load libraries for the visualization and analytics. For example, WordCloud and Matplotlib, which is a Python 2D plotting library https://matplotlib.org/
    - 5.2 – Load NLU Signals from Cloud object storage
        - prior steps of the process pulled NLU signal – structured data - that includes entities, categories and sentiment – that will be used for analysis and visualization.
        - This step re-imports it. If desired, other complementary data can be imported here.
        - API keys, endpoints and service names need to be provided here – the keys to the vault
    - 5.3 - Visualize Sentiment and Top Keywords using Watson NLU
        - PARSE - Not all NLU data from COS is needed – just

    - READ FILES - For this lab - each Transcript has its own JSON file – but if doing this at scale, with thousands or millions of transcripts – another method may be more sensible
    - DATA FRAME – this puts data into a structured and rectangular data frame
    - PIXIEDUST – PixieDust is an open source helper library that provides tables, charts and maps. It works as an add-on to Jupyter notebooks to improve user experience of working with data. It also fills a gap for users who have no access to configuration files when a notebook is hosted on the cloud. https://github.com/ibm-watson-data-lab/pixiedust
    - GROUPING – simple summary table of the things we care about – in this case a count of the number of calls classified at top level as positive, negative or neutral
    - PIE CHART – data from summary is visualized with labeled pie chart
    - KEYWORDS VISUALIZED HISTOGRAM – in the lab the sample calls are quite short so this is not very interesting – but more rich data can be examined by importing data from data/data frame into callcenterlogs_nluDF

    - KEYWORDS VISUALIZED WORDCLOUD – Focusing on the top keywords by sentiment score and relevance – a histogram is prod; to scale value is to adjust the importance of a frequency word https://github.com/amueller/word_cloud/blob/master/wordcloud/wordcloud.py
- 5.4 - Visualize Emotion Tone using Watson Tone Analyzer response
    - Here, we parse Tone Analyzer response file from Cloud Object Storage, as above – and use emotion tone labels and scores.
    - Separate JSON files are imported and translated into a Data Frame, as above
    - PIE CHART – data from summary is visualized with labeled pie chart

- 5.5 - SUMMARY
- Given the time constraints, the Matplotlib and Pixie Dust visuals above are VERY basic. If initial data is promising, users may want to assess value of signal from Call Center
- Instrumentation and Analytics may include
  - Clustering of types of CALLS together aggregation for a Holistic view of data // real time operations center data
  - Segmentation of types of CALLERS & AGENTS (e.g. HBR)

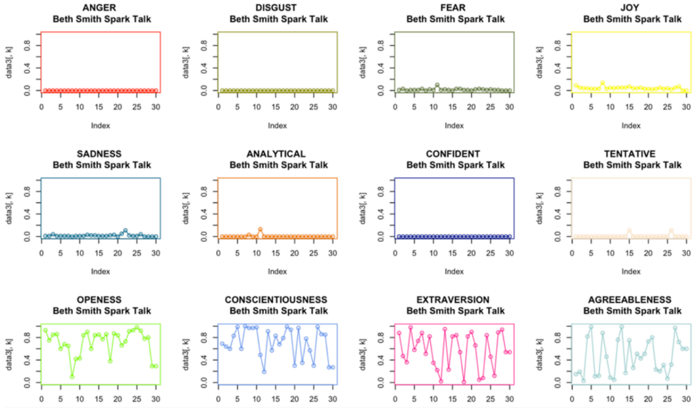
THE CONTROLLER
Outspoken and opinionated; likes demonstrating expertise and directing the customer interaction
Rank: #1
Proportion: 15%

THE ROCK
Unflappable and optimistic; doesn't take diffcult conversations personally
Rank: #2
Proportion: 12%

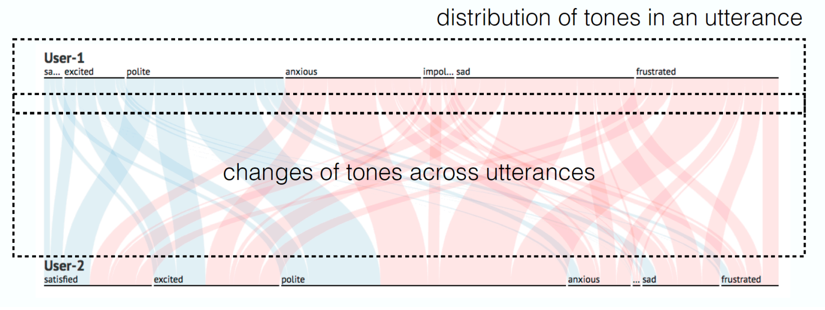
THE ACCOMMODATOR
Meets people halfway; involves others in decision making; eagerly offers discounts and refunds
Rank: #3
Proportion: 11%

THE EMPATHIZER
Enjoys solving others' problems; seeks to understand behaviors and motives; listens sympathetically
Rank:#4
Proportion: 32%

  - Topics, styles or methods strongly correlated with good or bad outcomes as measured by KPIs



  - For call quality monitoring, a visual analytic approach to understanding call conversations based on dynamic tones has been used successfully elsewhere


distribution of tones in an utterance
User-1
sa...
excited
polite
anxious
impol...
sad
frustrated
changes of tones across utterances
User-2
satisfied
excited
polite
anxious
...
sad
frustrated