TRANSCRIPTION NORMALIZATION

Transcription Normalization is a fundamental part of any NLP or NLU pipeline. It is predominantly used to process voice data, or speech-to-text data to make Information Retrieval and any analytics/processing simple — a key part of various NLP application pipelines, such as chatbots, sentiment classification, named entity recognition, and text summarization. In some models, normalized text represents any text with accurate casing information and sentence boundaries, while more sophisticated models can perform a diplomatic transcription, which preserves all orthographic oddities of the original documents.

PROJECT GOALS

- Development of transcription normalization engine, using sequence labelling and bidirectional LSTM+CRF models
- Integration of normalization engine with the The organization's NLU microservice
- Development of engine capability to recognize names, addresses, dates, times, phone numbers and member ID's
- Develop true casing and sentence boundary detection models, specifically for call center voicemail transcription
- Develop framework to analyze any voice data using sequence labelling models
- Develop expertise in Keras, and sequence labelling and bidirectional LSTM+CRF models as a part of the ATC AI/ML team

INTRODUCTION

This project develops a Transcription Normalization engine, as well as a framework to analyze voice data using sequence labelling and bidirectional LSTM+CRF models.

- Interact with speech processing and NLP APIs.
- Implement an engine that parses input, adds sequence labels to training data.
- Train the model using **tsv** files so it 'learns' from tagged speech data.
- Wrap code into a transcription normalization API and prepare it for integration into the organization's NLU.

APPLICATION SCENARIOS

- Automated normalized transcription for calls and voicemails. Proper formatting of names, DOB's, ID's, phone numbers would prepare the text for more efficient and faster Information Extraction.
- A new way to prepare raw and unstructured voice data for any analytics/further processing — including further NLP applications, such as linguistic analysis, classification, Information

extraction, document insights, structure detection, language modelling, natural language generation etc.

- Be used as a part of the organization’s’s NLP Microservice, and a preprocessing tool for their various services.

LEARNING GOALS

On completion, the transcription normalization microservice should be running from a web UI and be able to:

- Take any unstructured or raw voice data.
- Run it through the model to regain sentence boundaries and casing information.
- Be "trainable" on other domains with minimal changes to the code, other than training data and labels.
- Run efficiently and accurately as a part of the organization’s’s NLP Microservice.
- Provide teams across the enterprise with opportunity to leverage the applications of NLP

ANAGO

This project implements anaGo, a Python sequence labelling library sequence labeling (NER, PoS Tagging,..), implemented in Keras.This library uses bidirectional LSTM + CRF model based on Neural Architectures for Named Entity Recognition. In anaGo, the simplest type of model is the Sequence model. Sequence model includes essential methods
like fit, score, analyze and save/load. For more complex features, you should use the anaGo modules such as models, preprocessing and so on.

anaGo supports following features:

- Model Training
- Model Evaluation
- Tagging Text
- Custom Model Support
- Downloading pre-trained model
- GPU Support
- Character feature
- CRF Support
- Custom Callback Support

anaGo officially supports Python 3.4–3.6.

SPEECH API

This is an internal API providing speech-to-text capabilities, based on Kaldi, a well-known ASR toolkits. This API will serve as our ASR backend.

NLP MICROSERVICE API

This is an internal API providing capabilities to train models for various NLP uses, such as benefit inquiry, text summarization, consumer complaint classification etc.

GENERAL READINGS

- https://cloud.google.com/speech-to-text/
- http://blog.voicebase.com/artificial-intelligence-machine-learning-deep-learning-love-triangle
- https://medium.com/@ageitgey/machine-learning-is-fun-part-6-how-to-do-speech-recognition-with-deep-learning-28293c162f7a
- https://github.com/pannous/tensorflow-speech-recognition
- https://sites.utexas.edu/firstbooks/2016/03/25/new-tools-for-modernized-transcription/
- https://web.stanford.edu/~jurafsky/slp3/2.pdf
- https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html
- https://arxiv.org/abs/1603.01360
- http://www.cse.unsw.edu.au/~billw/nlpdict.html
- https://www.cs.cmu.edu/~llita/papers/lita.truecasing-acl2003.pdf
-

MILESTONES

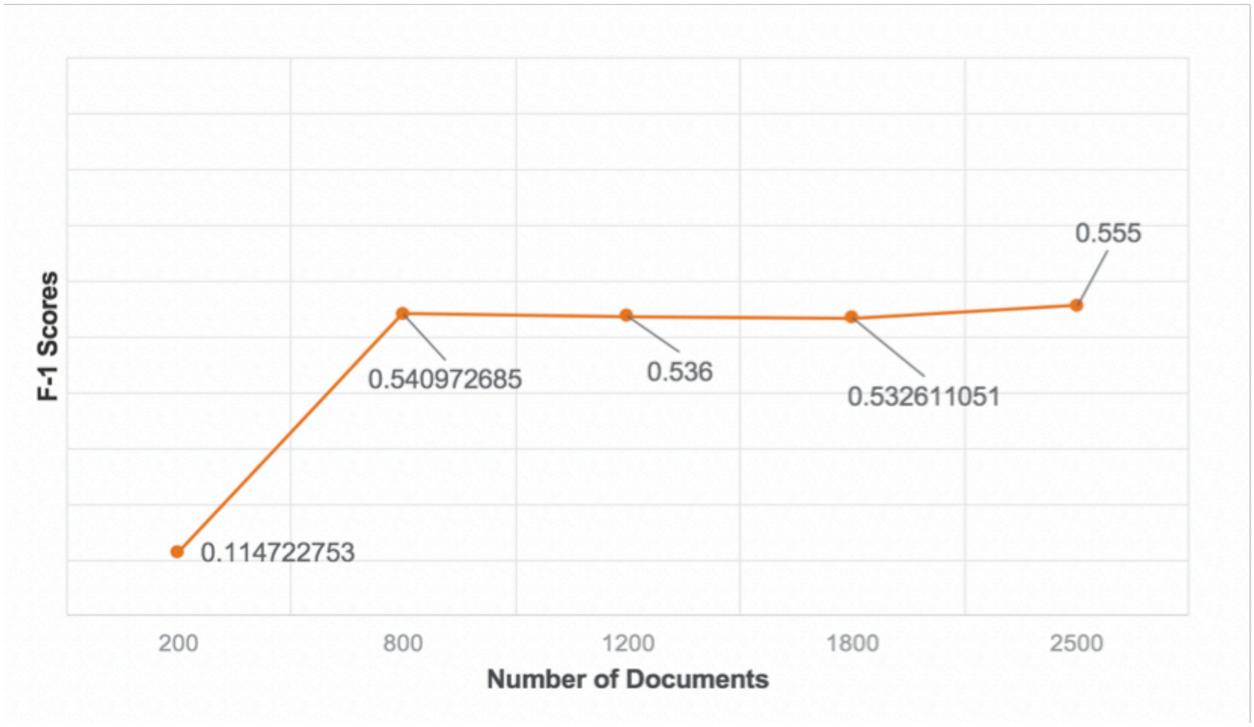
As the project progresses your team will autonomously split the work and break down tasks with the help of our scrum masters. The table below outlines the high-level milestones of the project. If you haven't worked with Python/Linux/anaGo, many of the milestones implicitly have *learning goals* on the related technologies.

Milestones	**Description**
Training	Learn about various new and relevant NLP/ML/Deep Learning technologies on Data Camp and The organization's Tech University.
Environment Setup & Data Access	Get access to the **voicemails** from an internal The organization's server. Setup your **Python** development environment. Learn about **git** and our coding practices. Familiarize yourself with the APIs you will be consuming. Learn your way with anaGo and familiarize yourself with predefines functions
Data Access and Search	Begin search for appropriate and comprehensive open source NLP data sets. Extract data from large databases. Begin preprocessing of data — remove unnecessary strings (eg. Hyperlinks, redirects, subreddit title references) Prepare 3000 such documents.
Write tsv files	Write script for writing tsv from training data. Add sequence tags of 'B-SENT' at sentence boundaries.
Load Training/Testing Data	Experiment with training/testing ratios — finally ~2500 training and ~300 testing. Load data into model, train for 15 epochs and calculate F-1 scores for boundary detection model. Train on new data — https://github.com/Hironsan/anago Demo — https://anago.herokuapp.com
True Casing Model	Repeat Load Training/Testing data int the same way for true casing model. Tsv files tagged with 'B-UPPER'
Saving & Loading Models	Add saving/loading capability for each trained model.

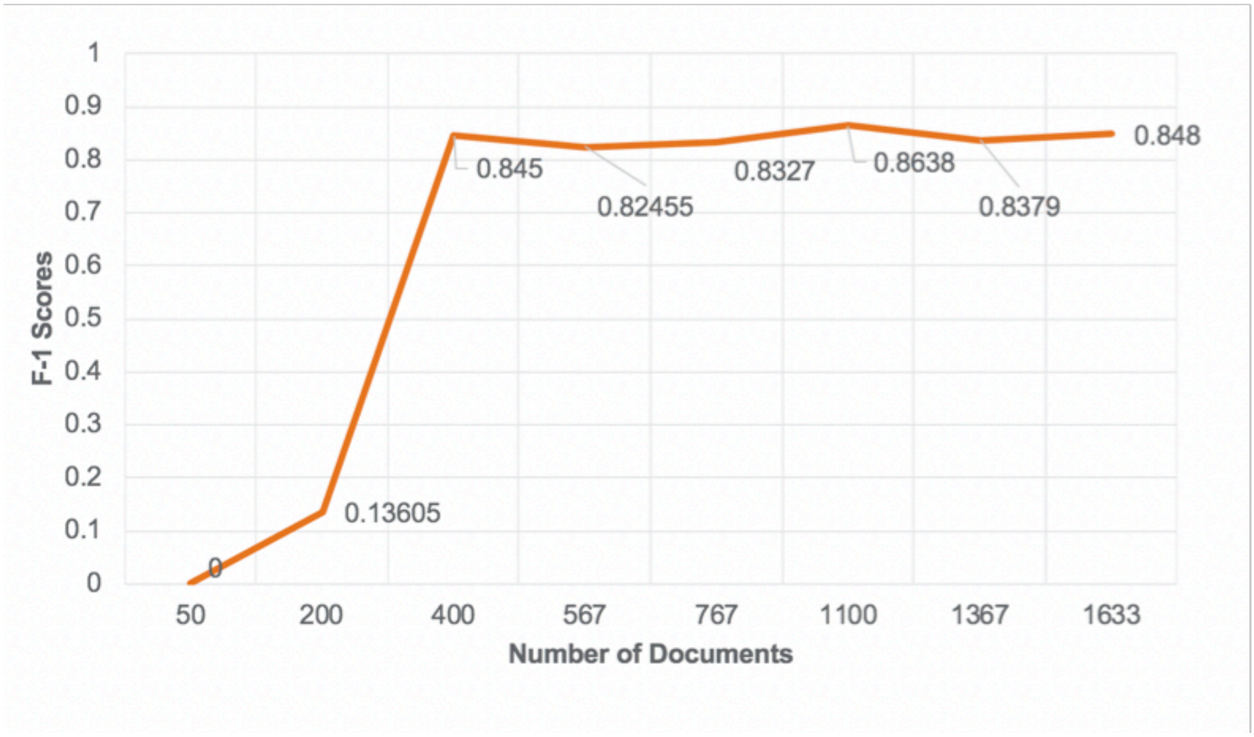
Evaluation	Run model on validation data sets of real voicemails.
Post-Processing	Write script to format dates of birth, phone numbers, member ID’s
Speech Normalization on a new domain	Use the same technology to train a model on a different domain, more relevant to The organization’s Ideally only the data and sequence labels should change.
Microservice Integration	Wrap code up into a single Class, prepare API to be integrated into The organization’s’s NLP API.

F-1 SCORES

- **BOUNDARY DETECTION MODEL**



- **TRUE CASING MODEL**



ACTUAL MODEL RESULTS

Input:

we're also trying to make sure that you're getting the right drug so that's partly why we or our staff may ask you a series of questions to verify your identity otherwise how can we be sure that your prescription is for the twenty two year old ian jamieson who lives off broadway and not the sixty two year old mary jones who lives off clark street

Output:

We're also trying to make sure that you're getting the right drug so that's partly why we or our staff may ask you a series of questions to verify your identity. Otherwise how can we be sure that your prescription is for the 22 year old Ian Jamieson who lives off Broadway and not the 62 year old Mary Jones who lives off Clark Street.

SUPPORTING TEAM

- **Kondadadi, Ravi**
- Nourkadi, Abdirahman
- Yerex, Robert P
- Garza, Nathan R

INTERNS (Matt Versaggi's Team)

- Ria Vinod
- Ian Jamieson

RESOURCES

- [SNLP] Bot Design
- [SNLP] Development Environment
- [SNLP] Documentation Artifacts
- [SNLP] Rasa
- Domain pieces for BnE bot
- Instructions of B&E bot
- Learning Resources
- Log: [SNLP] Speech Chatbot

PRESENTATIONS

FINAL INTERN EXPO HANDOUT — Intern Expo Handout

FINAL PRESENTATION — Final Presentation