Xerox Conversational AI Agent (XCAI) for Enterprise Knowledgebase Q&A

Niranjan Viladkar, Vivek Tyagi, Arunasish Sen, Sriranjani R, Pragathi Praveena
firstname.lastname@xerox.com
Xerox Research Center India (XRCI)

Large Vocabulary Speech Recognition (LVSR) using Kaldi

- Large Vocabulary Speech Recognition (LVSR) for different domains come with different set of vocabulary.
- There is also significant variations in speaker accent, dialect, pitch etc.
- A robust, highly accurate and real time speech recognition system – LVSR – is of prime importance for building a successful conversational AI agent.
- LVSR can extract information from spoken customer interactions and thus can help organizations to take more informed decisions about increasing their efficiency.
- In this poster, we present a real time speech recognition enabled conversational AI Agent.
- We use a GMM-HMM based acoustic model trained on TED-LIUM English speech corpus using Kaldi speech recognition toolkit.
- For Acoustic modelling (AM), we use Linear Discriminant Analysis (LDA) transforms and Maximum Likelihood Linear Transform (MLLT) estimation.
- For Language modelling (LM), we used CMU dictionary having around 133K word-pronunciation pairs as vocabulary and trained 3-gram LM using Witten-Bell smoothing over text corpus in the domain of Agent.
- We use GStreamer framework to create a pipeline of various audio processing plug-ins.
- We use various plug-ins like appsrc, decodebin, audioconvert, Kaldi's online GMM decoding plugin, etc.
- The audio is transferred to Xerox ASR servers' pipeline over TCP.

Demo URL :

XCAI Research Prototype

- Exponential growth in network bandwidths and computational power in recent years have boosted the development of conversational AI agents.
- Most of the existing AI agents are tightly coupled with world wide web or pre-determined command and control for Spoken language understanding and processing the semantics.
- A Conversational system as shown in Figure 1 has 3 major components, viz. Large Vocabulary Speech Recognition (LVSR), Natural Language Understanding (NLU) and Text to Speech (TTS).
- The LVSR architecture is shown in figure 2. This architecture fits in the GStreamer pipeline as a GStreamer plugin. And processes incoming audio for feature extraction and decoding.
- The decoded text is sent to the NLU unit which is OpenEphyra – this unit generates a response which is spoken back to the user by TTS module.

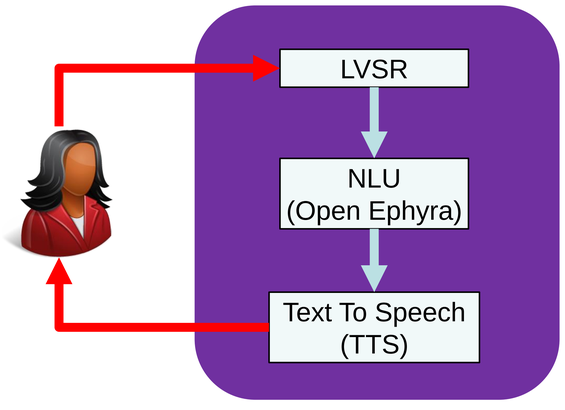

Fig 1. Xi-CVA Overall Architecture

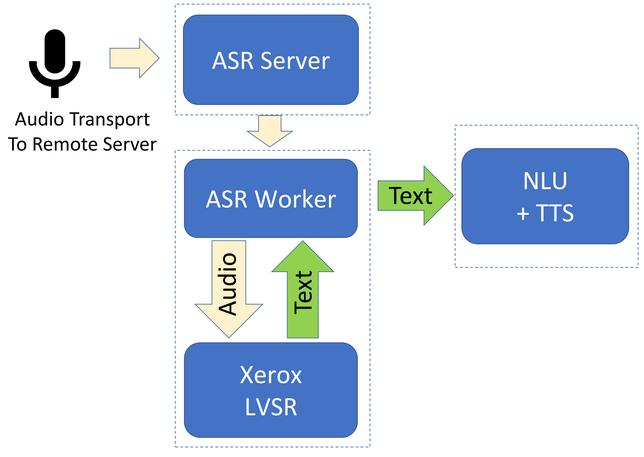

Fig 2. Xi-CVA LVSR Architecture

Test-To-Speech (TTS) using MaryTTS

- Mary-TTS is a multilingual, open source Text-To-Speech Synthesis system written in JAVA. It was originally developed as a collaborative project of DFKI's Language Technology Lab and the Institute of Phonetics at Saarland University
- It has a modular architecture, where each stage of the synthesis pipeline is written in form of a separate module. It supports Unit-Selection as well as Parameter Generation approach to T-T-S, which are currently, two predominant ways of synthesizing text. It has the ability to run as server and process API like calls to it. Also it has experimental support for synthesizing text with emotion and prosody overlay.

Natural Language Understanding using OpenEphyra

- OpenEphyra is a framework for answering textual natural language questions. For ex. 'Wh' queries like 'What is the height of mount Everest?'
- The framework consists of a question normalization module which has multiple natural language algorithms for word stemming, part-of-speech tagging, regular expression matching, etc.
- The question interpretation module identifies three components in the normalized question, viz. Property, Target and Context. Such that a question asks for a 'property' of a 'target' in a specific 'context'. It uses a pattern matching approach to determine these three components from pre-determined question patterns.
- The Query Generator forms one or more web queries which are then fed to the WWW via searcher module.
- Once the results are available, OpenEphyra applies various pattern matching based answer filters for answer extraction and answer selection.
- OpenEphyra provides a codebase that can run standalone and provide answers to the questions. We made OpenEphyra as a service so it can be kept running and can be made accessible for incoming questions using websockets.

References:

- Daniel Povey, et.al., "The kaldi speech recognition toolkit", IEEE 2011 Workshop on Automatic Speech Recognition and Understanding.
- Gstreamer: Open source multimedia framework, http://gstreamer.freedesktop.org/.
- Nico Schlaefer, et.al., "A pattern learning approach to question answering within the ephyra framework", Proceedings of the 9th International Conference on Text, Speech and Dialogue, TSD'06.
- MaryTTS: Text-to-Speech System, http://mary.dfki.de/.
- Anthony Rousseau, et.al., "TED-LIUM: an Automatic Speech Recognition Dedicated Corpus", LREC'12
- Tanel Alumae, "full-duplex speech recognition server", https://github.com/alumae/kaldi-gstreamer-server