Visual Voice Mail to Text on the iPhone/iPad

Andrej Ljolje, Vincent Goffin, Diamantino Caseiro, Taniya Mishra and Mazin Gilbert

AT&T Labs-Research
Florham Park, NJ 07932, USA

Abstract

Our visual Voice-Mail-to-Text (VMTT) transcription system takes a conventional voice mail and converts it to formatted text following standard punctuation, capitalization and presentation conventions. The text can then be used in a plethora of applications, from emails, to databases, text messages etc., which in turn allow searching, classification, data extraction, statistical analyses and other processes. Here we demonstrate our fully automated VMTT application by displaying the best scoring hypotheses from various recognition passes, the addition of punctuation and capitalization, formatting by using appropriate conventions for times, dates, dollar amounts and abbreviations, and finally applying grayscaling to lower the impact of the words recognized with low confidence scores.

1. Introduction

Although the VMTT task is underpinned by the very large vocabulary speaker independent automatic speech recognition technology it is much more than just ASR. We call it a Comprehension Task. The question one has to ask is: What is the purpose of transcribing a voice mail? Unlike the dictation systems, the quality of recognition is not measured by how many key-strokes it takes to correct the recognition misrecognitions. Unlike most dialog systems, it is not the understanding accuracy that determines if a dialog turn was successfully recognized. What, in the end, is shown to determine the success of the VMTT task is the satisfaction of the reader that the conveyed message was “comprehended”. This comprehension is very subjective and thus it is very difficult to estimate using conventional methods for determining the quality of the ASR output. In addition, unlike most speech recognition tasks, in many, if not most, cases there is a lot of information that aids in comprehending the message that is external to the message itself. Apart from the obvious, like the Caller ID information, most messages come from people we know, or relate to past events or places, which we, subconsciously or consciously, use to interpret the content of partially erroneously transcribed messages[1].

2. Training Data

Three sources of data dominated the training of all of the VMTT system components. The first consists of 100 hours of transcribed (no punctuations) voice mails from the Scanmail project [2]. The second consists of about 1,000 hours of internal voice mails, 300 of which have been transcribed, with punctuation and capitalization. The transcription quality is poor due to the imposed transcription speed requirement and high percentage of messages containing technical jargon. The final training database consists of several thousands of hours of voicemails. This speech was transcribed automatically with some human post-processing. The punctuation component was automated and corrected as appropriate at the same time text output was corrected. All the data sets were split into training, development and test sets. The development and test sets have also been accurately transcribed in house. We use high quality transcriptions for evaluating the quality of our VMTT system as well as the quality of transcription provided by external sources.

The data is always cleaned up automatically providing a uniform character set, spelling correction, enforcing transcription guidelines, and removing foreign, empty or very
noisy messages. In addition, the transcription of the bulk of our data is fully formatted text. This is good for training the punctuation and capitalization components but highly inappropriate for acoustic or language modeling. We use the mechanism shown in Figure 2 to convert formatted text to sequences of spoken words, removing unreliable components or compensating for untranscribed segments by introducing the word “_garbled” with pronunciation “grbl” (a single quasi-phoneme word).

Figure 3: Data unformatting mechanism.

3. Language Modeling

The ASR component uses a mixture language model trained on several text databases, with punctuation, capitalization and text formatting removed. The individual language models are interpolated with mixture weights tuned on development data from the target application, as shown in Figure 3.

Figure 4: Schematic representation of the interpolated language model used in the VMTT system.

4. Acoustic Models

Three acoustic models are used: the baseline model, Vocal Tract Length Normalized (VTLN) model and a model trained on adapted VTLN features using constrained model adaptation (CMA). They are all trained on all the transcribed voicemails, except that the VTLN and CMA trained models were limited to messages with more than 10 seconds of speech. They share the context dependency and the HMM structure to simplify different recognition passes and minimize the storage needed for search networks. The use of the acoustic models is shown in Figure 5, incorporating different multi-pass strategies. The system structure allows for three independent full decoding passes, or rescorings of lattices generated in the previous pass.

The recognition performance on the VMTT test set is shown in Figure 6, where the word accuracy vs. CPU total time is shown for each of the three passes for three decoding scenarios: full decode in each pass; lattice rescoring in the second pass; or lattice rescoring in the third pass. Search beams are adjusted for every pass, depending on the recognition scenario, greatly affecting speed and accuracy for that pass, with the first pass usually using a very narrow beam. Starting performance was with the baseline model trained on the manually transcribed data only. With streaming, latency effectively starts at the end of the first pass, adding to the other two dimensions in selecting the best configuration: speed and accuracy.

Figure 6: Recognition performance on the VMTT test set.

5. Demo

The demo is an HTML and Ajax app running on AT&T’s Talk browser, itself based on Apple’s WebKit rendering engine. Talk adds the ability to stream audio to and from an HTTP address using chunked HTTP requests. On startup the browser connects to an AT&T URL and downloads the app’s HTML pages. By pressing a button, input speech is streamed to the address of a Watson HTTP server. The audio is sent encoded in 8kHz AMR. Decoding occurs on the Watson server and a JSON object is returned with the recognition. The handheld browser then parses this reply and updates its display, Ajax style.

6. Summary

This paper describes our demo of the Visual VMTT system developed for fully automated voice mail transcriptions. It shows how various components interact to provide highly comprehensible written representations of the voice mails and how different passes of the ASR component provide progressively more accurate output with small additional computational load or latency.

References