Designing a Multi-lingual Corpus Collection System

Jonathan Law
Pace University

1. Abstract

One of the main requirements for the construction of a successful Automatic Speech Recognition (ASR) System is assembling sizeable speech corpora for language training and testing. While pre-existing corpora are available for many languages, adding a new language can require significant effort and cost. This paper describes a framework for corpus collection that integrates telephony systems and the Web using VoiceXML and HTML. The system is simple yet flexible and able to record a large, multilingual corpus of telephone speech for research in speech recognition related studies. We address the key issues in assembling corpora in different languages, including how to interact with callers to obtain the desired responses for both controlled and uncontrolled content, and the process of adding new languages with minimum effort. Because signal noise, equipment, and other factors can impact the quality of the data, we use a verification module to screen out unacceptable data. Thus, this system contains two modules: one for speech data acquisition and one for data verification. We leverage appropriate emerging technologies to establish this framework and show that the system is user-friendly, easy to use, and flexible in creating corpora for new languages.

2. Introduction

A speech corpus database plays an extremely important role in the field of multi-lingual language identification research. A public domain database of utterances was created by DARPA project for various algorithms for speech recognition in 1986[1]. Since 1993, Oregon Graduate Institute Speech Corpus [2] has created what becomes the de-facto standard data set, and over the years it has updated to 22 languages. Still, adding a new language is no simple task. This paper describes how we attempt to leverage the current advancement of voice over IP and Internet technologies to accomplish this task with much lower cost in setup, equipment, and labor. In addition, the design is flexible enough that it should handle virtually any language corpus without much modification.

This multilingual speech collection system basically contains two modules, the speech collection and speech verification system. For collection system, callers are encouraged to dial in via a toll-free number supported by TellMe voice application network[3]. In order to set up the attributes of these data, all callers are tagged with a system generated unique internal identifier, greeted with a welcome message and prompted for questions, like their gender, phone types to differentiate them in our naming convention. Then callers are presented a series of prompts containing pre-defined texts for callers to read (phonetically rich sentences, dates, times, spoken and spelled names), a set prompts to elicit responses (“what time is it now?”), and a set of questions aimed at obtaining spontaneous speech. The purpose of this is to collect both content-dependent and content-independent responses, as well as to collect various lengths of the utterances.

This paper will discuss the technical and design issues on the integration of telephony and internet technology to make this a dynamic system that are feasible to collect virtually any
telephone speech corpus. It will be built on the multilingual template-based application framework using voiceXML[4].

3. System Design and Acquisition

We plan to recruit the first batch of users from the university’s international student body, which is represented from over 94 foreign countries with over 100 dialects are spoken. There will be two groups of users, corpus speakers and corpus verifiers. All verifiers must be proficient in languages of their choice and will be given a login ID and password for accessing to the corpus verification web page. Corpus speakers will be accessing the phone network via a toll free number. Callers will be prompted with a series of questions and each response will be recorded into a RIFF format file and stored in the database. These speech data files will be verified by native speakers and assigned with ratings. A separate process will periodically move the approved files from one directory into archives for research usage.

3.1 Application Framework Architecture

By mapping the data flow process and the Voice over IP network application, we can easily visualize how these two modules form the template-based framework architecture as depicted in the follow diagram. Both the corpus collection module and the corpus verification module are built on the same backend processing, except the presentation layers for voice and graphical user-interface.

![Figure 1 Multi-lingual Corpus Collection Application Architecture](image)

The template-based application framework uses distinct layer architecture, as depicted in Figure 2 below. Each layer is responsible for different functions. The Presentation layer defines the user interface of the different devices it supports. The Gateway Services layer handles the transport of information to and from the different devices. The Application Services layer provides the web server and application server. The Business Logic layer defines the dynamic content generation and the Data Access layer is responsible for data maintenance and integrity.

9.2
4. Corpus Collection Module Overview

As depicted in Figure 1, this framework contains both the Voice and Web Servers. We chose to build this framework on the Microsoft NT platform. The Voice browser is the TellMe platform using VoiceXML 2.0 specifications. The web application and the web server is all open-source software from the Apache Software Foundation. mySQL has been selected as the database; and PHP has been selected as the server side scripting language which is combined with an Apache server. From PHP Resources center, we also selected two object classes for easy code generation and maintenance. FastTemplate object class has been selected for parsing templates for variables and returning HTML with the appropriate values. Likewise, AbstractDB object class has been selected to create an abstract interface for different database access.

4.1 Framework Concepts and Components

The concept of this template-based framework is to separate the user interface from the actual content. In this scenario, the user interface is the phone via the dialog template engine and the content is dynamically delivered based on user selection and business logic. It also selects FastTemplate object class from PHP extension for content transformation. The file hierarchy and content of these folders are as follows:

- Resources - resource bundles, configuration files
- Prompts - pre-recorded wave files, text for prompts
- Grammar - grammars files
- Templates - dialog designs
- Includes – XML DTD, DB abstract, generic functions or method

The design of this framework also maps into the classic software paradigm of Model, View, and Controller, as shown in Figure 3 with respect to the templates, business logic and the VoiceXML output. The Model is the template engine, which manages the behavior of the View. The View is the output of the VoiceXML files, and the Controller interprets user input, commands the model, and changes the view accordingly.
The main component is the dialog and text templates inventory, where a subject matter expert can concentrate on design and can change it without modification to the code or concern of system failure due to recompilation. These templates are simply text files with variables. When a template is parsed, the variables are substituted into text or pre-recorded wave files according to user selection.

4.2 Corpus Prompts

There are two types of prompts, the language setup prompts and the language collection prompts. The setup prompts are used for new language creation. The collection prompts are used during the corpus collection. The prompts and questions should be the same in all languages. Native language wave files are recorded via the language prompt setup module when new language is created.

Below are some of the setup prompt examples in baseline version (English), and new prompts can be added into the database via the web.

**Setup Prompts**

<table>
<thead>
<tr>
<th>Welcome</th>
<th>Thank you for calling the Pace University Speech Lab.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>We are studying the languages of the world, and we need to collect different samples of speech. Please response to the following questions after the tone and answer in your native language in a clear voice. Thank you for your support.</td>
</tr>
<tr>
<td>Gender</td>
<td>Are you male or female, or press 1 for MALE and 2 for FEMALE</td>
</tr>
<tr>
<td>Phone</td>
<td>Please say what kind of phone are you using, or press 1 for Cellular, 2 for landline, 3 for others</td>
</tr>
</tbody>
</table>

**Collection Prompts**

<table>
<thead>
<tr>
<th>Date</th>
<th>What is today’s date?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td>What time is it?</td>
</tr>
<tr>
<td>DaysOfTheWeek</td>
<td>Please say all the days of the week?</td>
</tr>
<tr>
<td>Route</td>
<td>Please describe the route you take to work or to school</td>
</tr>
</tbody>
</table>
4.3 Recording Details

All of the data in this corpus will be collected over regular landlines or cellular phones. Data will be sampled at 8 kHz 8-bit and stored as ulaw files. All of the data will be stored in standard 16-bit linear RIFF format. The actual recording of the response from the caller is making use of the `<record>` element of the VoiceXML. A reference to the recorded audio will be stored in the field item variable, which can be played back (using the `expr` attribute on either `<value>` or `<audio>`) or submitted to a server. Using our template-based framework, all prompts will be looped through the reference name defined as `{short_collection_prompt}` and `{prompt_type}` in the following example:

```xml
<?xml version="1.0"?>
<vxml version="2.0">
<form>
  <record name="responses" beep="true" maxtime="50s"
          finalsilence="4000ms" dtmfterm="true" type="audio/wav">
    <prompt>
      <audio>
        {short_collection_prompt}
      </audio>
    </prompt>
    <noinput>
      I didn't hear anything, please try again.
    </noinput>
  </record>

  <field name="confirm" type="boolean">
    <prompt>
      <audio>
        You just said
      </audio>
      <break size = "small"/>
      <audio expr ="responses"/>
      <audio>
        Please review and approve it, say yes or cancel.
      </audio>
    </prompt>
    <grammar type="application/x-gsl" mode="voice">
      <![CDATA[
        [yes ok yeah] {<conf "yes">}
        [no nope] {<conf "no">}
        [cancel] {<conf "cancel">}
      ]]
      ]]
    </grammar>
  </field>
  <filled>
    <if cond="confirm">
      <submit next="save_response.php"
              method="post" namelist="response"/>
    </if>
    <clear/>
  </filled>
</form>
</vxml>
```

4.4 Types of Speech Data

In order to fulfill NIST language requirement for testing, we designed our prompt to expect responses in various speech lengths. For content dependent protocol, we expect short
utterances of 5 to 10 seconds. For content-independent protocol, we encourage caller to give us at least 30 seconds speech. Obviously, the prompt is an important factor to produce this outcome.

Some of the content dependent prompts that are less than 5 seconds are:
- What day is today?
- What time is it?
- Say a familiar telephone number?

Content dependent prompts that are about 10 seconds are:
- Describe the route you take to work or school.
- Tell us something that you like about the town you are currently living.
- Tell us about the climate in your hometown.

Content independent prompts that expect utterance over 30 seconds are:
- Please read the major story today for at least 30 seconds.
- Please give us a summary of your favorite book that you have finished recently.

### 4.5 Language Selection by Caller or Creation of New Language

Callers access the corpus data collection module via a toll-free number on TellMe platform. After the setup prompts on caller’s gender and type of phone he/she is using for identification of each corpus’s data attributes, caller will then be asked to select the language of choice. The languages option is a dynamic grammar logic generated from the selection of all entries from the database table “Corpus_languages”. If there is a no match of option, caller has the option to create a new language by spelling the name of the new language. Once the valid language is accepted, an entry will be inserted into the “Corpus_language” table. Then the same caller will be asked if he/she would like to record the audio files of the caller prompts that would be used for other callers of this new language.

### 4.6 Caller Prompt Generation

There will be maintenance page for the administrator to add/delete/insert all entries in this corpus collection database. All caller prompts entries are stored in “Corpus_prompts” table as described in the database section. These caller prompts are generated dynamically from this table and are being built into voiceXML syntax with audio clips intermingled with synthesized speech. Example of the “DIGIT” prompt for English Language is as follows.

```xml
<prompt>
<prompt_type = DIGIT;>
<short_collectiont_Prompt= 'please say digits from 0 to 10.
<audio src='http://www.pace.edu/corpus/prompts/EN_P_DIGIT.wav'/>
</prompt">
```

thus, for the same prompt for Spanish, the wave file name is substituted as

```xml
<short_collectiont_Prompt= 'please say digits from 0 to 10.
<audio src='http://www.pace.edu/corpus/prompts/SP_P_DIGIT.wav'/>
</prompt>
```

Notice the audio filename and prompt are mapped into the appropriate language and naming convention. If no audio file of the specific filename is found, default is English prompt.
4.7 Data Recordings

All calls for language corpus are stored in the same subdirectory, to simplify archiving and verification of the calls later by the verifiers. An entry with appropriate filename representing this speech data is inserted into the “Corpus_SpeechProfile” table, with initial rating of 0, and 5 for the most accurate result.

The following different types of recording problems are handled:
- The caller hangs up unexpectedly before completing all responses.
- No responses detected from the caller.
- An error occurs and caller is asked to repeat the last response.
- Caller barge-in and system will move on to the next prompts.

Thus there are times when all not prompts are recorded successfully, but the unique caller ID counter is till incremented.

4.8 Speech Data Files Naming Convention

The speech data filename is the concatenation of
- LangType – First 2 character of the language, or specified by the administrator
- Gender - gender of the caller
- PhoneType – Cellular or landline
- UserID - unique identifier of the caller
- Speech Type - short name of the speech content

Example : Speech data file in Cantonese language spoken by a female over her cellular phone on the days of the week corpus.

CA_F_101_DaysOfWeek.WAV

4.9 Corpus Data Collection Module Scenario

The following sections depicts the scenario when caller dial into the Collection system

<table>
<thead>
<tr>
<th>Actors</th>
<th>Dialog</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computer</td>
<td>Thank you for calling the Pace University Speech Lab.</td>
<td>Welcome prompt</td>
</tr>
<tr>
<td>Computer</td>
<td>We are studying the languages of the world, and we need to collect different samples of speech. Please respond to the following questions after the beeping tone and answer in your native language in a clear voice. Thank you for your support.</td>
<td>Introduction prompt</td>
</tr>
<tr>
<td>Computer</td>
<td>Are you male or female, please press 1 for MALE and 2 for FEMALE</td>
<td>Set file attribute</td>
</tr>
<tr>
<td>Caller</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computer</td>
<td>What kind of phone are you using, please press 1 for Cellular, 2 for landline, 3 for others</td>
<td>Set file attribute</td>
</tr>
<tr>
<td>Caller</td>
<td>Landline</td>
<td></td>
</tr>
<tr>
<td>Computer</td>
<td>Please say the language for our Corpus Collection ?</td>
<td>Language selection</td>
</tr>
<tr>
<td>Caller</td>
<td>English</td>
<td></td>
</tr>
<tr>
<td>Computer</td>
<td>The purpose of the following recording is</td>
<td>5 seconds prompt</td>
</tr>
<tr>
<td>Computer</td>
<td>What day is today?</td>
<td>What day is today? Data file will be recorded as DATE type.</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>Caller</td>
<td>February 10, 2002</td>
<td>Being recorded as DATE type.</td>
</tr>
<tr>
<td>Computer</td>
<td>Please say the days of the week ?</td>
<td>10 seconds prompt, and data file will be recorded as DaysOfWeektype.</td>
</tr>
<tr>
<td>Caller</td>
<td>Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday</td>
<td>Being recorded as DaysOfWeektype.</td>
</tr>
<tr>
<td>Computer</td>
<td>We now would like to record a longer snapshot of your natural speaking. We need to tell us your thoughts on a breaking news story or tell us the plot of your favorite movie, or a book you recently have finished.</td>
<td>Long utterance prompt, and data file will be recorded as STORY type.</td>
</tr>
<tr>
<td>Caller</td>
<td>The 16-year-old Hughes stunned the skating world -- and herself -- by upsetting Michelle Kwan to win the Olympic figure skating title. She has no plans to disappear now that she’s reached the pinnacle of her sport.</td>
<td></td>
</tr>
</tbody>
</table>

### 5. Corpus Verification Module Overview

The Corpus Verification Module is done through the web application where pre-qualified user can evaluate speech data via their audio plug-in. Once user is authenticated by the system, the verification page is displayed with random lists of raw speech data files of his/her qualified language. For users that is qualified for multiple languages, an option screen for their selection.
Native speakers of each language will do verification. The verifiers will be asked to listen to each utterance and decide if the speaker responded appropriately to the prompt. Each speech data files will be evaluated by at least two different verifiers. Logic will be built to ensure that raw data files will be not selected if it has been validated by same userId.

Figure 4 Corpus Verification Home Page

5.1 Verification Rating Criteria

A set of criteria will be used to determine if speech data can be acceptable or rejected. Major requirement are:

- Utterance must match data attributes, namely the gender, language and content.
  Using the same example as above
  CA_F_101_DaysOfWeek.WAV = Verifier must validate this speech is from a Cantonese female caller speaking all days of the week

- Content-independent utterance should be at least 30 seconds and spontaneous

- Too many dead silences in the utterance

- Significant background noise not due to telephone channel.

For those accepted into the data feed, a rating system is being assigned from 1 to 5 minimal to excellent result. Once the verifier submits rating, an entry is inserted into the database and rating field is recorded.

A daemon process will periodically move the acceptable data files from the raw feed directory into an archive to be used as speech recognition training data or testing data.
6. Conclusion

In this paper we have described the technical design, dialog design and recording of a large, multi-lingual corpus for language identification or speech recognition research. There are two main goals of this research, proving the flexibility and powerful usage of VoiceXML integration with web application, and attempt to design a large, multilingual corpus for automatic language detection. The speech corpus from this system depends on the quality of the native speaker who helps design the prompts, verification, phone equipment and the noise background. But the technical framework provides an appealing and efficient approach for quick deployments, inexpensive maintenance, and universal access to the same database of content, while providing a foundation for future multi-modal application development.

7. References