System for Automated Speech and Language Analysis (SALSA)

Kyle Marek-Spartz, Benjamin Knoll, Robert Bill, Thomas Christie, Serguei Pakhomov

University of Minnesota, Minneapolis, USA

`mare0132@umn.edu, knol0061@umn.edu, bill0154@umn.edu,`
`tchristie@umn.edu, pakh0002@umn.edu`

Abstract

SALSA automates cognitive test administration and scoring. These tests are used to characterize cognitive impairment resulting from neurodegenerative disease, traumatic brain injury, and drug toxicity; however, they are currently performed manually, limiting their utility for large clinical populations and longitudinal assessments. We present a fully automated, comprehensive system for collecting spoken test responses with mobile and telephony platforms, using an open-source automatic speech recognition engine (KALDI) to calculate a range of speech characteristics that may be useful in assessment of cognitive function.

Index Terms: speech recognition, cognitive testing

1. Introduction

A number of widely used cognitive test batteries include speech-based tasks such as picture description, picture naming, spontaneous narrative, and verbal fluency (naming words that begin with a letter of the alphabet or belonging to a specific category). While many of these tests were initially designed to assess aphasia, they have been demonstrated to be useful for assessment of other conditions that affect cognition including neurodegenerative disease and neurotoxic medications. Test administration and scoring is currently fully manual and requires trained personnel, thus limiting their use for routine assessment of large numbers of people for clinical or research purposes. This limitation can be overcome with automation. Indeed, validated computerized neuropsychological test batteries exist and are beginning to be widely used (e.g. Cogstate[1] and CANTAB[2]); however, these batteries tend to rely on keyboard-based interaction with the person being tested (measuring reaction time, for example) and currently do not include speech-based assessments. To address these limitations we developed a system based on automatic speech recognition (ASR) for analyzing spoken responses to cognitive tests.

2. SALSA

SALSA is an end-to-end web-based system that collects and processes audio recordings made during cognitive testing. SALSA consists of the following components:

- Database and file store.
- Web service.
- A set of multi-platform data-collection clients.
- Speech processing system.
- Web application for administrative tasks.

3. Database schema and file store

We use a relational database and file store (for audio) to keep metadata collected from tests, descriptions of what tests can be performed, file attachments, and for access control. The database and web services schema follows a typical neuropsychological test administration paradigm. In this paradigm, a subject comes in for one or more visits to participate in one or more tests. Thus, our databases stores metadata for *Projects, Subjects, Visits,* and *Tests*.

The database also stores descriptions of what tests our assessment clients can perform, consisting of *Test Prototypes* and *Stimuli*. *Test Prototypes* specify the *Stimuli* that will be presented to the subject during testing, their sequence and behavior (e.g. duration, ability to pause and resume, repeat, etc.), as well as processing directives used to define language and acoustic models and other speech processing parameters.

All files are stored as *Attachments* to projects, subjects, visits, tests, test prototypes, and stimuli. For example, *Test Attachments* can be audio recordings of a subject's test response, or a transcription of that recording, or a processing artifact. Attachments consist of metadata about the file and reference a specific file in our file store.

Lastly, the database provides rudimentary access control with *Clients, Privilege Lists,* and *Client Access* tables. Clients store client authentication information (username and password hash) and a boolean for whether they are an administrator. Privilege lists are a set of booleans for each kind of access, e.g. reading the subject table.

4. Web service

The SALSA web service provides a RESTful API using Python, Flask, and SQLAlchemy. JSON is used as the serialization protocol. The web service API provides several routes that specify access tables and parameters using HTTP verbs to specify interaction (i.e. `GET`s correspond to `SELECT` statements in the database and `POST`s to `INSERT`s).

For security, we use HTTPS combined with HTTP Basic authentication. The web service compares information supplied against the username and password hash in the database. Additionally, queries are denied or results are filtered according to access controls in the database.

5. Assessment clients

Our assessment clients collect data and report back to the web service. We have implemented assessment clients for multiple platforms, including iOS 6 and 7 and Windows 8. A secondary web service provides a telephony interface for tests that do not require visual stimuli (e.g. verbal fluency).

[1] http://www.cogstate.com
[2] http://www.cambridgecognition.com

Tablet-based assessment clients have two different interfaces. In the simplified interface, the project (or test prototype) is set upon installation. Subject information is entered, and a test begins. This interface is designed for subjects to use on their own, e.g. at a kiosk. The full interface is designed for test administrators. and allows for choosing between multiple projects without changing configuration, and creation of subjects and visits. Tests are stored in a local database in case of network connectivity issues, and subsequently uploaded.

The telephony assessment client consists of a thin web service that interfaces Twilio$^{\mathrm{TM}3}$ with our primary web service. When Twilio receives a call, it GETs a resource from our telephony client. We route Twilio phone numbers to specific test prototypes by specifying the initial route for Twilio to GET. If the test prototype specified is compatible with the telephony client (i.e. no images), the telephony client generates and responds with TwiML$^{\mathrm{TM}}$, Twilio's markup language used for describing call flows. The TwiML specifies which routes to fetch next under certain conditions. We collect a subject identifier, which is then verified to exist in the database. After the subject is confirmed to exist, we begin testing, telling Twilio to record subject responses if stimuli require recording. The telephony client utilizes Memcached and supplies its own stimuli attachments to minimize load on the primary web service.

6. Processing

SALSA can be configured to operate in three modes: ASR mode, forced-alignment mode, and VF-meter mode.

In all three modes we rely on KALDI, an open-source ASR toolkit [1]. A speaker-independent acoustic model trained for VF-Meter consists of a set of Hidden Markov Models that represent 88 base phones occurring in multiple acoustic contexts collected from a large corpus of general English speech (a combination of the Wall Street Journal and the TRAINS). The phone set of 88 phones was derived from the Carnegie Mellon University dictionary (CMU dictionary) of pronunciations and included 84 consonants and vowels with preserved stress marking, a special silence phone and special phones to represent speech noise, non-speech noise and filled pauses ('ah' and 'um') [2]. In all modes, the input speech signal is preprocessed by splitting it into 25 millisecond frames sifted by 10 milliseconds and each frame was coded as a standard set of 13 Mel-spectrum Frequency Cepstral Coefficients (MFCCs) with added delta coefficients, resulting in a vector of 26 coefficients for each frame. For each set of MFCC vectors representing the speech input frames, the KALDI ASR decoder was used to find the highest likelihood path through the lattice of hypotheses constructed based on the language and acoustic models described above. All language models were constructed separately for each individual task using the Stanford Research Institute LM toolkit [3].

In the *ASR mode*, we use a phoneme-level language model where bi-phone probabilities are estimated from the Carnegie Mellon University pronouncing dictionary [2]. The phoneme-level ASR output is then used to estimate utterance boundaries and calculate a set of acoustic characteristics including utterance count, mean duration of utterances and silent pauses, silent pause density (of various lengths), ratio of silence to speech, ratio of silence to total duration, mean utterance intensity standard deviation, mean F0 variability, hesitation count, hesitation rate, and speaking rate.

In the *forced-alignment mode*, the audio samples are first manually transcribed verbatim and are subsequently converted into deterministic networks used to force-align the transcriptions with the audio signal. Similarly, to the ASR-based mode, various speech characteristics are calculated; however, in this mode they are calculated based on the word rather than the utterance boundaries and thus tend to be more precise.

In the *VF-meter mode*, the language models are trained separately for each verbal fluency test. For semantic fluency tests, we trained a bigram language model for each category (animals, fruits/vegetables). For letter verbal fluency tests, we created simple unigram models using a subset of words from the CMU dictionary extracted separately for each letter.

After speech analysis has completed, the processing system reports results back to the web service.

7. Web application

The web application is used for project administration and transcription. It provides the following functionality:

- Querying the database, similar to the web service, but in human-readable form
- Traversing between linked objects in the database
- Creating objects in the database
- Transcribing recordings
- Starting individual and batch test processing
- Viewing attachments (including results)

Similar to the web service, it is implemented in Python using Flask. We reuse the same SQLAlchemy models from the web service to interact with the database. Again, HTTPS, HTTP Basic authentication, and access control from the database are used to secure the web application.

8. Acknowledgments

This work was supported in part by grants from NIH – NINDS (R01NS076665), Alzheimer's Association (DNCFI-12-242985), Geoffrey Beene Foundation Alzheimer's Disease Challenge, and the University of Minnesota Academic Health Center Faculty Development Grant.

9. References

[1] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *Proc. ASRU*, 2011, pp. 1–4.

[2] R. L. Weide, "The CMU pronouncing dictionary," 1998. [Online]. Available: http://www.speech.cs.cmu.edu/cgibin/cmudict

[3] A. Stolcke, "SRILM: an extensible language modeling toolkit." in *INTERSPEECH*, 2002.

[3] https://www.twilio.com/