Software Project Proposal: Generating Pronunciation Lexicons for Small-Vocabulary ASR in LRLs

Anjana Vakil (anjanav@coli.uni-saarland.de)
Max Paulus (mpaulus@coli.uni-saarland.de)

November 4, 2013

1 Project overview

Developers trying to incorporate speech recognition interfaces in a low-resource language (LRL) into their applications currently face the hurdle of not finding recognition engines trained on their target language. Although tools such as Carnegie Mellon University's Sphinx simplify the creation of new acoustic models for recognition, they require large amounts of training data (audio recordings) in the target language. However, for small-vocabulary applications, an existing recognizer for a high-resource language (HRL) can be used to perform recognition in the target language. This requires a pronunciation lexicon mapping the relevant words in the target language into sequences of sounds in the HRL.

Our goal is to build an easy-to-use application that will allow even naive users to automatically create a pronunciation lexicon for words in any language, using a small number of audio recordings and a pre-existing recognition engine in a HRL such as English. The resulting lexicon can then be used to add small-vocabulary speech recognition functionality to applications in the LRL.

2 Core functionality

A simple user interface allows the user to easily specify one orthographic form (text string) and and one or more audio samples (`.wav` files) for each word in the target vocabulary, and to set other options (e.g. number of pronunciations per word, name/save location of lexicon file, etc.). The audio is then passed to a speech recognition engine for a HRL (English). An automatic pronunciation generation algorithm such as Salaam [1–3] is employed to find the best pronunciation(s) for each word in the LRL vocabulary. The program outputs a pronunciation lexicon (`.pls` XML file). This lexicon file is then ready for direct inclusion in a speech recognition application, as it follows the standard pronunciation lexicon format [4].

3 Tools and platforms

The back-end requires an existing high-quality speech recognition engine with a usable API. The two main options under consideration are the Microsoft Speech Platform (http://msdn.microsoft.com/en-us/library/hh361572) and CMU's open-source ASR toolkit Sphinx (http://cmusphinx.sourceforge.net/).

The front-end interface will initially be console-based and require no special technology. If time allows, a graphical user interface will be implemented in a framework suitable to the back-end (see below).

4 Possible extensions

- GUI for web or desktop application (using Python or C#/.NET)
- Capability to record audio samples within the application
- Option to choose source HRL (recognizer language)

References

[1] Hao Yee Chan and Roni Rosenfeld. "Discriminative pronunciation learning for speech recognition for resource scarce languages". In: *Proceedings of the 2nd ACM Symposium on Computing for Development.* ACM DEV '12. Atlanta, Georgia: ACM, 2012, 12:1–12:6. URL: http://doi.acm.org/10.1145/2160601.2160618.

[2] Fang Qiao, Jahanzeb Sherwani, and Roni Rosenfeld. "Small-vocabulary speech recognition for resource-scarce languages". In: *Proceedings of the First ACM Symposium on Computing for Development.* ACM DEV '10. London, United Kingdom: ACM, 2010, 3:1–3:8. URL: http://doi.acm.org/10.1145/1926180.1926184.

[3] Jahanzeb Sherwani. "Speech interfaces for information access by low literate users". PhD thesis. Pittsburgh, PA, USA: Carnegie Mellon University, 2009. URL: http://reports-archive.adm.cs.cmu.edu/anon/anon/home/ftp/usr/ftp/2009/CMU-CS-09-131.pdf.

[4] World Wide Web Consortium (W3C). *Pronunciation Lexicon Specification (PLS) Version 1.0.* Tech. rep. 2008. URL: http://www.w3.org/TR/pronunciation-lexicon/.