lex4all: A language-independent tool for building and evaluating pronunciation lexicons for small-vocabulary speech recognition

Abstract

This paper describes *lex4all*, an open-source PC application for the generation and evaluation of pronunciation lexicons in any language. With just a few minutes of recorded audio and no expert knowledge of linguistics or speech technology, individuals or organizations seeking to create speech-driven applications in low-resource languages can use this tool to build pronunciation lexicons enabling small-vocabulary speech recognition in the target language using a high-quality commercial recognition engine designed for a high-resource source language (e.g. English). This is possible thanks to an existing algorithm for cross-language phoneme-mapping; we give an overview of this method and describe its implementation in *lex4all*. Beyond the core functionality of building new lexicons, the tool also offers a built-in audio recorder that facilitates data collection, and an evaluation module that simplifies and expedites research on small-vocabulary speech recognition using cross-language mapping.

1 Introduction[1]

In recent years it has been demonstrated that speech recognition interfaces can be extremely beneficial for applications in the developing world, particularly in communities where literacy rates are low or where PCs and internet connections are not always available (Sherwani and Rosenfeld, 2008; Bali et al., 2013; Sherwani, 2009). Typically, the languages spoken in such communities are under-resourced, such that the large audio corpora typically needed to train or adapt recognition engines are unavailable. However, in the absence of a recognition engine trained for the target low-resource language (LRL), an existing recognizer for a completely unrelated high-resource language (HRL), such as English, can be used to perform small-vocabulary recognition tasks in the LRL. All that is needed is a pronunciation lexicon mapping each term in the target vocabulary to one or more sequences of phonemes in the HRL, i.e. phonemes which the recognizer can model.

...

2 Background and related work

Many commercial speech recognition systems offer high-level Application Programming Interfaces (APIs) that make adding voice recognition capabilities to an application as simple as specifying (in text) the words/phrases that should be recognized; this requires very little general technical expertise, and virtually no knowledge of the inner workings of the recognition engine. If the target language is supported by the system – the Microsoft Speech Platform, for example, currently supports recognition and synthesis for 26 languages/dialects (Microsoft, 2012) – this makes it very easy for small-scale software developers (i.e. individuals or small organizations without much funding) to create new speech-driven applications.

While many such individuals or organizations in the developing world may be interested in using such platforms to create speech-driven applications for use in their communities, the low-resource languages typically spoken in these areas are obviously not supported by such commercial systems. And though many effective techniques for training or adapting recognizers for new lan-

[1] Parts of this paper (Sections 1 and 2) overlap with a paper submitted to the 4th Workshop on Spoken Language Technologies for Under-resourced languages (SLTU '14, `http://www.mica.edu.vn/sltu2014`). That paper, which is currently under review, concerns related research not reported here, and makes no mention of the lex4all application.

guages exist (see e.g. the CMUSphinx toolkit[2] or the Rapid Language Adaptation Toolkit[3]), these typically require hours of training audio to produce effective models, and even the highest-level tools for building new models still require a non-trivial amount of expertise with speech technologies; such data and expertise may not be available to the small-scale developers in question.

However, many useful development-oriented applications (e.g. for accessing information or conducting basic transactions) require only small-vocabulary recognition tasks, by which we mean those requiring discrimination between a few dozen terms. For such tasks, an unmodified HRL recognizer can be used as-is to perform recognition of the LRL terms; we simply need an application specific grammar describing the allowable combinations and sequences of words/phrases to be recognized, and a pronunciation lexicon which maps each of the target words/phrases to a sequence of phonemes in the source language for which the recognizer has been trained.

This is the thinking behind the Speech-based Automated Learning of Accent and Articulation Mapping (Salaam) method (Sherwani, 2009; Qiao et al., 2010; Chan and Rosenfeld, 2012). This method of cross-language phoneme-mapping enables the automatic discovery of source-language pronunciation sequences for words or phrases in the (unrelated) target language, and thus constitutes the foundation on which the lex4all tool is built.

The basic idea of phoneme-mapping is to discover the best pronunciation sequence for a given word in the target language by using the source language recognizer to perform phone decoding on one or more audio samples of the target word. However, the APIs for commercial recognizers such as Microsoft's are designed for word-decoding, and do not usually enable the use of the phone-decoding mode. The insight of the Salaam approach is to use a specially designed grammar to mimic this phone decoding (Chan and Rosenfeld, 2012, §3.2). Specifically, Qiao et al. (Qiao et al., 2010, 4.1) create a recognition grammar representing a phoneme "super-wildcard" that guides pronunciation discovery. This grammar allows the recognizer to treat an audio sample of the target word as a "phrase" made up of 0-10 "words", where each "word" can be matched to any possible sequence of 1, 2, or 3 source language phonemes (Qiao et al., 2010, 4.1).

Given this super-wildcard grammar and one or more audio recordings of the target word, Qiao et al. (Qiao et al., 2010, 4.1) use an iterative training algorithm to discover the best pronunciation(s) for that word, one phoneme at a time. In the first pass, the recognizer finds the best match(es) for the first phoneme, then for the first two phonemes in the second pass, and so on until a stopping criterion is met, e.g. the recognition confidence score assigned to the resulting "phrase" stops improving (Qiao et al., 2010, p. 4).

Compared to expert-crafted pronunciations, using pronunciations generated automatically by this algorithm improves recognition accuracy substantially (Qiao et al., 2010, 5.2). By training on samples from two speakers instead of one, and by using a pronunciation lexicon containing multiple pronunciations for each word (i.e. the n-best results of the training algorithm instead of the single best result), Qiao et al. are able to further improve accuracy. Chan and Rosenfeld (Chan and Rosenfeld, 2012) achieve still higher accuracy by applying an iterative discriminative training algorithm, identifying and removing pronunciations that cause confusion between word types.

In sum, the Salaam method is fully automatic, requiring expertise neither in speech technology (to modify acoustic models) nor in linguistics (to manually generate seed pronunciations), and for each new target language it requires only a few minutes' worth of training data from one or more speakers, an amount that can be collected in a short time with little effort or expense. At the same time, it enables the construction of pronunciation lexicons that can help bring speech recognition applications to LRLs. This has been demonstrated in at least two developing-world projects that have successfully used the Salaam method to add voice interfaces to real applications: an Urdu telephone-based health information system in Pakistan (Sherwani, 2009), and a text-free Hindi smartphone application to deliver agricultural information to farmers in rural India (Bali et al., 2013).

Our work directly builds on the Salaam method, wrapping it in an easy-to-use graphical application so that non-expert users can take advantage of this algorithm to quickly and easily create the pronunciation lexicons necessary for creating speech in-

[2] http://www.cmusphinx.org

[3] http://i19pc5.ira.uka.de/rlat-dev

terfaces in LRLs using existing HRL recognizers. In the following sections, we describe the lex4all application and the modified implementation of the Salaam method which is at its core.

3 System overview

4 User interface

Easy for non-experts (include informative screenshot)

Audio input: use existing or new

4.1 Audio recording

Things to mention (Max, please add to this):

- NAudio

5 Pronunciation mapping

5.1 Implementation of the Salaam method

(Qiao et al., 2010)

5.2 Running time

Evaluation results in table/graph

5.3 Discriminative training

(Chan and Rosenfeld, 2012)

6 Evaluation module for research

7 Conclusion and future work

References

Kalika Bali, Sunayana Sitaram, Sebastien Cuendet, and Indrani Medhi. 2013. A Hindi speech recognizer for an agricultural video search application. In *Proceedings of the 3rd ACM Symposium on Computing for Development*, ACM DEV '13, pages 5:1–5:8, New York, NY, USA. ACM.

Hao Yee Chan and Roni Rosenfeld. 2012. Discriminative pronunciation learning for speech recognition for resource scarce languages. In *Proceedings of the 2nd ACM Symposium on Computing for Development*, ACM DEV '12, pages 12:1–12:6, New York, NY, USA. ACM.

Microsoft. 2012. Language support. In *Microsoft Speech Platform SDK 11 Documentation*. http://msdn.microsoft.com/en-us/library/hh378476.

Fang Qiao, Jahanzeb Sherwani, and Roni Rosenfeld. 2010. Small-vocabulary speech recognition for resource-scarce languages. In *Proceedings of the First ACM Symposium on Computing for Development*, ACM DEV '10, pages 3:1–3:8, New York, NY, USA. ACM.

Jahanzeb Sherwani and Roni Rosenfeld. 2008. The case for speech technology for developing regions. In *Proc. HCI for Community and International Development, Florence, Italy*. ACM.

Jahanzeb Sherwani. 2009. *Speech interfaces for information access by low literate users*. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, USA.