8-2013

Formalizing Expert Knowledge for Developing Accurate Speech Recognizers

Anuj Kumar
Carnegie Mellon University, akumar1@andrew.cmu.edu

Florian Metze
Carnegie Mellon University, fmetze@andrew.cmu.edu

Wenyi Wang
Carnegie Mellon University

Matthew Kam
American Institutes for Research

Published In
Proceedings of INTERSPEECH, 1121-1125.

Formalizing Expert Knowledge for Developing Accurate Speech Recognizers

Anuj Kumar[1], *Florian Metze*[1, 2], *Wenyi Wang*[2], *Matthew Kam*[1, 3]

[1]Human-Computer Interaction Institute, Carnegie Mellon University, Pittsburgh, PA, USA
[2]Language Technologies Institute, Carnegie Mellon University, Pittsburgh, PA, USA
[3]American Institutes for Research, Washington, D.C., USA

anujk1@cs.cmu.edu, fmetze@cs.cmu.edu

Abstract

The expertise required to develop a speech recognition system with reasonable accuracy for a given task is quite significant, and precludes most non-speech experts from integrating speech recognition into their own research. While an initial baseline recognizer may readily be available or relatively simple to acquire, identifying the necessary accuracy optimizations require an expert understanding of the application domain as well as significant experience in building speech recognition systems. This paper describes our efforts and experiments in formalizing knowledge from speech experts that would help novices by automatically analyzing an acoustic context and recommending appropriate techniques for accuracy gains. Through two recognition experiments, we show that it is possible to model experts' understanding of developing accurate speech recognition systems in a rule-based knowledge base, and that this knowledge base can accurately predict successful optimization techniques for previously seen acoustic situations, both in seen and unseen datasets. We argue that such a knowledge base, once fully developed, will be of tremendous value for boosting the use of speech recognition in research and development on non-mainstream languages and acoustic conditions.

Index Terms: adaptations, knowledge base, speech recognition

1. Introduction

Speech-user interfaces (SUIs) such as Apple's Siri [1], Samsung's S Voice [2] and Google's Voice Search [3] are emerging as a popular alternative to graphical user interfaces. However, despite years of research, these interfaces work well only for specific user groups, such as adult native speakers of English, and in specific acoustic situations such as non-noisy backgrounds, when in fact, users such as non-native speakers or children stand to benefit at least as much, if not more. The problem in developing SUIs for non-traditional users and non-typical acoustic and language contexts is the expertise, time and cost needed in building a reasonably accurate initial speech recognition system, which can then be used in iterative development of other components e.g. the user interaction. It is particularly difficult for non-speech experts (or novices) such as researchers in human-computer interaction who wish to test their ideas with a working speech system, but instead routinely resort to Wizard-of-Oz experiments that are unable to test real usage of the system. At the same time, speech recognition technology has reached a point where, given enough data, a working recognizer can be developed for any scenario.

Researchers who are beginners in speech recognition, or are novices can benefit from a "speech recognition" knowledge base (KB) and associated toolkit because: (A) it is difficult to find speech recognition experts easily, and (B) even when an expert is available, hiring them for a project can be expensive. In simplifying the task for novices, we focus on providing automatic guidance about the type of optimizations to perform, since optimizations are the most challenging tasks in the development process [4][5][6][7]. To do this, we take the view that well-trained speech experts who routinely build working recognizers have accumulated years of experiential knowledge that is hard for them to explicitly teach to non-experts or novices, but by observing them in action, we can study and formalize their tacit knowledge. This formalized knowledge can then be used for the benefit of novices for automatic analysis and recommendation of appropriate optimization techniques.

Accordingly, we present two contributions: first, we detail the process and results from interviews with professional speech developers that led to formalizing expert intuition and know-how in a knowledge base. Second, using two datasets, we outline empirical results that evaluate the above knowledge base in its ability to predict correct optimization steps to a non-expert. As such, the work presented in this paper should be seen as a feasibility study towards automatically identifying and recommending optimization steps for the benefit of non-experts.

2. Related Work

Many systems that model experts' knowledge for a specific domain (also known as expert systems) have been successfully developed in the past, e.g. see review by [8]. For instance, in the medical domain, MYCIN [9] was one of the first expert systems that recommended antibiotics for bacterial infection. It took a set of symptoms as input, found the best rule that matched the criteria, and then, recommended the corresponding antibiotic. In a study conducted by Shortliffe et al. [9], MYCIN out-performed the recommendations made by several doctors. Since then, we have seen a surge in rule-based expert systems, such as, in recommending mortgages [10], tax consultation [11], or also in computational domains, such as recommending correct cleansing technique in big data [12], or finding issues in code that limit interoperability of web interfaces [13]. To our knowledge, no expert system to date has been developed to guide the development of an accurate speech recognizer.

At the same time, researchers have attempted to tackle the problem of enabling non-experts to rapidly develop and test speech-user interfaces in a number of other ways. For instance, SUEDE [14] is a toolkit that allows any user-interface designer to rapidly mock-up a prompt/response speech interface and test it in a Wizard-of-Oz study. It does not, however, support development of a working recognizer. Another toolkit, SPICE [15] supports rapid development of a baseline recognizer for new languages. It does so by allowing any researcher to input a set of

audio files, corresponding phoneme set, and a dictionary to generate the baseline acoustic model. It does not automatically perform acoustic or language context-specific adaptations, when in fact these adaptations are key in improving accuracy [5][6][7]. Open-source toolkits such as Kaldi [16] support development of both the baseline recognizer and an adapted version; however, they don't provide any automatic guidance on what adaptation(s) to perform, and leave it to the developer's expertise to understand the context of the application, and apply the suitable adaptation technique(s). As a result, non-expert researchers find them substantially difficult to use. For instance, in 2012 alone, a discussion forum of a widely used recognizer, Sphinx [17] saw over 6000 posts with over 1000 unique topics from non-experts asking help on various issues of speech recognition, e.g. adapting acoustic models, generating pronunciation variants, etc.

Another method of developing "working" recognizers is to collect a large amount of training data from representative speakers (> hundreds of hours), and then train an acoustic model on those files. Google [3] and Apple [1], for instance, follow this model. However, this approach is very costly, and collection of large speech corpora is a research challenge in and of itself [18].

3. Knowledge Elicitation Methodology

As is common in knowledge engineering literature [19] for eliciting knowledge from experts, we conducted semi-structured interviews with five experts, all senior PhD students at CMU. Later in this paper, we will call them "old" experts. The interview methodology draws on ideas from distributed cognition [20][21], which posits that knowledge lies not only with the individual, but is also distributed over the tools in the individual's social and physical environment. Thus, our interviews covered questions on experts' interaction with: (i) other individuals, e.g. data collectors, application designers, or other experts, and (ii) with the machines, tools, data sources, etc., e.g. listening to audio files using an audio processing tool [22][23]. We divided the interviews in two phases: first, we asked the participants to describe a general adaptation process, the common challenges they faced, the people they consulted, and the tools they used. For the second phase, we observed them in-action for a speech recognition optimization task. We gave each expert a dataset from a project that contained utterances from Indian children, recorded in a noisy background on a mobile phone. Since this dataset contained several potential degrading factors, e.g. noise, children's speech, accent, etc., we felt it was a good choice to understand an expert's thought process while developing recognizers for challenging contexts. We asked each expert to explain the steps (by thinking aloud) that they would take to build the best recognizer on this dataset.

Post interviews, the transcripts of the above interviews became the basis for line-by-line open coding process [24] to identify relevant concepts and themes that enhanced our understanding of the optimization process and the associated intuition. Specific instructions or intuitions by the experts were formulated as rules. Once formulated, the same experts vetted these rules for consistency and accuracy of formulation.

4. Knowledge Formalization

The analysis of the above interviews led to two results: (A) a general process that experts follow while optimizing for accuracy, and (B) a set of rules that guide their choices at each step in the above process. Below we detail each of them.

4.1. Accuracy Optimization Process and Rules

The accuracy optimization process starts with experts receiving a small set of audio data (and associated transcripts), which is recorded in the representative setting of the application use. The expert's task is to develop a recognizer with best possible accuracy. A general process is summarized in five steps:

1. **Data validation and cleaning**: this step involves performing several sanity checks, e.g. validating that the transcripts are free of spelling errors, deciding how to handle half words in the dictionary, detecting out-of-vocabulary words, splitting adaptation data in tune, validation and test sets, what parts of data to label further to improve results [25], deciding whether to pick an existing "off-the-shelf" acoustic model for adaptation or build a new one, etc. For each such issue, we documented how the experts handled the specific situation: for instance, a part of a larger rule to decide whether or not to train a new acoustic model is: "*IF no matching "off-the-shelf" baseline model found, AND app. = dialogue system for single speaker, AND tune data > 1 hour of recording, AND Vocabulary Size < 60 words, THEN train your own acoustic model …*"[1]
2. **General adaptation**: experts separated most of the optimization techniques into three categories: general transformations, context- and user-specific adaptations, and recognizer tuning. Assuming that the experts are adapting the baseline models, this step concerns general transformations such as linear and Gaussian transformations for feature and model parameters. The idea is that barring concerns about overfitting, general adaptations tend to almost always improve recognition accuracy as this set of adaptations transform the baseline parameters closer to the characteristics of the tuning set, e.g. channel, noise, accent, etc. Several rules were documented for this step. E.g.: "*IF Adaptation Data Size < 20 sec per-speaker THEN perform Global mean-only MLLR*", "*IF Adaptation Data Size > 100 utterances per-speaker, THEN first perform MLLR using first 100 files, and next switch to MAP*."
3. **Context- and user-specific adaptation**: experts next explored recognition errors to identify the degrading factors i.e. factors that are negatively impacting recognition accuracy, e.g. noise, accent, pitch, speaking rate, etc. by manually listening to a few audio files that were incorrectly recognized and mentally comparing them with those used in the baseline model to understand the point(s) of difference. They would then use normalization or adaptation techniques specific to these degrading factors. In our experiments (as we will discuss in the next section), we automate the above qualitative process. The goal is to see whether an automated method could identify similar issues as the manual inspection by experts. To do so, we extract quantitative values for several degrading factors (as in section 5.2) for each audio file in the test set, and it's recognition result from the recognizer after step 2. Next, we perform univariate and multivariate regression to identify the impact of each factor on recognition accuracy [27]. Based on the statistical significance ($p<0.05$), we identify the significant

[1]Note that the rules in this paper are rewritten for readability; actual rules are instituted in a popular rule management system, CLIPS [26]. Also, due to space limitations, we cannot provide the entire set of rules, but provide various examples for the reader to follow.

factors, and based on the coefficient of correlation, we rank them from most to least impact. Next, we perform corresponding adaptations, as enlisted in the rules from the expert interviews. E.g. "*If degrading factor = SNR, perform Cepstral Variance Normalization; If degrading factor = F0, perform Vocal Tract Length Normalization…*"

4. **Tuning overall recognizer parameters**: this step includes tweaking recognizer parameters such as insertion penalty, silence penalty, or changing the language model weight based on inspecting the type of errors in the recognition output from step 3. It also includes analyzing confusion pairs, and possibly recommending alternative phrases to change the interface dialogue for reducing common errors.
5. **Final check for overtraining**: this step includes checking if the models have been trained on sufficient data, e.g. each Gaussian in the speech recognizer acoustic model should be trained or tuned on a minimum amount of data.

4.2. Why are rules a good representation for the KB?

We chose a rule-based knowledge representation for two reasons: (1) Ease of representation: we recognized that experts had developed many "unspoken rules" that followed an "IF … THEN … ELSE …" structure, and (2) Scalability: previous research in rule-based systems has shown that it is relatively simple for knowledge engineers (or domain experts) to add new techniques in a rule system [28]. This would be of importance, given the ever-expanding needs of speech recognition.

4.3. Dealing with Conflicts

On various occasions, we received conflicting recommendations from the experts. For instance, such conflicts arose for the initial setup condition, e.g. whether to start with an existing baseline or train a new acoustic model. When this happened, we instituted both options as alternates in the knowledge base, and ranked them based on the number of experts that had recommended the option. In the event a rule with multiple options got triggered, the knowledge base would first recommend the top most option; and if the non-expert developers were unsatisfied with accuracy, they could query the knowledge base again for an alternative, if any existed. Based on the results of what actually worked, the developer could provide feedback to the knowledge base that would update the ranking of the alternatives.

4.4. Characteristics of the Knowledge Base (KB)

In addition to the interviews, we also reviewed over 15 publications on speech adaptations and error analysis, e.g. see [6][29][30][31] to further understand techniques that experts use. Based on these, our current knowledge base covers the following variations spanning over 80 rules:

1. Application: isolated words, dialogue system, or dictation.
2. Recording Device: external headset, telephony, etc.
3. Speakers: single, multiple, speaker-independent or not.
4. Vocabulary Size: small (<60 words), medium (60-500 words), large (>500 words).
5. Vocabulary Type: expressions, technical, conversation, etc.
6. Adaptation Data Size: small (1hr), medium (5hrs), large (10hrs), very large (>50hrs).
7. Availability of Labels: exists for the entire set or partial set.
8. Noise Conditions: quiet room, street, competing talkers, etc.
9. Speaker Factors: age, gender, emotion, dialect, etc.

5. Evaluation

5.1. Datasets

We used the following two datasets in our evaluation:

5.1.1. "Seen" Dataset A: Indian Children's Speech

Dataset A – also used during the knowledge elicitation process – comes from another project of the authors [32]. In total, it comprises of 6250 single-word English utterances from 50 rural Indian children (~2 hours of speech). The speakers are equally split across gender and grades 4-5. Each child spoke 125 words selected from a set of 325 words, ensuring equal coverage for all words. The audio recordings were made on Nokia N810 using a close-talk microphone in out-of-classroom, noisy backgrounds. In our experiments, we set aside data from 10 speakers (5 males) as test set, and the remaining available as tune set.

5.1.2. "Unseen" Dataset B: CMU Kids Corpus

Dataset B, popularly known as "CMU Kids Corpus" [33], comprises of English sentences read aloud by 76 children (24 males). In total, the dataset has 5180 recordings where each recording represents one sentence. The recordings were made in a classroom, and there are some files with background noise. There are two types of speakers: "good" readers (44) and "poor" readers (32). For the latter, the speech has some typical reading errors that children make when not confident about a text. In our experiments, we set aside data from 16 "poor" speakers (8 males) as test set, and remaining as tune set.

5.2. Annotation

We annotated our data on the following features that correlate with popular reasons of recognizer failure. These were used in picking the rule from step 3 of section 4.1.

1. **Pronunciation score** measures the correctness in pronunciation on a scale of 1-10. To calculate the score, we first force-aligned the audio file with its transcript to obtain a phone-segmented file. Next, we calculated several aggregated statistics of each phone from these files and compared them to exemplar statistics of native English pronunciations from TIMIT dataset to get a score [34].
2. **Signal-to-noise ratio (SNR)** measures the strength of the signal with respect to the background noise.
3. **Fundamental frequency** (F0) of voice is the lowest frequency at which human vocal cord vibrates. **Formant frequencies (F1-F3)** are its spectral peaks. These were calculated using Praat [35], and measured in Hz.
4. **Speaking Rate (SPR)** is calculated as the average time for the user to say one phoneme i.e.
   $$SPR = \frac{number\_of\_phonemes}{start\_speech\_signal(ms) - end\_speech\_signal(ms)}$$
5. **Intensity** is the sound power per-unit area and is a measure of how loud the sound is. It was also calculated using Praat.

5.3. Experiment 1: Dataset A – Old Experts vs. KB

After annotating our datasets, the first question we sought to answer was whether the KB correctly modeled experts' knowledge and whether we could quantitatively identify the degrading factors that the experts recognized through "manual inspection" (step 3 of section 4.1). In other words, we wanted to

assess its internal validity. To do so, we compared the recommendations made by the KB with those made by the "old" five experts interviewed earlier, and how those translated into accuracy gains. The objective was to build a speaker-independent isolated word recognizer for 325 words in dataset A.

Table 1 summarizes the recommendations from the experts and the KB, and the corresponding accuracy gains. Overall, the KB outperformed all experts on test set for Dataset A. One point of difference came from starting with an existing acoustic model (AM) such as WSJ and then adapting it for the tune set using general adaptation rules such as MLLR and MAP (E1-3), or training one from scratch (E4-5 & KB) using the tune set. E1-3 had reasoned that the amount of tune data available for training a new AM is very small, and so starting with an existing AM is best; whereas E4-5 had reasoned that the characteristics of the data were different from any available AM, and given that it's a small vocabulary recognition task, it would be best to train a new AM. While designing the KB, we had instituted both the options as alternatives for small vocabulary, low-resource situations. Given that 3 experts had recommended to adapt an existing model, the KB first recommended that option, but once the low accuracy was noticed, it switched to the next best alternate i.e. train a new acoustic model. In addition, after "seeing" the results, it updated the ranking of the two alternatives.

Optimization Techniques	**Experts**					**KB**
	E1	**E2**	**E3**	**E4**	**E5**	
1- Baseline: existing AM	94.7	94.7	94.7			94.7
1- Baseline: train AM				25.4	25.4	25.4
2- MLLR	77.3	77.3	77.3			
2- MAP	78.3	78.3				
3- VTLN		76.9	76.9		24.9	
3- Dictionary			71.7	22.2		22.2
3- Frame-rate						20.8
Final	**78.3**	**76.9**	**71.7**	**22.2**	**24.9**	**20.8**

Table 1: *Word error rates (in %) for test set in Dataset A; the numbers in the first column are the step nos. from Section 4.1*

From the univariate and multivariate linear regression, accent (i.e. pronunciation score) and speaking rate (SPR) turned out to be significantly correlated with accuracy. This implied that they significantly impacted accuracy and their impact did not suffer from interaction effects with other factors. E3 and E4 too, from their manual inspection, recognized accent as an issue and recommended adding pronunciation variants to the dictionary as a potential solution (as in Table 1), while E1 acknowledged accent as an issue, but mentioned that adding pronunciation variants may not result in accuracy gains. Instead, he was working on a new accent modeling technique, but since it was not published yet, we did not include it in our analysis. None of the experts, identified SPR as an issue, when in fact, frame-rate tuning had an (minor) improvement on the accuracy results.

5.4. Experiment 2: Dataset B – New Experts vs. KB

Our second evaluation focused on assessing the generalizability of the KB for an unseen dataset B, and comparing its recommendations against new experts (E6 & E7) i.e. assess its external validity. We asked E6 & E7 to develop a speaker-independent reading speech recognizer for "poor" speakers, as in Dataset B. Table 2 summarizes their recommendations and the accuracy comparison with those from KB. The KB correctly identified data cleaning issues such as half words (E6-7) and spelling errors (E7), and recommended that we train our own acoustic model (E7). E6, however, felt that since we had access to data from "good" readers, we should first train a "high-quality" model on good speakers, and then use it to find the alignments on the data from poor readers. This technique, known as model bootstrapping, is useful in low-resource situations, especially when you have access to a high-quality recognizer. Since we did not have a high-quality recognizer when the KB was developed, this technique was not covered by it. None-the-less, it is simple to add because we have a clear definition of its use i.e. "presence of high-quality recognizer", and "low-resource situation". For the next optimizations, the regression correctly pointed SNR (E6) and SPR (E7) as the degrading factors.

Techniques	**E6**	**E7**	**KB**
1- Add half words to dict.	✔	✔	✔
1- Correct spelling errors		✔	✔
1- Baseline: train AM		56.1	56.1
1- Baseline: train AM + model bootstrapping	42.2		
3- CVN (SNR)	40.9		54.3
3- Frame-rate (SPR)		54.8	52.2
Final	**40.9**	**54.8**	**52.2**

Table 2: *Word error rates (in %) for test set in Dataset B*

6. Conclusion & Future Directions

With this work, we aim to lower the difficulty that novices encounter when building accurate speech recognizers. In particular, we present the design and characteristics of an initial rule-based knowledge base that models experts' understanding of speech recognition issues. Further, we assess if with the help of this knowledge base (KB), we can automatically recommend appropriate optimization steps. Our results show that the KB can successfully predict optimization techniques for previously seen acoustic situations during the interviews or the papers reviewed, both in seen and unseen datasets. Moreover, by the nature of its rule design, it can be expanded to incorporate additional insights on specific acoustic and language contexts by other experts [28].

We are exploring several extensions of this research. First, it is important to note that the task of building an accurate recognizer consists not only of applying the most suitable m optimization techniques from the available n, but might also require changes to the interface, e.g. replacing a word with its semantic equivalent to reduce confusion pairs (as in dialogue systems). Providing this level of guidance is a natural extension of our work. Second, we will look at the types of difficulties that an expert might have while inserting new rules. Will that lead to conflicts in the KB, and if so how will they be resolved? Finally, we are exploring how we can host this infrastructure over the web or in a Virtual Machine, so that the KB can improve from the experiences of test runs of many potential users.

7. Acknowledgements

This research is NSF funded: grant IIS-1247368 "EAGER: A Research Infrastructure for Analyzing Speech-based Interfaces," and CNS-1205589 "The Speech Recognition Virtual Kitchen."

8. References

[1] Apple – iOS6 – Use your voice to do even more, http://www.apple.com/ios/siri/, last accessed Feb. 15, 2013.

[2] Samsung Galaxy SIII | S Voice, (accessed Mar. 18, 2013) http://www.samsung.com/global/galaxys3/svoice.html.

[3] Voice Search – Inside Search – Google, http://www.google.com/insidesearch/features/voicesearch/index.html, last accessed Feb. 15, 2013.

[4] Shinoda, K. "Speaker Adaptation Techniques for Automatic Speech Recognition." *Proc. APSIPA ASC 2011 Xi'an*, 2011.

[5] van Doremalen, J., Helmer S., and Catia C. "Speech Technology in CALL: The Essential Role of Adaptation." Interdisciplinary Approaches to Adaptive Learning: A Look at the Neighbours: First International Conference on Interdisciplinary Research on Technology, Education and Communication, ITEC 2010, Kortrijk, Belgium, May 25-27, 2010. Revised Selected Papers. vol. 126. Springer, 2011.

[6] Woodland, P. C. Speaker adaptation for continuous density HMMs: A review. *Proc. of the ISCA Workshop on Adaptation Methods for Speech Recognition* (pp. 11–19). Sophia-Antipolis, France, 2001.

[7] The Importance of Adaptation in Speech Recognition, http://www.speechtechmag.com/Articles/Editorial/Sounding-Board/The-Importance-of-Adaptation-in-Automatic-Speech-Recognition-68360.aspx, last accessed Feb. 11, 2013.

[8] Compton, P. "Situated Cognition and Knowledge Acquisition Research," International Journal of Human-Computer Studies, 71, 184-190, 2013.

[9] Shortliffe, E.H. "Computer-based Medical Consultations: MYCIN," Elsevier, New York, 1976.

[10] Steinmann, H., Chorafas, D.N. "Expert systems in banking: a guide for senior managers," New York: New York University Press, 222–225, 1990.

[11] Shpilberg, D., Graham, L., and Schatz, H. ExperTAX^{sm}: An expert system for corporate tax planning. Expert Systems, 3: 136–151, 2012.

[12] Dani, M.N., Faruquie, T.A., Garg, R., Kothari, G., Mohania, M.K., Prasad, K.H., Subramaniam, L.V., Swamy, V.N. "Knowledge Acquisition Method for Improving Data Quality in Services Engagements," In: Proceedings of the IEEE International Conference on Services Computer, Miami, pp. 346–353, 2010.

[13] O'Connor, M., Knublauch, H., Tu, S.W., Musen, M.A. "Writing Rules for the Semantic Web using SWRL and Jess," Protégé With Rules Workshop, Madrid, 2005.

[14] Klemmer, S., Sinha, A., Chen, J., Landay, J., Aboobaker, N., Wang, A. "SUEDE: A Wizard of Oz Prototyping Tool for Speech User Interfaces," UIST: ACM Symposium on User Interface Software and Technology, 2001.

[15] Schultz, T., Black, A., Badaskar, S., Hornyak, M., and Kominek, J. "SPICE: Web-based Tools for Rapid Language Adaptation in Speech Processing Systems," Interspeech, Belgium, 2007.

[16] Kaldi, http://kaldi.sourceforge.net/, last accessed Feb. 11, 2013.

[17] CMU Sphinx, http://sourceforge.net/p/cmusphinx/discussion/, last accessed Feb. 11, 2013.

[18] Baker, J., Deng, L., Khudanpur, S., Lee, C., Glass, J., and Morgan, N. "Historical Development and Future Directions in Speech Recognition and Understanding," MINDS Report of the Speech Understanding Working Group, NIST, 2007.

[19] Schreiber, G., Akkermans, H., Anjewierden, A., de Hoog, R., Shadbolt, N., Van de Velde, W., Wielinga, B. "Knowledge Engineering and Management: The CommonKADS methodology," MIT Press, Cambridge Mass, 2000.

[20] Hutchins, E. "Cognition in the wild," MIT Press, 1995.

[21] Perry, M. "Distributed Cognition," In J.M. Carroll (Ed.) HCI Models, Theories, and Frameworks: Towards and Interdisciplinary Science. Morgan Kaufmann. pp. 193-223, 2003.

[22] Norman, D.A. "Things that make us Smart. Defending Human Attributes in the age of Machine". New York: Addison-Wesley, 1993.

[23] Perkins, D.N. "Person-plus: A Distributed View of Thinking and Learning," In G. Salomon (Ed.) Distributed Cognitions. Psychological and Educational Considerations. Cambridge University Press, pp. 88-110, 1993.

[24] Strauss, A.C. and Corbin, J. "Basics of Qualitative Research". Sage Publications, 1998.

[25] Nallasamy, U., Metze, F., Schultz, T. "Active Learning for Accent Adaptation in Automatic Speech Recognition," *Spoken Language Technology Workshop (SLT), 2012 IEEE*, vol., no., pp.360, 365, Dec. 2012.

[26] CLIPS, A Tool for Building Expert Systems, http://clipsrules.sourceforge.net/, last accessed Mar 12, 2013.

[27] Haberman, S.J. "Analysis of Qualitative Data: New Developments," vol 2, Academic Press, New York, 1979.

[28] Richards, D. "Two Decades of Ripple Down Rules Research," Knowledge Engineering Review 24 (2), 159-184, 2009.

[29] Goronzy, S., and Ralf K. "A combined MAP+ MLLR approach for speaker adaptation." *Proceedings of the Sony Research Forum*. Vol. 99, No. 1, 1999.

[30] Yee, C. and Rosenfeld, R. "Discriminative pronunciation learning for speech recognition for resource scarce languages." *Proceedings of the 2nd ACM Symposium on Computing for Development*. ACM, 2012.

[31] Bali, K., Sitaram, S., Cuendet, S., & Medhi, I. A Hindi speech recognizer for an agricultural video search application. In *Proceedings of the 3rd ACM Symposium on Computing for Development* (p. 5). 2013.

[32] Kumar, A., Reddy, P., Tewari, A., Agrawal, R., Kam, M. "Improving Literacy in Developing Countries Using Speech Recognition-Supported Games on Mobile Devices." In ACM Conference on Human Factors in Computing Systems (CHI), Austin, Texas, May 5-10, 2012.

[33] Eskenazi, M., J. Mostow, and D. Graff. "The CMU kids speech corpus." Corpus of children's read speech digitized and transcribed on two CD-ROMs, with assistance from Multicom Research and David Graff. Published by the Linguistic Data Consortium, University of Pennsylvania, 1997.

[34] Srikanth, R., Bo, L., and Salsman, J. "Automatic Pronunciation Evaluation And Mispronunciation Detection Using CMUSphinx" Google Summer of Code, CMU Sphinx, 2012.

[35] Praat: doing phonetics by computer (accessed Nov. 8, 2012). http://www.fon.hum.uva.nl/praat/.