INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

Experiences with Shared Resources for Research and Education in Speech and Language Processing

Rebecca Bates[1], Eric Fosler-Lussier[2], Florian Metze[3], Martha Larson[4,5], Gina-Anne Levow[6], Emily Mower Provost[7]

[1]Minnesota State University, Mankato, USA
[2]The Ohio State University, USA
[3]Carnegie Mellon University, USA
[4]Delft University of Technology, The Netherlands
[5]Radboud University Nijmegen, The Netherlands
[6]University of Washington, USA
[7]University of Michigan, USA

bates@mnsu.edu, fosler@cse.ohio-state.edu, fmetze@cs.cmu.edu, m.a.larson@tudelft.nl, levow@uw.edu, emilykmp@umich.edu

Abstract

Resource barriers can prevent capable researchers from participating in the speech and language community and can make it difficult to support learning and participation in our field at a wide variety of institutions. Sharing resources, whether software, processed data, experimental methodologies or virtual machines, can reduce the barrier to entry and potentially broaden participation in speech and language research and improve workforce development. As an introduction to the special session on Sharing Research and Education Resources for Understanding Speech Processing, we outline current trends and requirements for expanding participation in speech processing research. A qualitative research approach was used. Faculty at a variety of institutions have been interviewed and have participated in reflection writing about needs, tools, challenges, and successes. Themes from reflections were generated using a grounded theory approach and were used to code interviews for related evidence. This paper describes the educational and research challenges experienced by faculty as users of resources, rather than the details of specific resources provided. The goal is to engage in a stronger dialog between users and providers so that needs and resources are better aligned. A case study of a shared resource used at several universities highlights this dialog.

Index Terms: speech recognition, education resources, research resources

1. Introduction

Speech processing systems have become increasingly complex and difficult to share across sites. Significant time is spent reimplementing published methods; even when software is shared, the lack of common environments between sites means that reproducing results can require significant effort. Open software repositories, virtual machines, and tools for automatically building container environments in the cloud are beginning to facilitate cross-site collaboration.

A desire to increase awareness and discussion about the need for and use of shared resources and the quality of how they are shared led the first authors to host a special session at Interspeech 2016 on Sharing Research and Education Resources for Understanding Speech Processing. In developing the call, we realized that discussion is typically driven by resource providers and about available tools. We therefore also put out a call for reflective essays on the needs for shared resources by those likely to be consumers of shared resources, particularly academics who are active both in research and education. This paper draws on three reflective essays (by Larson, Levow, and Mower Provost) to highlight challenges and successes in using shared resources for speech and language research and education.

The goal of this paper is to define resource barriers and address the educational and research challenges experienced by multiple users of resources, as well as to expand the conversation between users and producers. After a description of the methodological framework of our approach to data gathering, we present a range of user perspectives and a case study of a shared resource used at multiple sites.

2. Defining the Resource Barrier

Resource constraints impact what kinds of speech and language processing work can be conducted at different sites. Smaller groups who would like to conduct high quality speech recognition research, or groups that would like to build on state-of-the-art systems for other purposes (such as building a dialog system) may need to rely on externally developed systems that can be resource intensive. We conducted a survey of system resources needed to conduct publishable research by sampling papers from acoustic modeling, language modeling, and spoken language understanding sessions of Interspeech 2015 [1]. This provided insight into resource constraints and served as a proxy for educational needs since research tools are often repurposed for academic use. Acoustic modeling papers were quite resource intensive: of the 20 papers selected from 2015 Interspeech acoustic modeling sessions, only three used less than 100 hours of speech, and almost all of them used DNNs or some other neural network model. Interestingly, language modeling papers were very similar in acoustic data requirements: of the 20 papers sampled, only 5 used less than 100 hours of speech data (and typically used much more text data). Again neural network approaches abounded.

Interestingly, the papers on spoken dialogue systems tend to be a bit more open in terms of shared resources. Of the six papers in the spoken dialogue systems session, two of them used external systems (CMU Let's Go [2] and Microsoft's Speech SDK [3]), while three of them used ASR systems that were internal to the developing site. The remaining dialogue paper only evaluated simulated dialogue strategies and did not use a speech recognition front end.

http://dx.doi.org/10.21437/Interspeech.2016-1223

Table 1: Faculty Interview Questions

1) What course have you taught ASR or related issues in? Is this the course you will use the Virtual Machines in?
2) What are your learning goals related to speech recognition?
3) What have you used before? (HTK, internal systems, etc)
4) How much time did you spend preparing for class?
5) What problems did you have as a teacher?
6) What problems did you see your students having?
7) What successes did you observe?
8) If there were mixed responses/successes for your students, what helped the successful students?
9) What do you think will help your students learn?
10) What do you think will help you support student learning?
11) What do you hope to gain from using the Virtual Machines?

Table 2: Student Interview Questions

1) What was it that you built using the virtual kitchen toolkit?
2) Describe what you did while building the recognizer [or other item]. How did you do this?
3) Did you seek any external help? If so, who and for what topic?
4) What did you find particularly useful about the toolkit?
5) What did you find least useful about the toolkit?
6) Are there any specific features that would have worked better for you?
7) What have you learned from the experiments you have run so far - about speech recognition, or building speech systems, if anything?
8) Follow up: Can you describe a general process used in developing a speech recognizer?

Table 3: Writing Prompts

1) Describe your experiences in either using available resources or developing tools. As a user or developer, you could describe what the resources replaced in your context.
2) Please describe your biggest successes. These could relate to student learning, research output, or a variety of other things.
3) What were your biggest challenges? What advice would you have for someone beginning to use or develop the resources?
4) Within a classroom context, what were key student experiences, successes or challenges?
5) Within a research paradigm, how are you integrating found resources, such as available libraries, and how are they affecting your work?

This survey shows that the barrier to entry in this field in terms of data, processing, and memory resources is high. It highlighted that computational resources may be a challenge for smaller groups (e.g., ASR systems like Kaldi [4] require significant amounts of memory to train state-of-the-art systems; DNN-based acoustic models often require GPUs for efficient model training). The conversation about supporting research and education needs to consider how to support entry level research and educational projects that might benefit from a pre-built high-quality ASR system. Educationally, there are different needs in terms of teaching — both from the need to understand the basics of speech recognition, to being able to build large scale systems, to being able to use ASR as a subsystem in larger projects.

The good news is that there are much wider opportunities for using community resources than a few years ago. Speech recognition toolkits have had a long history as a shared resource in this community, notably HTK [5], CMU Sphinx [6], and more recently Kaldi [4]. These standalone toolkits were invaluable for those conducting speech recognition research, but were difficult to deploy for more casual users. Tools have improved over time — the design of neural network based models for ASR has been made much easier by the advent of a number of modeling toolkits [7, 8, 9, 10]. We also now have a wide array of web services for speech [11, 12, 13, 14] that can allow development of speech-recognition based projects using large-vocabulary recognizers in an internet setting, although these web services limit either students or researchers to roles of consumer, rather than designer.

With the proliferation of increasingly complex toolkits and resources, it can be bewildering for new entrants into the speech community to know how to get started. Many toolkits provide a "how to" tutorial that can be helpful. For example, Kaldi [4] provides a library of potential starter scripts. Another option, that also eases the need for matching system requirements, is to use middleware in the form of virtual machines; the Speech Recognition Virtual Kitchen (SRVK) [15] facilitates sharing experimental resources by encapsulating working systems in ready-to-run virtual machines.

Based on the current state of the field, the use of these resources provides challenges and opportunities for educators and researchers who work with students. In the next section, we present a qualitative study of people who have used or hope to use shared resources for improved learning.

3. Framework of This Study

Two primary approaches were used to gather information about past experiences and future needs. Qualitative interviews using a semi-structured approach were done to gather information about teaching or learning experience, learning goals, problems, and successes. Written responses to similar issues were also solicited through a special session call as well as personal invitations. Interviews with faculty were done as part of the planning and development process of the resource, prior to formal use of SRVK. Six interviews were done with people who had expressed interest in using SRVK, lasting from 30-90 minutes, with two participants from teaching-focused and four from research-focused institutions. Two students who used the SRVK in a formal class setting were interviewed as well. Interview questions are shown in Tables 1 and 2. Extensive field notes were taken during the interviews and interview recordings were transcribed. Writing prompts were provided to guide essay writing, along with requests for information about the institutional, course and/or project context. Authors could choose to address some or all of the prompts. Prompts are shown in Table 3. Three essays were included.

The written reflections were analysed qualitatively [16, 17, 18, 19]. A grounded theory approach was used to identify emerging themes in the written reflections. Themes were discussed amongst the authors. Interview transcripts and field notes were then coded for these themes by a single labeler. Evidence presented is in the form of themes and quotes from reflections and interviews that support the themes.

4. Themes & Perspectives

Common themes identified in the reflections include learning objectives that may go beyond automatic speech recognition systems and address a variety of levels of detail, motivation for learning these topics, instructional methods, barriers to entry for both research and learning, fragility of support for both students and teachers, and shared resources as structure for learn-

ing. Reflections were also organized by their authors to address concepts such as a variety of technical content, efficiency and effectiveness, motivation and momentum, and challenges.

4.1. Learning Objectives

There was a wide variety of learning objectives discussed, ranging from understanding the process of ASR as a system, to details about individual components of a dialog system, to HMMs and their implementation, to signal processing, to awareness of tools, and to potential applications. However, when connected with the experience of using available tools, one interviewee noted, "Installing sketchy software is not a learning objective."

Building on ASR techniques: HMMs are widely used, not just in speech recognition, but also, more broadly, in the behavior modeling community. However, the overhead associated with the installation and first application of this technology can be extremely daunting for students. One faculty member notes the beauty of the Speech Kitchen approach is its distillation of the critical concepts and technologies. Students are not required to install an instantiation of an HMM toolkit, instead, they download a virtual machine, on which the toolkit is already installed. There are no dependency issues or version controls for students to worry about.

Connecting to Dialog Systems: At least one faculty member focuses on teaching courses on spoken dialog systems. Students find dialog systems exciting and engaging, but the many complex subcomponents required pose a significant barrier to entry. Implementing such systems from scratch would be infeasible for the students and available resources are often limited by whether systems are in active development.

4.2. Instructional Methods

Multiple constraints affect how topics can be taught, such as 10 week quarters where there is limited time for learning tools, varying student background knowledge, and available software and computing resources. Even with these constraints, all faculty interviewed try to incorporate projects or hands-on learning experiences for their students. The use of projects has been shown to support technical learning outcomes and related outcomes that can produce better team members, researchers and project managers [20]. Using effective shared resources, with fewer distractions from unrelated issues like software dependencies and outdated components, results in successful technical experiences so that students meet technical learning outcomes as well as other professional and design skills.

Many classes start with a sequence of hands-on system experiences (labs), and then may move to projects. Projects range from teacher-defined to completely open ended. Curriculum design using labs with defined activities increases efficiency. As one faculty member describes: "The shared education resources that we used for our labs were the VM provided by the Speech Kitchen and also CMU Sphinx. These resources provided a basis for a lab in which students trained a system capable of recognizing a spoken phone number. The lab taught the students the functions of the components of an HMM speech recognizer, and the impact of design choices and training data on the recognizers' performance. Our goal was to develop a set of exercises that we could use in our course and in turn share with the Speech Kitchen community via the repository. It turned out that using shared resources not only made us more efficient, they also made us more effective. Teaching assistants spent less time on technical issues, and more time on questions related to speech and speech technology. In the end, we were amazed at the amount of ground we could cover, considering that speech processing was only one of the many topics covered by the course."

Whether working with pre-defined labs, open-ended projects, or undergraduate student research projects, presenting ASR or SLU systems as a black box or sequence of them works well (when it works), but there are trade-offs. Faculty noted that students eager to understand details often get frustrated by not knowing what's going on inside, and why software behaves like it does. Without access to the software, students stuck on a step may become confused. Going beyond black box systems to open source systems may have similar pitfalls. One professor noted "in developing a tutorial based on an open-source speech recognizer, I found that despite having had lectures on speech recognition technology, students would sometimes get confused as to what the different parts of the speech recognizer were doing (what does this file do?)."

Fortunately, the experience of a functioning system in a formal setting can be positive for students. One student had wanted to create a project on his own, but even though the tutorials were good, "it's very difficult to get it up and running by yourself, especially having the kind of infrastructure for all these different components to work together. It was really convenient having that from the get go." Noting later, "it was very nice to work with a system that acually works."

4.3. Fragility of support

Lack of documentation or help with installing and using tools affects both research and formal learning. There was frustration on both faculty and student ends when documentation is poor or insufficient. Students find that "there's not as many answers to questions online if you're looking at a specific tool kit." The frustration is compounded by the fact that students are used to finding answers by web search, as multiple people noted.

With standalone toolkits or research systems, software dependencies and lack of continuing support were significant barriers. Many existing systems rely on now out-dated software or operating system components. However, these frameworks allow students to directly engage and manipulate all components of the dialog system pipeline. In contrast, VoiceXML hosting platforms have proven to be more successful for in-class use in the longer term, and in the past few years, the vast majority of one faculty member's student projects have employed these resources. The platforms do not require the students to perform extensive system installation or maintenance, and the platform providers can maintain or update the systems over time and often offer basic technical support for even free accounts. Despite the ease of using cloud-based services, user experiences are at the whim of the providers, and access may shift over time.

4.4. Structures for Learning

Existing resources can affect what and how students are able to learn. As a teaching-focused faculty member stated, "There is no easy to run on a reliable platform speech recognition, even a demo version, out there, so it's hard to actually show students all the different parts in a demonstrative way. So I can talk to students about what parts would go into speech recognition, but then, they can't look at the language model or look at other things that are working or even see something coming out the other end. And so that's a problem. That's why I've never really done recognition as a hands on actual application—I talk my way through the theory." Hands on experiences come from other topics, like labeling, signal processing, and synthesis. An-

other faculty member discussed the tricky balance of getting the right tools at the right time for the particular audience, by "trying to provide students with the right level of background so the class isn't 50% background, but they're not completely lost."

Having the resources of lecture slides, homeworks or exercises and associated data, and tutorials, along with tools that allow for implementation of projects, supports student learning. Semester systems have the privilege of more time, but students still struggle to remain engaged when they are faced with software and hardware incompatibilities. In a shortened term, it can be even harder to get students through background learning and into project activity. "If there's an array of virtual machines that have systems trained to do different things, that helps me a lot. There's less overhead for setting up assignments." Not surprisingly, faculty felt they would benefit from the structure of functional tools and supporting materials.

4.5. Motivation

Multiple participants discussed the ways that experiences with shared resources could motivate engagement and further learning. Faculty members had a range of ideas about how these learning experiences would motivate students and expand participation. "I'm hoping this is a way into having people in minority language communities be able to build their own speech recognition systems." "People actually built systems they liked and wanted to show off to friends, like a pizza ordering system. They get into it and do creative tasks." "People learn on their own. They get new language and programming experience. Since they build a system, they get to talk about it in job interviews."

Interestingly, one faculty member shared that significant time investment in lab development was balanced with the inspiring realization that "our efforts would benefit not just our own students, but potentially other students internationally. The motivation helps to keep up the momentum for developing and maintaining labs." Forward momentum is necessary since the task of developing education resources is never done. Each year, exercises can be refined based on the experience with and feedback from the students of the year before. Speech processing technology develops at such a rapid pace that it is necessary to update the labs to keep them abreast with new developments.

5. SRVK: A Case Study

The Speech Recognition Virtual Kitchen (speechkitchen.org) was created to improve community research and education infrastructure for automatic speech recognition [15]. The resource includes state-of-the-art Linux-based virtual machines (VMs) with pre-compiled software tools to run various ASR and SLU experiments as well as a repository that supports sharing of VMs and experiments created by community members. Because the resource creators are also teachers, this served as a microcosm of the larger conversation about resources that this special session addresses.

Separate VMs, with tools and student assignments, were developed at both Ohio State and Carnegie Mellon. The Ohio State resources were shared with CMU for a class offered in fall of 2014. In this class, students worked on a variety of projects and two used a VM from the Speech Kitchen for theirs.

Even with access to the VMs, resources still limit students: "I have a fairly low power laptop so running the virtual machine for development was fairly difficult." But the student was strategic: "I would try and create or write all the code outside and then kind of plant it in and test it incrementally." Collaborating was also tricky for team projects. Students found that using tools like Vagrant [21] made sharing systems easier.

While some things, like using the Kaldi recognizer, were quite straightforward, training language models required using a second VM and had fewer instructions or documentation resources. Time was spent, or lost, simply trying to find files. A student noted that "We need to figure [it] out by ourselves, and we're also unable to google some examples. ... It seems like few people use [the software] so we probably need more popular [APIs]." The student recognized the need for a larger community using the same resources to have better tuned documentation and user experiences.

While system design was considered an important learning objective by faculty, one student dismissed this as "engineering stuff" and focused on describing his learning as what happened with "paper", i.e., theory learned in a classroom. However, the other student did identify this as learning, just more related to engineering: "I think I learned a lot about how a big software project like this works, more just in the engineering side. It's cool seeing all the pieces work together and seeing how it can be just packaged up and used over and over, like anywhere."

Participating faculty found the experience as a provider to be challenging, to have to change materials that had worked well before in the local classroom setting. On the other hand, it was pleasing to see the materials being used by others. As users, participating faculty found VMs to provide an excellent way to distribute tools in a well-defined environment which students can easily maneuver and understand, reducing faculty and TA support hours. A well thought-out, functioning experiment represents an easy way for students to grasp Hidden Markov Models, and not just read about them superficially. The collegial interactions between the researchers, in their "new" roles as user and provider allowed for an iterative process that supported user needs and the quality of the provider's system. In this case, shared resources provided the motivation of significant broader impact for the provider, as well as better learning experiences for students. Within the SRVK group, there has been an opportunity to broaden the conversation between users and providers over the course of the project.

6. Conclusions

The need for shared resources in a resource-intensive field continues to motivate community members. However, the scarce resource of time limits us all. Our experience has led us to conclude that the value of resource sharing to educational institutions lies not in saving time or person power, but rather in improvement in the quality of the student experience, and in the sense of connectedness to the larger, international speech processing community. A framework for sharing resources that can be honed and modified addresses the need to be current, while taking advantage of contributions over time, would clearly benefit the community. As a student said, "To create a high quality kind of system that you can just plug and play with– I think that was pretty incredible." Student responses like this suggest that functional tools and successful experiences can help broaden participation in the field, whether in research or academia.

7. Acknowledgments

This material is based on work supported by the National Science Foundation under grant nos. CNS-1305365, CNS-1305319, and CNS-1305215.

8. References

[1] *INTERSPEECH 2015 – 15th Annual Conference of the International Speech Communication Association, September 6-10, Dresden, Germany, Proceedings*, 2015.

[2] A. Raux, B. Langner, D. Bohus, A. W. Black, and M. Eskenazi, "Lets go public! taking a spoken dialog system to the real world," in *in Proc. of Interspeech 2005*. Citeseer, 2005.

[3] Microsoft Corporation, "Microsoft Speech Platform," https://msdn.microsoft.com/en-us/library/hh361572(v=office.14).aspx, 2016.

[4] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, 2011.

[5] S. Young, G. Evermann, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book*. Cambridge University Publishing Department, 2002. [Online]. Available: http://htk.eng.cam.ac.uk

[6] P. Lamere, P. Kwok, W. Walker, E. B. Gouvêa, R. Singh, B. Raj, and P. Wolf, "Design of the CMU Sphinx-4 decoder," in *INTERSPEECH*. Citeseer, 2003.

[7] R. Collobert, S. Bengio, and J. Mariéthoz, "Torch: a modular machine learning software library," IDIAP, Tech. Rep., 2002.

[8] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, and Y. Bengio, "Theano: new features and speed improvements," *arXiv preprint arXiv:1211.5590*, 2012.

[9] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems, 2015," *Software available from tensorflow. org*.

[10] Microsoft Corporation, "Computational Network Toolkit," http://www.cntk.ai, 2016.

[11] IBM Corporation, "IBM Watson speech to text," http://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/speech-to-text.html, 2016.

[12] Microsoft Corporation, "Project Oxford," https://www.projectoxford.ai, 2016.

[13] Google, "Google Cloud Speech API," https://cloud.google.com/speech/, 2016.

[14] Amazon.com, "Alexa Voice Services," https://developer.amazon.com/appsandservices/solutions/alexa/alexa-voice-service, 2016.

[15] F. Metze, E. Riebling, E. Fosler-Lussier, A. Plummer, and R. Bates, "The speech recognition virtual kitchen turns one," in *Proceedings of Interspeech*, vol. Show and Tell Session, 2015.

[16] B. L. Berg, *Qualitative research methods for the social sciences*. Boston, MA: Pearson, 2004.

[17] K. Charmaz, *Constructing grounded theory*. London: SAGE Publications, Ltd., 2014.

[18] J. Corbin and A. Strauss, *Basics of qualitative research: Techniques and procedures for developing grounded theory*. Thousand Oaks, CA: SAGE Publications, Inc., 2014.

[19] A. Strauss and J. Corbin, *Basics of qualitative research: Grounded theory procedures and techniques*. Newbury Park, CA: SAGE Publications, Ltd., 1990.

[20] A. Komos and E. de Graaff, "Problem-based and project-based learning in engineering edducation: Merging models," in *Cambridge Handbook of Engineering Education Research*, A. Johri and B. M. Olds, Eds. New York: Cambridge University Press, 2014, ch. 8, pp. 141–160.

[21] HashiCorp, "Vagrant," https://www.vagrantup.com, 2016.