ETS
Measuring the Power of Learning.™

Research Report
ETS RR–16-16

Bootstrapping Development of a Cloud-Based Spoken Dialog System in the Educational Domain From Scratch Using Crowdsourced Data

Vikram Ramanarayanan
David Suendermann-Oeft
Patrick Lange
Alexei V. Ivanov

Keelan Evanini
Zhou Yu
Eugene Tsuprun
Yao Qian

May 2016

ETS Research Report Series

Since its 1947 founding, ETS has conducted and disseminated scientific research to support its products and services, and to advance the measurement and education fields. In keeping with these goals, ETS is committed to making its research freely available to the professional community and to the general public. Published accounts of ETS research, including papers in the ETS Research Report series, undergo a formal peer-review process by ETS staff to ensure that they meet established scientific and professional standards. All such ETS-conducted peer reviews are in addition to any reviews that outside organizations may provide as part of their own publication processes. Peer review notwithstanding, the positions expressed in the ETS Research Report series and other published accounts of ETS research are those of the authors and not necessarily those of the Officers and Trustees of Educational Testing Service.

The Daniel Eignor Editorship is named in honor of Dr. Daniel R. Eignor, who from 2001 until 2011 served the Research and Development division as Editor for the ETS Research Report series. The Eignor Editorship has been created to recognize the pivotal leadership role that Dr. Eignor played in the research publication process at ETS.

ETS Research Report Series ISSN 2330-8516

RESEARCH REPORT

Bootstrapping Development of a Cloud-Based Spoken Dialog System in the Educational Domain From Scratch Using Crowdsourced Data

Vikram Ramanarayanan,[1] David Suendermann-Oeft,[1] Patrick Lange,[1] Alexei V. Ivanov,[1] Keelan Evanini,[2] Zhou Yu,[3] Eugene Tsuprun,[2] & Yao Qian[1]

1 Educational Testing Service, San Francisco, CA
2 Educational Testing Service, Princeton, NJ
3 Carnegie Mellon University, Pittsburgh, PA

We propose a crowdsourcing-based framework to iteratively and rapidly bootstrap a dialog system from scratch for a new domain. We leverage the open-source modular HALEF dialog system to deploy dialog applications. We illustrate the usefulness of this framework using four different prototype dialog items with applications in the educational domain and present initial results and insights from this endeavor.

Keywords Spoken dialog systems; crowdsourcing; computer-assisted language learning; automated assessment

doi:10.1002/ets2.12105

Spoken dialog systems (SDSs) consist of multiple subsystems, such as automatic speech recognizers (ASRs), spoken language understanding (SLU) modules, dialog managers (DMs), and spoken language generators, among others, interacting synergistically and often in real time. Each of these subsystems is complex and brings with it design challenges and open research questions in its own right. Rapidly bootstrapping a complete, working dialog system from scratch is therefore a challenge of considerable magnitude. Apart from the issues involved in training reasonably accurate models for ASR and SLU that work well in the domain of operation in real time, one has to ensure that the individual systems also work well in sequence such that the overall SDS performance does not suffer and guarantees an effective interaction with interlocutors who call into the system.

The ability to rapidly prototype and develop such SDSs is important for applications in the educational domain. For example, in automated conversational assessment, test developers might design several conversational items, each in a slightly different domain or subject area. One must, in such situations, be able to rapidly develop models and capabilities to ensure that the SDS can handle each of these diverse conversational applications gracefully. This is also true in the case of learning applications and so-called formative assessments: One must be able to quickly and accurately bootstrap SDSs that can respond to a wide variety of learner inputs across domains and contexts. Language learning and assessments add yet another complication in that systems need to deal gracefully with nonnative speech. Despite these challenges, the increasing demand for nonnative conversational learning and assessment applications makes this avenue of research an important one to pursue; however, this requires us to find a way to rapidly obtain data for model building and refinement in an iterative cycle.

Crowdsourcing is one solution that allows us to overcome this obstacle of obtaining data rapidly for iterative model building and refinement. Crowdsourcing has been used to rapidly and cheaply obtain data for a number of spoken language applications in recent years, such as native (Suendermann-Oeft, Liscombe, & Pieraccini, 2010) and nonnative (Evanini, Higgins, & Zechner, 2010) speech transcription and evaluation of quality of speech synthesizers (Buchholz & Latorre, 2011; Wolters, Isaac, & Renals, 2010). Crowdsourcing, and particularly Amazon Mechanical Turk, has also been used for assessing SDSs and for collecting interactions with SDSs. In particular, McGraw, Lee, Hetherington, Seneff, and Glass (2010) evaluated an SDS with a multimodal web interface using MIT's WAMI toolkit, collecting more than 1,000 dialog sessions. Rayner, Frank, Chua, Tsourakis, and Bouillon (2011) tested a computer-assisted language learning

Corresponding author: V. Ramanarayanan, E-mail: vramanarayanan@ets.org

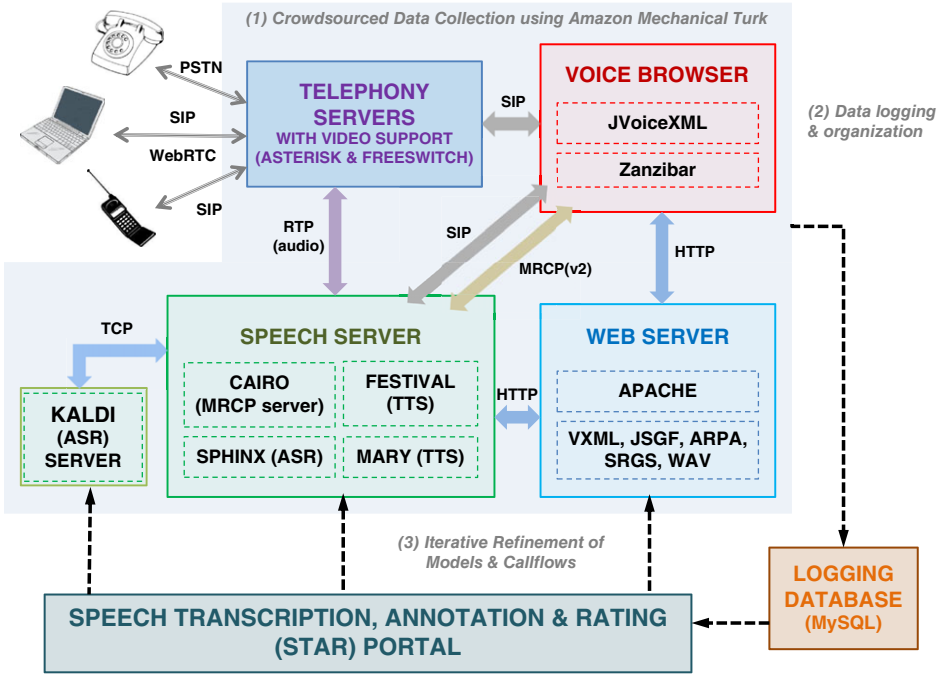

Figure 1 Proposed crowdsourcing-based iterative bootstrapping setup for rapid spoken dialog system development.

application with spoken input with altogether 129 interactions. Jurccek et al. (2011) deployed a phone-based SDS for the restaurant information domain, collecting 923 calls. However, to our knowledge, crowdsourcing has not been applied to the *iterative* development of a SDS (and its components), particularly in the educational domain, previously.

Therefore the goal of this report is to propose an iterative framework wherein a spoken (and potentially multimodal) dialog system can be architected from very generic models to more domain-specific models in a continuous development cycle and to present the initial results of an ongoing successful deployment of such a bootstrapped dialog system that is tailored to educational domain applications.

The HALEF Dialog Ecosystem

We use the open-source HALEF dialog system[1] to develop conversational applications within the crowdsourcing framework. Please see Figure 1 for a schematic overview of this framework. Because the HALEF architecture and components have been described in detail in prior publications (Ramanarayanan, Suendermann-Oeft, Ivanov, & Evanini, 2015; Suendermann-Oeft, Ramanarayanan, Teckenbrock, Neutatz, & Schmidt, 2015), we only briefly mention the various modules of the system here:

- Telephony servers Asterisk (van Meggelen, Smith, & Madsen, 2009) and FreeSWITCH (Minessale, Schreiber, Collins, & Chandler, 2012), which are compatible with Session Initiation Protocol (SIP), Public Switched Telephone Network (PSTN), and web Real-Time Communications (WebRTC) standards and include support for voice and video
- A voice browser, JVoiceXML (Schnelle-Walka, Radomski, & Mühlhäuser, 2013), which is compatible with VoiceXML 2.1 and can process SIP traffic and which incorporates support for multiple grammar standards, such as Java Speech Grammar Format (JSGF), Advanced Research Projects Agency (ARPA), and Weighted Finited State Transducer (WFST)
- An Media Resource Control Protocol (MRCP) speech server (Prylipko, Schnelle-Walka, Lord, & Wendemuth, 2011), Cairo, which allows the voice browser to initiate SIP or Real-Time Transport Protocol (RTP) connections from/to the telephony server and incorporates two speech recognizers (Sphinx and Kaldi; see respectively Lamere et al., 2003; Povey et al., 2011) and synthesizers (Mary and Festival; see respectively Schröder & Trouvain, 2003; Taylor, Black, & Caley, 1998).

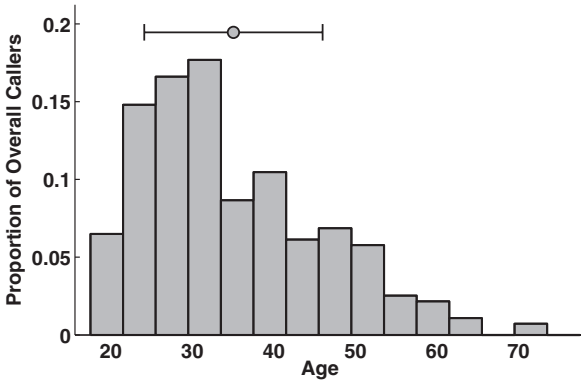

Figure 2 Average age of Turkers who called in to the spoken dialog system using Amazon Mechanical Turk. The bar on top represents the mean ± 1 standard deviation.

- An Apache Tomcat-based web server,[2] which can host dynamic VoiceXML pages, web services, and media libraries containing grammars and audio files
- OpenVXML,[3] a VoiceXML-based voice application authoring suite: generates dynamic web applications that can be housed on the web server
- A MySQL[4] database server for storing call logs
- A speech transcription, annotation, and rating portal that allows one to listen to and transcribe full-call recordings, rate them on a variety of dimensions such as caller experience and latency, and perform various semantic annotation tasks required to train ASR and SLU modules

Because we are bootstrapping a dialog system from scratch, we used generic models for ASR (trained on the *Wall Street Journal* corpus) and rule-based models (defined as part of the dialog flow in VXML, using the OpenVXML software) for SLU and DM. Although we plan to use the data flowing in continuously to iteratively refine statistical models for ASR, SLU, and DM in the future, for the purposes of this report, we focus only on the initial models.

Crowdsourcing Data Collection

We used Amazon Mechanical Turk for our crowdsourcing data collection experiments. Each spoken dialog task was its own individual human intelligence task (called a HIT on Amazon Mechanical Turk). In addition to reading instructions and calling in to the system, users were requested to fill out a 2- to 3-minute survey regarding the interaction. There were no particular restrictions on who could do the spoken dialog task, as we did not want to constrain the pool of people calling in to the system initially. As we continue to develop better models, we plan to restrict this pool of speakers to nonnative speakers of English. For the initial study that we report here, however, participants were mostly native speakers of American English (there were only 14 nonnative speakers) hailing from all over the continental United States; 43% were male, whereas 57% were female, and participants were well distributed across age groups (see Figure 2). In all, we collected 676 production calls over approximately 1 month of data collection, amounting to approximately 23 hours of speech.

Spoken Dialog Tasks

We deployed four conversational tasks for the purposes of this experiment: two tasks that tested pragmatic appropriateness of responses spoken during common scenarios in the workplace, a job interview, and a pizza-order-taking scenario.

One workplace pragmatics item involved interactive schedule negotiation, requiring several exchanges of information. The caller's task was to study and comprehend a weekly meeting schedule (provided as stimulus material) and then respond appropriately to an automated coworker who was trying to schedule a lunch meeting with the caller. The caller then dialed in to the system and proceeded to answer the sequence of questions posed by the automated coworker. Depending on the semantic class of the caller's answer to each question (as determined by the output of the speech recognizer and the natural language understanding module), he or she was redirected to the appropriate branch of the dialog tree, and the

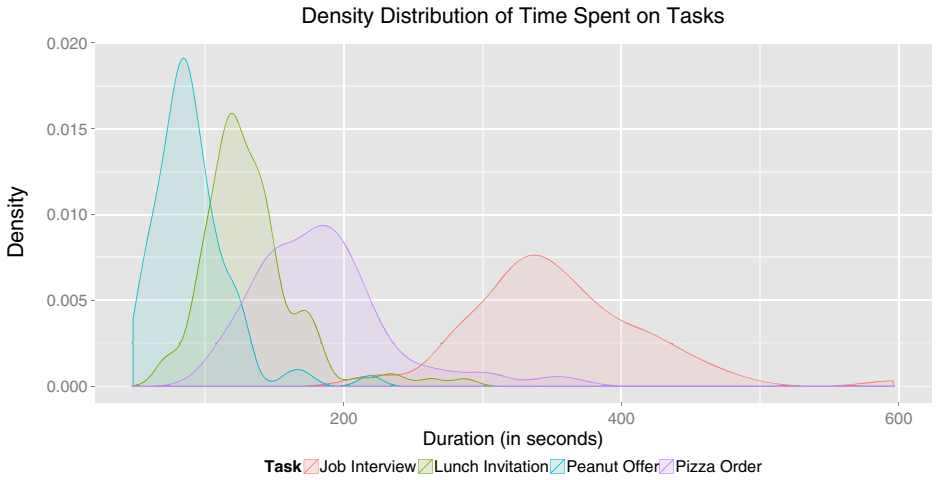

Figure 3 Distributions of call handling times (call durations) for each of the different items deployed.

conversation continued until all such questions were answered. This item was designed to measure the caller's ability to (a) understand the visual and oral stimuli (a work schedule and the automated coworker's questions and responses) and (b) politely and appropriately accept and decline invitations.

Another similar task involved testing how pragmatically appropriate callers' responses were in accepting or declining an offer of food in the workplace. Yet another task provided callers/test takers with a sample résumé stimulus and acted as a job seeker in an interview with an automated interviewer. Please see Ramanarayanan et al. (2015) for more detailed call-flow schematics corresponding to these tasks.

Whereas the three aforementioned tasks were system-initiated dialog scenarios, the fourth involved user-driven dialog. In this task, callers were required to act as customer service representatives at a pizza restaurant and to take an order from an automated customer who wanted to order a pizza. In the scenario, the automated customer waited for the user to ask a question ("what is your name?" "what toppings would you like on your pizza?" etc.) before replying with the appropriate response. Therefore this task might have been harder than the other three, imposing more cognitive load on the user.

Qualitative and Quantitative Performance Analysis

Figure 3 and Table 1 depict the distributions of call durations (or call handling times) and call completion rates, respectively, for each of the four items deployed. The pragmatics items were much shorter than the caller-initiated pizza item or the interview item, but as might be expected, they had higher completion rates. This was possibly because (a) there were more dialog states in the latter two items as compared to the first two or (b) the relatively more open-ended nature of the interview questions elicited longer and more spontaneous responses from callers. The longer items with more dialog states were also more likely to run into system issues at this initial stage of deployment and therefore had lower call completion rates. However, as Figure 4 shows, completion rates for the longer items improved statistically significantly over time ($p \approx .01$). This graph in particular shows the usefulness and effectiveness of the iterative development framework, which allowed us to find and correct issues with the system (whether they were in the VXML call flows, system code, or models) and redeploy the system to obtain rapid feedback about the modifications made.

To better understand how the system performed when actual test takers call in, we asked all Turkers to rate various aspects of their interaction with the system on a 5-point scale ranging from 1 (*least satisfactory*) to 5 (*most satisfactory*). The results of this user evaluation are depicted in Figure 5. We further had expert reviewers listen to each of the full-call recordings, examine the call logs, and rate each call on a range of dimensions (Suendermann-Oeft, Liscombe, Pieraccini, & Evanini, 2010). Histograms of these ratings are also shown in Figure 5 and include the following:

- *Audio quality of system responses*. This metric measured, on a scale from 1 to 5, how clear the automated agent was. A poor audio quality would be marked by frequent dropping in and out of the automated agent's voice or by muffled or garbled audio.

Table 1 Completion Rate for Each of the Four Items Deployed

Item	No. dialog states	No. calls	Completion rate (%)
Pragmatics (food offer)	1	131	61.83
Pragmatics (scheduling)	3	166	66.87
Job interview	8	192	35.42
Pizza customer service	7	187	47.06

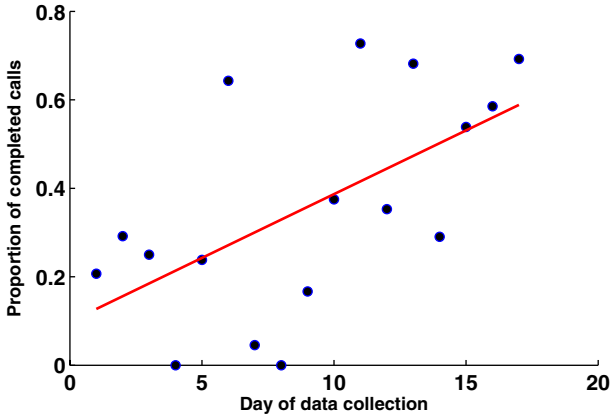

Figure 4 Completion rates over time for the two longer items deployed (interview and pizza), depicted by filled circles. The days are in chronological order but not necessarily consecutive. Note the increasing trend, depicted in red. The linear regression slope was significant at the 95% level ($p \approx .01$), and a left-sided Wilcoxon rank sum test showed that the completion rates after the ninth day were significantly higher than those on or before ($p \approx .001$).

- *Qualitative latency score*. A score measuring how debilitating the average delay is between the automated agent's response from the time the user finishes speaking to the conversation.
- *Caller experience*. A qualitative measure of the caller's experience using the automated agent, with 1 for a very bad experience and 5 for a very good experience.
- *Caller cooperation*. A qualitative measure of the caller's cooperation, or the caller's willingness to interact with the automated agent, with 1 for no cooperation and 5 for fully cooperative.

We observed that most users provided a high median rating to the extent they were able to complete their calls (4) as well as for the intelligibility of the system audio (5). Overall, users felt that the system performed well, with a median self-rated caller experience rating of 4. Experts tended to agree with user ratings in these cases, with similar median ratings for caller experience and audio quality. However, there was still plenty of scope for improvement with respect to how easy it was to understand the system prompts and how appropriate they were, with a median rating of 3. The median user rating of 3 ("satisfactory") for the system understanding category is not surprising, given that we are using unsophisticated rule-based grammars and natural language understanding. Another interesting observation is that users tended to find the system latency more debilitating on average than experts did (median rating of 3 vs. 4) on listening to full-call recordings. We will continue to investigate this going forward as more calls come in and system enhancements are made.

Conclusions and Outlook

We have presented a crowdsourcing framework that allows rapid prototyping and iterative development of dialog system components and models. Such a framework allows one to iteratively improve the system over time, as seen by the improvement of call completion rates over time in our case (Figure 4). This is because the framework enables developers to rapidly identify and resolve multiple issues in call flows, the source code of various components, and various modeling enhancements, such as the addition of semantic classes or the modification of synthesis voices. The exciting aspect of having a continuous influx of data and system feedback is that this opens up many avenues for ongoing and future research,

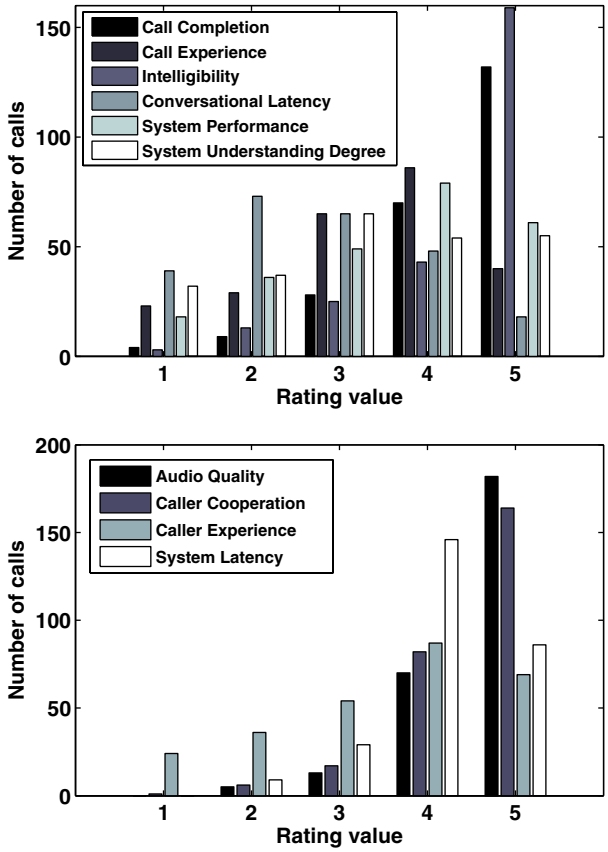

Figure 5 (top) User ratings. (bottom) Expert ratings.

including, but not limited to, better statistical models for ASR, SLU, and DM; iterative improvements to the item design; and parallelization and other code enhancements to improve system robustness and efficiency.

Acknowledgments

The authors would like to thank Lydia Rieck, Elizabeth Bredlau, Katie Vlasov, Juliet Marlier, Phallis Vaughter, Nehal Sadek, and Veronika Laughlin for their help in designing the conversational items. We would also like to thank Xinhao Wang, Ayana Stevenson, and Zydrune Mladineo for help with rating and transcribing calls and Robert Mundkowsky for help with system issues.

Notes

1 https://sourceforge.net/p/halef/
2 http://tomcat.apache.org/
3 https://github.com/OpenMethods/OpenVXML/
4 https://www.mysql.com/

References

Buchholz, S., & Latorre, J. (2011). Crowdsourcing preference tests, and how to detect cheating. In P. Cosi, R. De Mori, & G. Di (Eds.), *Proceedings of INTERSPEECH 2011: 12th annual conference of the International Speech Communication Association* (pp. 3053–3056). Baixas, France: International Speech Communication Association.

Evanini, K., Higgins, D., & Zechner, K. (2010). Using Amazon Mechanical Turk for transcription of non-native speech. In C. Callison-Burch & M. Dredze (Chairs), *Proceedings of the NAACL HLT 2010 workshop on creating speech and language data with Amazon's Mechanical Turk* (pp. 53–56). Stroudsburg, PA: Association for Computational Linguistics.

Jurccek, F., Keizer, S., Gašic, M., Mairesse, F., Thomson, B., Yu, K., & Young, S. (2011). Real user evaluation of spoken dialogue systems using Amazon Mechanical Turk. In P. Cosi, R. De Mori, & G. Di (Eds.), *Proceedings of INTERSPEECH 2011: 12th annual conference of the International Speech Communication Association* (pp. 3068 – 3071). Baixas, France: International Speech Communication Association.

Lamere, P., Kwok, P., Gouvea, E., Raj, B., Singh, R., Walker, W., … Wolf, P. (2003, April). *The CMU SPHINX-4 speech recognition system*. Paper presented at 2003 IEEE international conference on acoustics, speech, signal processing, Hong Kong, China.

McGraw, I., Lee, C.-Y., Hetherington, I. L., Seneff, S., & Glass, J. (2010). Collecting voices from the cloud. In N. Calzolari, K. Choukri, B. Maegaard, J. Mariani, J. Odijk, S. Piperidis, M. Rosner, & D. Tapias (Eds.), *LREC 2010: Seventh international conference on language resources and evaluation* (pp. 1576 – 1583). Paris, France: European Language Resources Association.

Minessale, A., Schreiber, D., Collins, M. S., & R. Chandler (2012). *FreeSWITCH cookbook*. Birmingham, England: Packt.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., … Vesely, K. (2011, December). *The Kaldi speech recognition toolkit.* Paper presented at the 2011 IEEE workshop on automatic speech recognition and understanding, Hilton Waikoloa Village, Hawaii.

Prylipko, D., Schnelle-Walka, D., Lord, S., & Wendemuth, A. (2011, September). *Zanzibar OpenIVR: An open-source framework for development of spoken dialog systems*. Paper presented at the text, speech, and dialogue (TSD) workshop, Pilsen, Czech Republic.

Ramanarayanan, V., Suendermann-Oeft, D., Ivanov, A., & Evanini, K. (2015, September). *A distributed cloud-based dialog system for conversational application development*. Paper presented at the 16th annual SIGdial meeting on discourse and dialogue, Prague, Czech Republic.

Rayner, E., Frank, I., Chua, C., Tsourakis, N., & Bouillon, P. (2011, August). *For a fistful of dollars: Using crowd-sourcing to evaluate a spoken language call application.* Paper presented at SLaTE 2011, Dorsoduro, Italy.

Schnelle-Walka, D., Radomski, S., & Mühlhäuser, M. (2013). JVoiceXML as a modality component in the W3C multimodal architecture. *Journal on Multimodal User Interfaces, 7*, 183 – 194.

Schröder, M., & Trouvain, J. (2003). The German text-to-speech synthesis system MARY: A tool for research, development and teaching. *International Journal of Speech Technology, 6*, 365 – 377.

Suendermann-Oeft, D., Liscombe, J., & Pieraccini, R. (2010). How to drink from a fire hose: One person can annoscribe 693 thousand utterances in one month. In *Proceedings of the 11th annual meeting of the special interest group on discourse and dialogue* (pp. 257 – 260). Stroudsburg, PA: Association for Computational Linguistics.

Suendermann-Oeft, D., Liscombe, J., Pieraccini, R., & Evanini, K. (2010). "How am I doing?": A new framework to effectively measure the performance of automated customer care contact centers. In A. Neustein (Ed.), *Advances in speech recognition: Mobile environments, call centers and clinics* (pp. 155 – 179). New York, NY: Springer.

Suendermann-Oeft, D., Ramanarayanan, V., Teckenbrock, M., Neutatz, F., & Schmidt, D. (2015, December). *HALEF: An open-source standard-compliant telephony-based modular spoken dialog system — A review and an outlook*. Paper presented at the IWSDS Workshop 2015, Busan, South Korea.

Taylor, P., Black, A., & Caley, R. (1998). The architecture of the Festival speech synthesis system. In G. Baily & C. Benoit (Eds.), *Proceedings of the third ESCA workshop on speech synthesis* (pp. 147 – 152). Retrieved from http://www.cstr.ed.ac.uk/downloads/publications/1998/Taylor_1998_d.pdf

van Meggelen, J., Smith, J., & Madsen, L. (2009). *Asterisk: The future of telephony*. Sebastopol, CA: O'Reilly Media.

Wolters, M. K., Isaac, K. B., & Renals, S. (2010). Evaluating speech synthesis intelligibility using Amazon Mechanical Turk. In Y. Sagisaka & K. Tokuda (Chairs), *SSW7: Proceedings of the 7th ICSA tutorial and research workshop on speech synthesis workshop* (pp. 136 – 141). Retrieved from http://isw3.naist.jp/ tomoki/ssw7/www/doc/ssw7_proceedings_rev.pdf

Suggested citation:

Ramanarayanan, V., Suendermann-Oeft, D., Lange, P., Ivanov, A. V., Evanini, K., Yu, Z., … Qian, Y. (2016). *Bootstrapping development of a cloud-based spoken dialog system in the educational domain from scratch using crowdsourced data* (Research Report No. RR-16-16). Princeton, NJ: Educational Testing Service. http://dx.doi.org/10.1002/ets2.12105

Action Editor: Beata Beigman Klebanov

Reviewers: Su-Youn Yoon and Klaus Zechner