



SPIRE-SST: An automatic web-based self-learning tool for syllable stress tutoring (SST) to the second language learners

Chiranjeevi Yarra^{1,a}, Anand P A², Kausthubha N K¹, Prasanta Kumar Ghosh^{1,b}
¹Electrical Engineering, Indian Institute of Science (IISc), Bangalore-560012, India
²National Institute of Technology Karnataka (NITK), Surathkal-575025, India
{^achiranjeeviy,^bprasantg}@iisc.ac.in

Abstract

Correct stress placement on the syllables in a word or word groups is important in the spoken communication. Thus, incorrect syllable stress, typically made by second language (L2) learners, could result in miscommunication. In this demo, we present SPIRE-SST tool that tutors to learn correct stress patterns in a self-learning manner. Thus, the proposed tool could also benefit the learners without any access to the effective training methods. For this, we design a front-end containing self-explanatory instructions that can be easily followed by the user. Using the front-end, learners can submit their audio to the back-end and can view the corresponding feedback from the back-end. In the back-end, we divide the entire audio from the learner into syllable segments and detect each syllable as stressed or unstressed. Using these stress markings, we compute a score representing the stress quality in comparison with the ground-truth stress markings and send it to the front-end as a feedback. We also send a set of three features by comparing the audio from the expert and learner as the feedback, which we assume to be useful for correcting the pronunciation errors.

1. Introduction

In the language learning, localized pronunciation errors, typically made by second language (L2) learners, could be minimized with correct usage of syllable stress patterns [1]. Often, incorrect stress patterns would result in miscommunication [2]. There have been many works proposed to detect the syllable stress automatically for the purpose of automatic language training [3]. However, we observe that the training tools developed from those algorithms are limited. For the benefit of L2 learning, we design SPIRE-SST¹ tool that trains the L2 learners with the correct usage of syllable stress in an automated way. The proposed tool automatically assesses the learner's stress patterns with respect to the ground-truth (referred to as experts) stress patterns and provides a feedback. Moreover, it has been shown that such online tools benefit the learners for whom effective training methods are not easily accessible [4]. In this demo, we present SPIRE-SST tool. To the best of our knowledge, there have been no similar online tools available.

2. Proposed architecture

The architecture of the proposed web-based tool is shown in Figure 1. It has two major components – front-end (user interface) and back-end (web-server). The front-end is available at the learner's location and the back-end is situated at our location. Both the front-end and the back-end communicate via



Figure 1: Architecture of the proposed web-based tool

Internet. The learner can access SPIRE-SST using electronic devices such as Desktop, laptop, mobiles, tablets etc. In addition, these devices are required to connect to a microphone for

recording the learner's voice. When a learner logs-in, the microphone is controlled by SPIRE-SST according to the learner's input until he/she logs out. We discuss more details of the front-end and the back-end in the following sub-sections.

2.1. Front-end

In order to train the learners, the proposed front-end provides following three main functionalities – 1) submission of the learner's voice 2) practice by listening to the expert 3) view the learner's performance/feedback. After learner logs-in, these three functionalities can be accessed by clicking the respective following three buttons – 1) Submit the recording 2) Listen to expert 3) Know your performance.

2.1.1. Submit the recording

Figure 2a shows an exemplary screen that appears for the first function in SPIRE-SST. On this screen, we provide a stimuli to read and four buttons to control the interface – 1) Submit the recording 2) Previous 3) Next 4) button with microphone symbol. With mouse click on the microphone symbol, learner can start recording his/her voice and at the same time the microphone symbol is replaced with a stop symbol as shown in the figure. With on click of the stop button, the recording is stopped and the stop symbol is replaced with the microphone symbol. After the recording is stopped, a play button appears below the stimuli as shown in the figure. On clicking the play button, the recorded voice can be listened, thus the learner can verify his/her recorded voice before submitting for the analysis. If he/she feels, the voice can be re-recorded and listened till his/her expected recording is achieved.

On click of 'Submit the recording', the most recent recorded voice is sent to the back-end and a score representing the stress quality is displayed after it is received from the back-end. The black dotted rectangular box in the figure encloses the window that displays the quality score. In the meantime, 'Submit the recording' button is replaced with the buttons, 'Listen to expert' and 'Know your performance'. Now, learner can choose to view the detailed feedback from functionalities associated with those two buttons or can move to previous/next stimuli by clicking 'Previous/Next' button.

2.1.2. Listen to expert

Figure 2b shows an exemplary screen that appears by clicking 'Listen to expert'. Using this, learner can correct syllable stress errors in his/her pronunciation by following pronunciation of the expert. On this screen, we provide a play button to listen the expert pronunciation and display the syllable transcriptions (arpabet format) in the expert's pronunciation with stress markings (stressed syllable is in bold). In addition, we show three bar-graphs containing syllable specific information. In the bar-graphs, we indicate values of the following three parameters – 1) syllable duration 2) average loudness in the syllable segment 3) peak loudness in the syllable segment. Each bar in the bar-graph indicates the parameter value corresponding to the expert's voice in each syllable. These parameters are typically used for computing the features in detecting the syllable stress and have been shown to influence the syllable stress significantly [1, 3]. Thus, we display these parameters during the

¹<https://spire.ee.iisc.ac.in/SPIRE-SST>

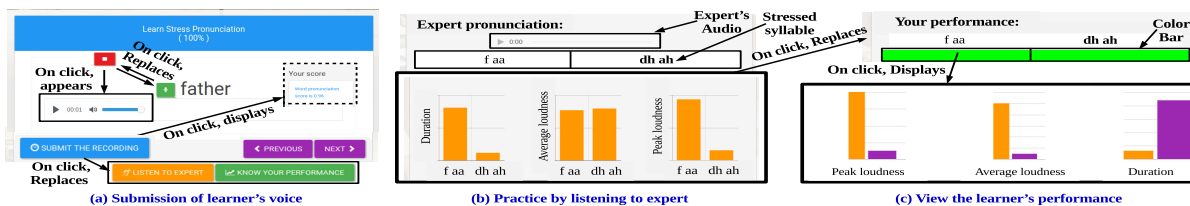


Figure 2: An illustration of the three main functionalities in SPIRE-SST

practice. This helps learners to adapt their pronunciation according to the parameter values to achieve the stress markings similar to that of the experts. Further, they can view the parameter values in their pronunciation using ‘Know your performance’ button.

2.1.3. Know your performance

Figure 2c illustrates the function associated with ‘Know your performance’ button. On click, the expert’s bar-graph (highlighted with thick black rectangular box) in Figure 2b replaces with the learner’s performance as shown in Figure 2c. In the learner’s performance, we display the syllables (transcriptions) in the learner’s pronunciation which are estimated using the force-alignment process. Among all the syllables, one syllable, which is estimated as stressed, is indicated in boldface. In addition, we also display a color bar with red and green color blocks, whose length equals the number of syllables in the expert’s pronunciation. In this color bar, each color block indicates the learner’s performance in each syllable. A red color indicates a mismatch between the stress parameters in the pronunciations of the learner and the expert. The green color indicates no mismatch. Further, on click of each color block, it shows three bar-graphs, which display the values of the three parameters in the respective syllable as shown in the figure. In each bar-graph, we show the parameter values in every syllable from the expert’s and the learner’s pronunciation. Using this information, we assume that the learners can identify the mismatches with the expert pronunciation and can train themselves to achieve the expert like pronunciation.

2.2. Back-end

Given a learner’s audio, in order to obtain the syllable transcriptions and its stress markings, we perform force-alignment on the audio using an automatic speech recognition (ASR) tool kit and estimate phoneme transcriptions and its boundaries. From these transcriptions, we obtain syllable transcriptions using automatic syllabification software and, thus, the syllable boundaries can be obtained. Following this, for each syllable, we estimate the stress markings as well as scores representing the confidence in estimating those markings. Further, these stress markings are color encoded in the color bar for each syllable. We consider the red color when the stress marking in the learner’s pronunciation is not matched with that in the expert’s pronunciation, otherwise, we use the green color.

2.2.1. Stress quality score computation:

Further, for the entire audio, we compute a score representing the stress quality. For this, we consider the scores belonging to every syllable obtained from the algorithm [1]. Let $S_E(i)$ and $S_L(i)$ are the scores corresponding to the expert and the learner for i -th syllable in a set of N syllables in the expert’s pronunciation. Here, we assume that N as well as syllable transcriptions are same in both the expert’s and learner’s pronunciation. In case of any mismatch, we indicate the learners about the incorrect pronunciation. Using the syllable level scores, we compute the score for the entire stimuli as $\frac{1}{N} \sum_{i=1}^N \left(1 - \frac{\min(S_E(i) - S_L(i), 1)}{S_E(i)}\right)$, such that 1 represents the highest quality and 0 for least quality.

3. Demonstration

In order to demonstrate SPIRE-SST, in the force-alignment process, we consider Kaldi speech recognition tool kit [5], P2TK syllabifier [6], and a lexicon containing pronunciations for each word. We use JavaScript and HTML for front-end coding and Node.js for back-end coding [7]. We set-up the server using LAMP (Linux, Apache, MySQL, PHP) stack on Ubuntu 14.04 LTS operating system. At the back-end, we obtain the stress markings and the confidence scores for both the learner and the expert by following the work proposed by Yarra et al. [1]. We implement their work using Python programming language. We consider a set of 204 stimuli taken from the training material used for spoken English training [2]. We divide the entire stimuli into four parts and are available in four lessons. We obtain the expert’s audio by recording the stimuli from a voice-over artist, proficient in British English spoken communication.

4. Conclusion

We present a web-based tool, named SPIRE-SST, that tutors the syllable stress to L2 learners. We design the front-end for SPIRE-SST with Javascript and HTML coding and the back-end with Node.js coding and Python programming language. Using this tool, we provide a feedback by showing the stress quality score of the learners with respect to experts as a assessment measure as well as the three parameters belonging to the expert’s and learner’s pronunciation as corrective measures. Further investigations are required to measure the effectiveness of the proposed tool as well as analyze sufficiency of the feedback parameters in the self-learning process.

5. Acknowledgement

We thank Pratiksha Trust for their support and Chetak R M of Maiyas Enterprises for his support in developing SPIRE-SST.

6. References

- [1] C. Yarra, O. D. Deshmukh, and P. K. Ghosh, “Automatic detection of syllable stress using sonority based prominence features for pronunciation evaluation,” *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5845–5849, 2017.
- [2] J. D. O’Connor, *Better English Pronunciation*. Cambridge University Press, 1980.
- [3] L. Ferrer, H. Bratt, C. Richey, H. Franco, V. Abrash, and K. Precoda, “Classification of lexical stress using spectral and prosodic features for computer-assisted language learning systems,” *Speech Communication*, vol. 69, pp. 31–45, 2015.
- [4] M. Kam, A. Kumar, S. Jain, A. Mathur, and J. Canny, “Improving literacy in rural india: Cellphone games in an after-school program,” *International Conference on Information and Communication Technologies and Development (ICTD)*, pp. 139–149, 2009.
- [5] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz et al., “The kaldi speech recognition toolkit,” *IEEE workshop on automatic speech recognition and understanding (ASRU)*, 2011.
- [6] J. Tauberer, “P2TK automated syllabifier,” Available at <https://sourceforge.net/p/p2tk/code/HEAD/tree/python/syllabify/>, last accessed on 10-03-2018.
- [7] M. Cantelon, M. Harter, T. Holowaychuk, and N. Rajlich, *Node.js in Action*. Manning Publications, 2017.