Pronunciation Checking with ASR

User Documentation

Evan Nichols

MFF 2021/2022

What Is It?

This application allows the user to test their pronunciation of various English sentence prompts. It supports multiple concurrent users. Users record themselves, via the web page, and then wait for their pronunciation to be evaluated. The user's speech input is evaluated on the basis of phonemes, which are the smallest units of sound that can distinguish one word from another. After a short delay, they may pull up their score and review, or try again. It uses a neural network module built for automatic speech recognition to evaluate the speech. The module compares the users' spoken phonemes to those of the (converted) text prompt, and reports the differences. The application is built using Flask, a web framework for Python, with a web page run in JavaScript.

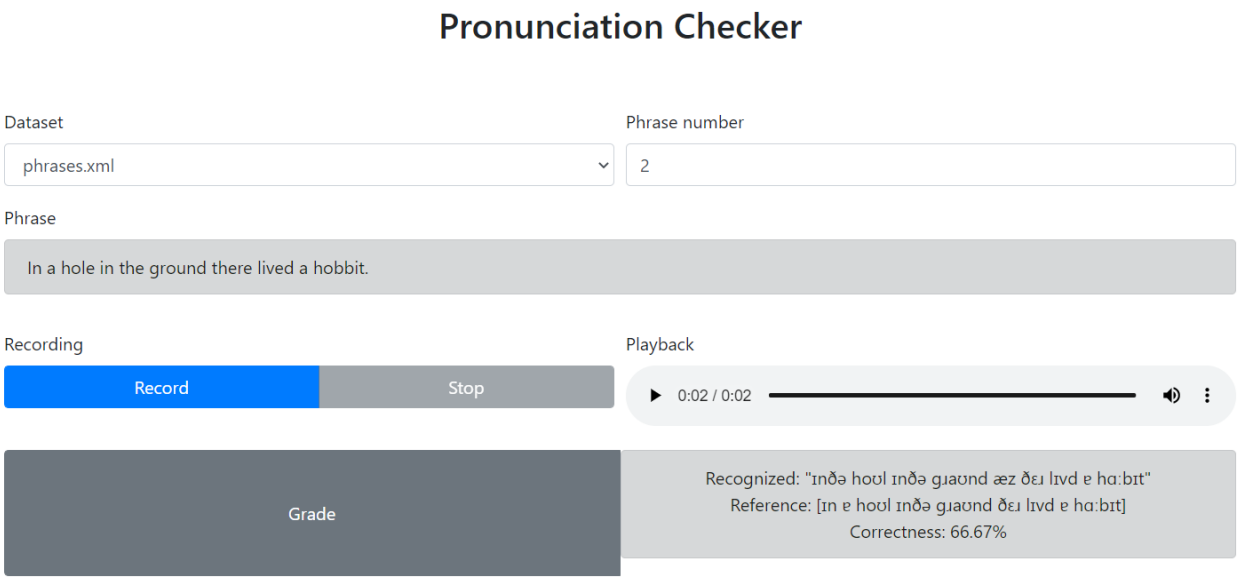

Figure 1: Web page after user has input speech and received a score.

How to Use this Appplication

Upon loading the web page, the user is presented with a simple interface of a few fields, and will be prompted by most browsers for microphone access. Immediately, the **Record** button may be pressed. This will begin recording, and the user should read the **Phrase** text out loud. After speaking the phrase and pressing **Stop**, the recorded audio is sent to the server-side along with the text prompt and processed and packaged into a data set. This data set is then processed by an automatic speech recognition module and the user's speech audio is graded against the prompt phrase. A score, marking how well the user pronounced the prompt, is stored in a text file and once when the user presses **Grade** this score is output to them. To add varying challenge, the user may choose from a number of phrases.

Phrase Prompts

The phrases can be any English words and sentences. The prompts do not NEED to be legitimate English, but for the sake of pronouncing words well it makes more sense for them to be actual language. Special characters are irrelevant in the grading, as the program strips items such as '!' and ':' before passing the prompt key to the ASR module. The phrases included in the app as of this writing are taken from various books and poems.

Input Audio

The user's speech speech is recorded, the raw data is sent to the back-end and then reformatted to a *.wav* file with a sampling rate of 16000 and 1 audio channel. A **spectrogram** is also created which displays some information to the user. The spectrogram measures frequency (y-axis) over time (x-axis), and the intensity of the color marks the amplitude of the frequency at that moment.

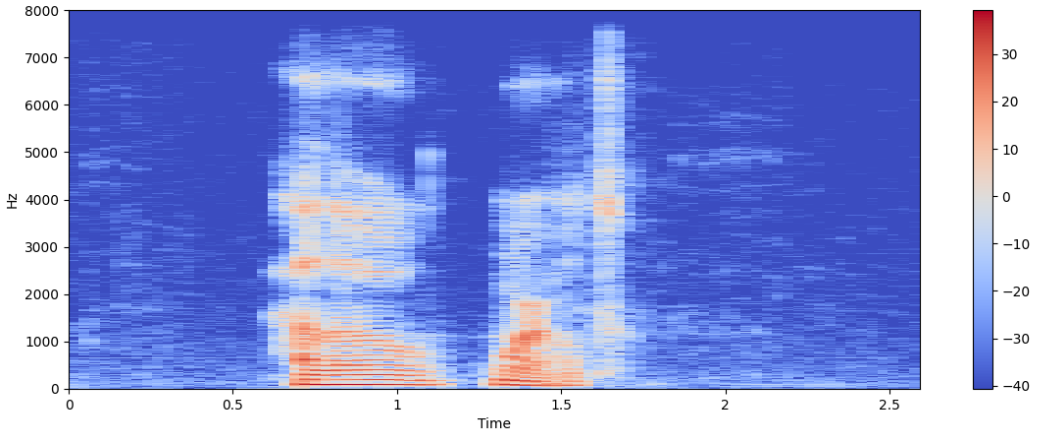

Figure 2: Spectrogram for speech "hello world".

User Interface

- **Dataset** is a drop-down list of the available phrase libraries. Each phrase is an item in an XML file with a single id, which denotes the items position in the file (1, 2 etc).
- **Phrase number** is a simple selector, pressing up or down pulls the corresponding phrase from the currently selected library. The selector loops around the dataset.
- **Phrase** is a text box that displays the currently selected phrase. It is this text the user is expected to read out loud. The phrase is sent along with the audio to the ASR module, but before that it is converted to its phonetic representation (phonemes) and stripped of any non-alphabetic characters, eg. '-', or '!'.
- **Record** is a button that begins recording, assuming the user has given microphone access to the browser when prompted. The user should read the text in **Phrase** out loud, and hit **Stop** when they are finished speaking. Hitting **Record** again (after **Stop**) will overwrite any previous input, allowing the user to adjust their submission.
- **Stop** closes the recording and, behind the scenes, sends the audio and prompt to the back-end and activates the **Grade** and **Playback** elements (deactivated by default).
- **Playback** allows the user to review their recorded speech.
- **Grade** pulls the results from '{UUID}_graded.txt'[1], and displays them to the right. The results include the recognized phonemes (what the user submitted), the correct phonemes (from the prompt), and a percentage score marking grading the user's pronunciation. Since the ASR module requires a variable amount of time to run, based on the size of the input data, a delay might occur between the user pressing **Stop** and **Grade** returning the appropriate results, in which case a message communicates this.

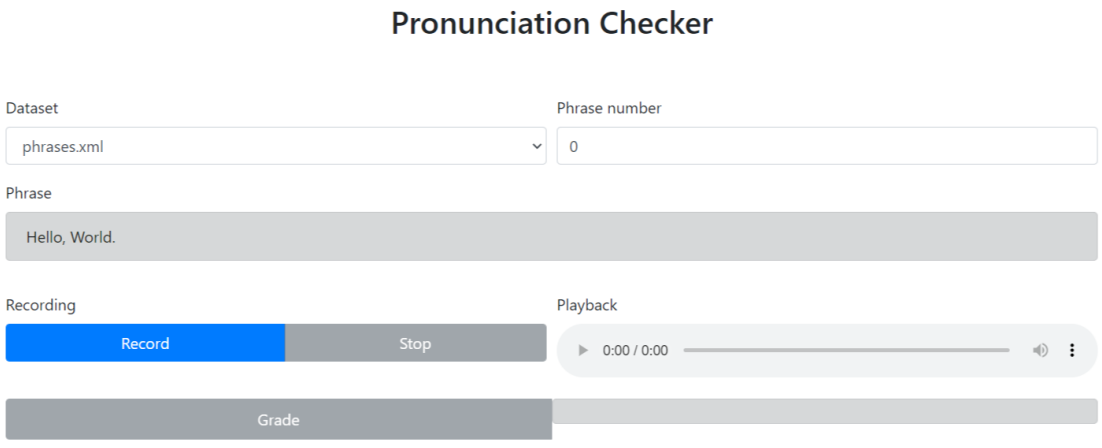

Figure 3: Application's web page.

[1]{UUID} indicates the universal unique identifier that is generated for each instance of the web page.