

Clarisse Taboy  
Methods II  
Term paper write-up  
June 3, 2021

### 1. Data used:

I used the Wikipron pronunciation dictionaries for French to write and test my g2p conversion program.

(URL: [https://github.com/sigmorphon/2021-task1/blob/main/data/medium/fre\\_dev.tsv](https://github.com/sigmorphon/2021-task1/blob/main/data/medium/fre_dev.tsv))

### 2. What I did:

To compose my g2p conversion rules for French, I consulted a PDF document that provided pronunciation charts for French vowels, consonants, and spellings with their IPA equivalents (URL: <https://mymission.lamission.edu/userdata/wentzj/docs/French%20IPA%20Charts.pdf>). I didn't implement all the rules listed, only some (randomly chosen) rules.

I developed unit tests for my rules in 'french\_g2p\_test.py' by choosing 20 example French words from the Wikipron data. The resulting number of g2p rules written in the program 'french\_g2p.py' was 39. I continued to edit my rules until all unit tests for the 20 words passed.

To test and evaluate the accuracy of the g2p converter, I wrote 'g2p\_eval.py' which applies the rules to all 1000 of the words in the Wikipron set. However, I realize after the fact that I think I made the mistake of picking example words for my unit tests from the Wikipron data itself, since the 20 example words would be included in the accuracy calculation. I also spent a long time trying to figure out if I could use sklearn to train and evaluate the program, but because I had a hard time with the previous assignment and fully understanding sklearn (such as which parts of the data would be in 'features' and 'labels'), I went with a very basic program to get the accuracy for now.

### 3. Evaluation results:

The program 'g2p\_eval.py' iterates over each French word in the tsv file of the Wikipron data ('fre\_dev.tsv') and checks to see if the output of the rules is equivalent to the expected phonemic transcription. Since the phonemic transcriptions in the tsv had spaces between each letter, I used `replace(" ", "")` so that the g2p program can apply the rules to it, and to make it easier to compare my output string to the expected string.

If the output didn't match the expected string, the pair was put into

a list called 'error\_list'. If the output did match the expected string, they were appended to 'correct\_list'. The accuracy was calculated by dividing the length of the correct list by the total number of words:  $172/1000 = 0.172$ .

4. Automating g2p conversion and its evaluation can help provide a pronunciation guide for novel words and indicate the trustworthiness (accuracy) of the generated phonemic transcription. I imagine that the higher the accuracy, the more useful these programs can be for applications like text to speech and speech recognition technologies. Having a personal interest in French, I could also benefit from having a g2p program on hand if I wanted to know how to pronounce words I don't know.

One thing I wish I could take into account (perhaps in the future) is pronunciations where there is a liaison between two French words, which affects the combined pronunciation of the first word's last letter and second word's first letter. In speech technologies, I feel like this would be useful too since without accounting for liaison, the synthesized speech might sound unnatural to fluent/native speakers.