VOWEL TUNER: A TOOL FOR FRENCH VOWELS PRONUNCIATION TRAINING

Soklong HIM
University of Lorraine
soklong.him6@etu.univ-lorraine.fr

Nora LINDVALL
University of Lorraine
nora.lindvall9@etu.univ-lorraine.fr

Maxime MÉLOUX
University of Lorraine
maxime.meloux4@etu.univ-lorraine.fr

Jorge VASQUEZ-MERCADO
University of Lorraine
jorge-luis.vasquez-mercado9@etu.univ-lorraine.fr

January 27, 2023

ABSTRACT

In this paper, we aim to create a tool that can help learners of French to improve their pronunciation of French vowels. This was done by creating an application that allows users to record vowels. A classifier then determines whether the vowel is pronounced correctly or not. If the pronunciation is incorrect, the user is provided with personalized feedback. In order to find a good classifier, we implemented two approaches: a linguistic one, based on formant extraction, and a deep learning one, based on mel-spectrograms and using a convolutional neural network architecture. After initially testing both models on the All Vowels corpus, consisting of 5,755 vowels, we built a web application and tested it in real-life conditions. The linguistic model proved more robust to real-life recording conditions and achieved good performance in most cases.

1 Introduction

Becoming proficient in speaking a foreign language is one of the tougher skills to master in terms of language learning. The learner needs to pronounce the words with sufficient precision to make themselves understood. This must be done in real-time while simultaneously attempting to find the right words and correct grammatical structures. The effort can be taxing to many language learners [1].

A common phenomenon that prevents learners from producing their target language with good pronunciation is phonetic substitution. Based on language transfer theory, phonetic substitutions occur due to superimposition of the phonetic inventory of first language (L1) onto a second language (L2) [2]. In other words, learners are often inclined to use the set of phonemes from their native language when speaking a foreign language. In order to stop defaulting to the phoneme inventory of one's L1, the learner needs to be made aware of the differences between their L1 and their target language.

The aim of this project is to provide a tool to help language learners improve their pronunciation of French vowels. The tool is meant to provide useful feedback that can help guide the learner so that their vowels fall within the boundaries of the French vowel phonemes. Having a foreign accent is not a problem in itself as long as the interlocutor can correctly identify the uttered phoneme. Pronouncing the wrong phoneme can lead to miscommunication as it changes the meaning of the word. For example, one can compare the French words "dessus" /dəsy/ ("above") and "dessous" /dəsu/ ("beneath") which only differ in terms of a single phoneme, however the meaning changes completely.

In the field of Natural Language Processing, phoneme recognition is an arduous and as of yet unsolved task. While in the past, linguistic-based methods were commonplace [3], they have recently been replaced by neural models. In particular, Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) have obtained high accuracy

on relatively little training data [4] [5] [6] [7]. These neural networks typically take as input the mel-spectrogram of the recording [8], which is a representation of the energy distribution of the audio in the frequency space.

1.1 French vowels

French has a large vowel phoneme inventory with around 16 vowel phonemes: i, e, ɛ, y, ø, œ, ə, u, o, ɔ, a, ɑ, ɛ̃, ɔ̃, ɑ̃, œ̃, although the exact number varies depending on the linguistic landscape. However, as Parisian French has gradually become established as the standard French variety [9], this is the variety that will be covered in this paper.

In most varieties of Parisian French, /ɑ/ has merged with /a/. In other words, "pâte" and "patte" are both realized as /pat/. Moreover, most Parisians no longer distinguish between *"brin"* and "brun" in speech, as /œ̃/ merges to /ɛ̃/ [10]. Lastly, the lax vowel /ə/ has been a source of difficulty for linguists due to its unclear phonetic properties. In standard dialects, it is often realized as /ø/ when stressed, and as /œ/ when unstressed. For example "sur ce" is often pronounced /syʁsø/ instead of /syʁsə/, and "prenait" is pronounced /pʁœnɛ/ instead of /pʁənɛ/ [11]. For the aforementioned reasons /ɑ/, /œ̃/ and /ə/ have been excluded in our analysis, and the vowel phoneme inventory representing Parisian French can be seen in Figure 1. X-SAMPA was used to write vowel names for the majority of this project. The translation table between X-SAMPA and IPA symbols can be found in Appendix A.

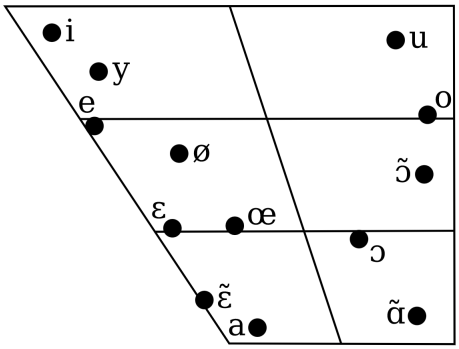

Figure 1: Vowels in Parisian French based on chart by [12]

2 Methodology

The methodology that the project followed was based on two different approaches: a linguistic approach, using findings from phonology, and a deep learning-based approach. Due to time constraints, only oral vowels were processed in the experiments.

2.1 Linguistic approach

The idea of the linguistic approach is to use formants as the main information to predict a vowel. Formants are accumulations of energy at a particular frequency in a speech wave, and each formant corresponds to a specific resonance in the vocal tract. Vowel formant frequencies are one of the most commonly used acoustic measures in speech processing applications [13]. In particular, the first formant (F1) is related to openness of the vowel, the resonance between the vocal chords and the oral cavity, while the second formant (F2) corresponds to frontness, reflecting the tongue position in the oral cavity. The first two formants are generally sufficient to identify a vowel. However, the third and fourth formants (F3 and F4) can provide useful additional information about lip rounding, nasality and paranasality [14].

Given an audio file containing the recording of a vowel, we extracted formants using the Python package `praat-parselmouth`[1], which provides a high-level interface for the Praat phonetics analysis software[2].

Since our goal is to provide feedback on recordings of entire words (and not a standalone vowel), one needs to consider the phonological surrounding of the vowel. In particular, we wish for the user of the application to say one monosyllabic word from a predetermined list, and to only predict the vowel of that word. We therefore decided to use the additional information of the phonemes preceding and following the vowel as part of the input to our classifiers.

[1] `https://parselmouth.readthedocs.io/en/stable/`

[2] `https://www.fon.hum.uva.nl/praat/`

We implemented two different approaches for vowel detection based on formants: one based on reference values, and a classifier-based one.

2.1.1 Reference approach

The first idea is to use a set of reference values for vowel formants in order to make prediction. Given a set $\{\overline{f_{1,v}}, \overline{f_{2,v}}, ..., \overline{f_{n,v}}\}$ of average formant values for each vowel v of the vowel inventory V and a candidate vowel c, we can obtain a prediction by taking $\hat{c} = \underset{v \in V}{\arg\min}\, d(v, c)$ where d is a suitable distance function. We implemented two such functions:

- Weighted Euclidean distance: $d_E(v, c, w_1, ..., w_n) = \left(\sum_{i=1}^{n} w_i(\overline{f_{i,v}} - f_{i,c})\right)^{1/2}$, where $(w_i)_{i \in [\![1..n]\!]}$. The standard Euclidean distance is a specific case of d_E where $w_1 = ... = w_n = 1$. The intuition behind the weighting is that since F1 and F2 should in theory be enough to determine most vowels, the partial distances associated with them should be weighted higher than for higher formants.
- Interval-aware Euclidean distance: In some cases, we found reference formants in which a standard deviation $\sigma_{i,v}$ was given along with the values of the formants. Taking the assumption $f_{i,v} \sim N(\overline{f_{i,v}}, \sigma_{i,v})$, we then defined a new distance $d_{EI}(v, c, w_1, ..., w_n) = \sum_{i=1}^{n} w_i(\overline{f_{i,v}} - f_{i,c})/\sigma_{i,v}$. The idea is to calculate how many standard deviations separate a candidate vowel from a reference one, while the intuition behind the weighting is as for the Euclidean distance.

The advantages of this reference approach are that it requires no training dataset if a suitable set of reference formants is found, and that it is very lightweight computation-weight. The disadvantages are that such a set is hard to find in practice, and that reference values should be found for every possible context (speaker gender, age, previous phoneme, next phoneme...) to ensure good accuracy.

2.1.2 Classifier approach

In a second approach, we consider the value of each formant of a vowel as an input feature, along with the gender of the speaker and the value of the preceding phonemes. We can then train different types of classifiers on a given dataset of vowels. For this approach, several types of classifiers were implemented:

- Decision Trees, an explainable model that makes use of information theory and divides the latent space recursively using binary decisions
- K neighbors, a simple and explainable model that considers the k neighbors closest to a given point in a latent space in order to predict its class
- Multinomial Logistic Regression, a generalization of logistic regression to multi-class problems
- Random Forests, an ensemble method that is a generalization of decision trees using randomization methods for initialization
- Multilayer Perceptron, a model based on a feedforward neural network
- Extra Trees, a specialized version of Random Forests that adds further randomization
- Bagging, an ensemble method that trains different copies of a classifier on parts of the initial dataset and aggregates their results
- Stacking, an ensemble methods that combines the output of multiple classifiers to form a final prediction

2.2 Neural approach

The objective of the deep learning approach is to have a neural network model capable of predicting the input vowel by taking into consideration the spectrogram representation of the input audio.

A convolutional neural network (CNN) architecture was used for this task. For every input vowel, the audio was turned into its mel-spectrogram, which was then normalized and turned into an image of height 256 and of variable width. The image width was chosen as the maximum width in the training set of each dataset, and other images were re-scaled (through horizontal stretching or contraction) to match that width. The neural network then outputs a probability distribution over our set of classes (one class per vowel), and the highest probability is chosen as the model's prediction.

2.3 Vowel extraction

In general, we found our models to perform better at predicting the vowel in a word when fed only with the audio of that vowel (rather than the entire word). Since the goal of our application is for learners to practice in context using full words, we decided to implement a way to automatically extract the vowel segment from the audio recording of a word. This was done in two steps:

- First, the leading and trailing silences are removed through the use of a simple sound level detection function, based on a hard-coded silence threshold.
- To predict the vowel boundaries given a sound file, we then used a regression model. Values between 0 and 1 were attributed to the start and end point of the vowel, where 0 indicates the beginning of the file, and 1 indicates the end of the file. We also used a CNN network architecture for this task, in which the model was fed the mel-spectrogram of the entire recording. The use of a neural model here was motivated by the fact that while it may be hard to detect a vowel from a spectrum, consonant-vowel transitions are usually easily detectable, as they translate to abrupt changes in the spectrogram.

3 Preliminary experiments

3.1 Datasets

Three datasets were used for our initial experiments:

- An informal corpus, recorded in real-life condition, made of the recordings of 8 students (6 male and 2 female), of which 5 native French speakers and 3 non-native learners coming from diverse countries and having an advanced level of French. Each speaker was recorded pronouncing list of words found in the table below, which contains one example of each of the typical oral vowels for a Parisian French speaker, including vowels that may or may not be distinguished depending on the speaker. Each file was segmented and each vowel annotated with the vowel perceived by a native French speaker.

Word	Pronunciation	Translation
si	/si/	'if'
fée	/fe/	'fairy'
fait	/fɛ/, /fe/	'does'
fêt	/fɛt/, /fɛt/	'party'
su	/sy/	'known'
ceux	/sø/	'those'
sœur	/sœʁ/	'sister'
ce	/sø/, /sœ/	'this'/'that'
sous	/su/	'under'
sot	/so/	'silly'
sort	/sɔʁ/	'fate'
sa	/sa/	'his'/'her'

- A subset of the InterFra corpus[3], a Swedish corpus of French native and non-native speakers with various ages and levels of proficiency. We selected 2 native and 2 non-native speaker recordings (one male and one female for each), and manually annotated the first 50 vowels or 30 seconds of each file with the vowel perceived by a native French speaker, along with the left and right phonemic context. This resulted in a final corpus of 225 vowels.
- The All Vowels corpus, a corpus developed at LORIA[4] and containing 67 recordings of native French speakers (34 female and 33 male speakers) pronouncing a list of 84 monosyllabic French words, all of the form CV(R), where C is one of /l/, /m/, /p/, /s/ or /t/ and (R) is /ʁ/, which is only present when V is one of {ɛ, œ, ɔ}. The full list of words can be found in Appendix B. While the corpus was already partially pre-annotated with word- and phoneme-level annotations when we obtained it, we found several errors which led us to re-annotate it entirely through the use of the same forced alignment tool that had been used to annotate it, Astali[5], while

[3] https://spraakbanken.gu.se/en/resources/interfra

[4] https://www.loria.fr/

[5] https://astali.loria.fr/

taking into account individual pronunciation differences from some vowels. This produced a final corpus of 5,755 vowels. Figure 2 shows the distribution of those vowels with respect to their first two formants. It can be seen that different vowels have distinct ranges of formant values.

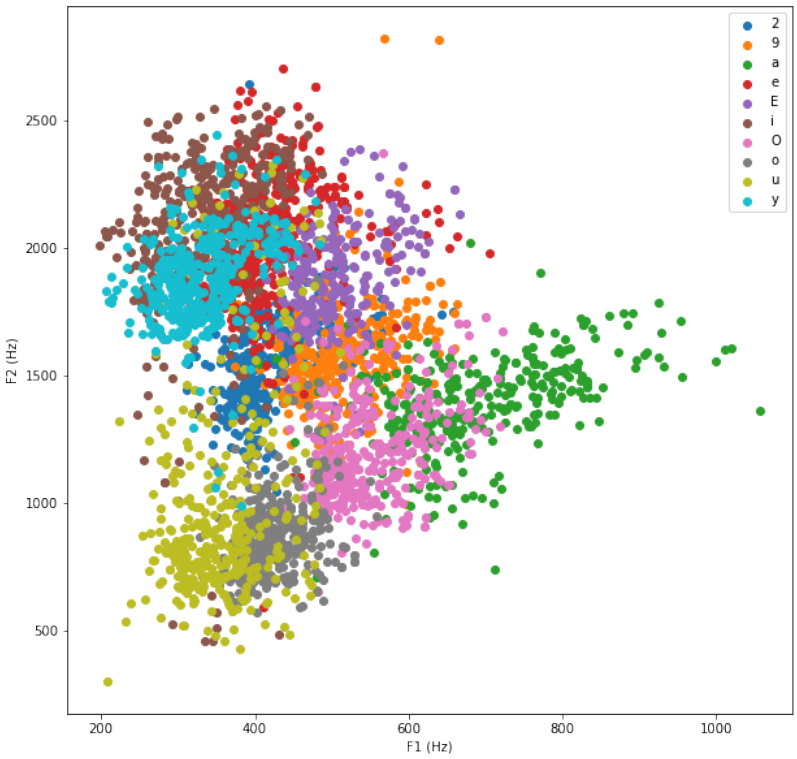

Figure 2: The distribution of oral vowels in the All Vowel dataset in the (F1, F2) formant space.

Due to chronological constraints, the first two corpora were only used to test the reference formants approach, after which further experiments were performed on the larger All Vowels corpus.

3.2 Parameters and results

3.2.1 Linguistic approach

We kept the standard Praat parameters for formant extraction, using a window length of 0.25 seconds, a formant ceiling of 5,000 Hz and a pre-emphasis filter for frequencies starting at 50 Hz. For each vowel, we extracted the first four formants (F1, F2, F3 and F4).

3.2.1.1 Reference approach

Two sets of reference formants were used in our experiments: one containing only mean values for formants [3] and one containing standard deviations [15]. The latter also contains variations of the formants based on the previous phoneme, but this data was not used in our project. An article containing a lot of information about reference formants for French oral vowels was found much later, and was therefore also not used here [16].

For the reference formants containing only average values, we obtained the best results using a weighted Euclidean distance using $w_1 = 1, w_2 = 1, w_3 = 0.5, w_4 = 0.2$. The resulting model's accuracy with respect to various speaker groups is described in Tables 1 and 2.

Subset	2 formants	3 formants	4 formants
Native speakers	0.120	0.133	**0.157**
Non-native speakers	0.170	**0.205**	0.114
Female speakers	0.178	**0.208**	0.168
Male speakers	0.100	**0.114**	0.086
Overall	0.146	**0.170**	0.135

Table 1: Accuracy between the detected and perceived vowels in the InterFra sub-corpus.

Subset	2 formants	3 formants	4 formants
Native speakers	0.312	**0.359**	0.344
Non-native speakers	**0.359**	0.256	0.282
Female speakers	**0.346**	0.192	0.038
Male speakers	0.325	0.364	**0.416**
Overall	**0.330**	0.320	0.320

Table 2: Accuracy between the detected and perceived vowels in the informal corpus.

For the reference formants containing both average values and standard deviations, we used an interval-aware distance function using $w_1 = 1, w_2 = 1, w_3 = 0.5, w_4 = 0.2$. The approach was tested using either the first 2, 3 or 4 formants in the distance computation. The results of this approach are given in Tables 3 and 4.

Subset	2 formants	3 formants	4 formants
Native	0.193 (+0.073)	0.289 (+0.156)	**0.398** (+0.241)
Non-native	0.114 (-0.066)	0.170 (-0.035)	**0.273** (+0.159)
Female	0.168 (-0.010)	0.198 (-0.010)	**0.297** (+0.129)
Male	0.129 (+0.029)	0.271 (+0.127)	**0.386** (+0.300)
Overall	0.152 (+0.006)	0.228 (+0.058)	**0.333** (+0.198)

Table 3: Accuracy between the detected and perceived vowels in the InterFra sub-corpus. The value displayed between brackets corresponds to the delta from the accuracy obtained using the first set of reference formants.

Subset	2 formants	3 formants	4 formants
Native	0.312 ($\approx$ 0)	0.203 (-0.156)	0.281 (-0.063)
Non-native	0.487 (+0.128)	0.308 (+0.042)	0.410 (+0.128)
Female	0.462 (+0.136)	0.308 (+0.116)	0.308 (+0.270)
Male	0.351 (+0.026)	0.221 (-0.143)	0.338 (-0.078)
Overall	0.379 (+0.049)	0.243 (-0.077)	0.330 (+0.010)

Table 4: Accuracy between the detected and perceived vowels in the informal corpus. The value displayed between brackets corresponds to the delta from the accuracy obtained using the first set of reference formants.

3.2.1.2 Classifier approach

Classifiers were implemented using the `scikit-learn` library[6]. A variety of hyperparameters was tested for each classifiers, but the best accuracy was often obtained with the default values. The exceptions are as follows:

- K neighbors: `n_neighbors` was set to 1

[6] `https://scikit-learn.org/`

- Multilayer Perceptron: `hidden_layer_sizes` was set to $(40, 50)$ and `activation` to "tanh"
- Random Forests: `n_estimators` was set to 500
- Extra trees: `n_estimators` was set to 400
- Bagging classifier: `n_estimators` was set to 100
- Stacking classifier: The classifier used one of each of the previous classifiers (Decision trees, K neighbors, logistic regression, multilayer perceptron, extra trees, random forest, bagging and stacking)

The classifiers were fed with several input features: the values of the first four formants (normalized to have a zero mean and unit variance), the gender of the speaker and the one-hot encoded previous phoneme (one of /l/, /m/, /p/, /s/ or /t/). Since the phoneme following the vowel can only be /ʁ/, and it is only the case for some specific vowels, we did not include that as a feature for the classification task, as it could lead to errors when testing on non-native speakers.

The obtained results can be found in Figure 5, and Appendix C contains additional quantitative results for some of the classifiers.

Classifier	**Accuracy**
Decision trees	74.25%
K neighbors	79.40%
Logistic regression	77.51%
Multilayer perceptron	81.30%
Extra trees	**82.93%**
Random forest	**82.93%**
Bagging	81.03%
Stacking	80.22%

Table 5: Accuracy of various classifiers on the test set of the All Vowels corpus.

3.2.2 Neural classifier and vowel extraction

Experiments were performed with various hyperparameters to find the best neural architecture for both the neural classifier and the vowel extractor, the full list of which can be found in Appendix F. The best neural architecture was found to be the same for both models, except for the size of the output (10 classes for the classifier, and 2 output values for the vowel extractor) and input: the vowel extractor takes as input a full recording, while the classifier only takes the part of the recording that contains the vowel. Another key difference was the choice of the cross entropy loss for the classifier and binary cross entropy for the vowel extractor. The final architecture is described in Figure 3.

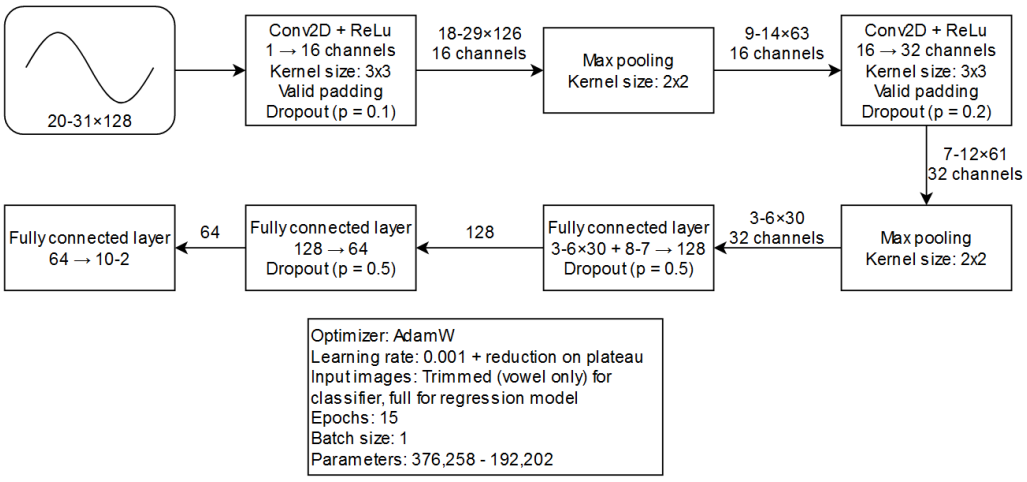

Figure 3: The best architecture for the neural classifier and vowel extractor. When two values are separated by a hyphen, the left value is the one used for the neural classifier, and the right one for the vowel extractor.

Both models were fed with the mel-spectrogram image of the recording as their primary input, using 128 mel bands. The one-hot encoded previous phoneme and gender of the speaker were fed as secondary input after the convolutional layers, as well as whether the following phoneme is /ʁ/ (only in the case of the vowel extractor).

3.3 Analysis

3.3.1 Reference formants

The reference formant method performed quite poorly on the dataset, which quickly led us to switch to the classifier and neural methods. Generally, obtained results where better on the informal corpus, but fail to achieve more than 37.9% accuracy in the best case, which is insufficient for any practical purposes (let alone teaching ones). We argue that this is due to several factors:

- As mentioned above, having one set of reference formants is not enough. Formants can very depending on the gender and age of the speaker, as well as the phonemic context. In particular, the first set of reference formants was explicitly given to be valid for male speakers. More generally, additional input features should be considered.
- All vowels in the datasets were annotated by a single native French speaker, which leads to high subjectivity in both the position of the vowel boundaries as well as the labeling of the vowel itself.
- The choice of the metric is also very subjective, and it is unclear whether it is the best way to compute vowel distance.
- The corpus themselves are very small and it is hard to draw proper conclusions from such a limited number of samples.

3.3.2 Linguistic classifier

The tested classifiers achieved a high accuracy on the All Vowels test set, ranging from 74.25% to 82.93%. In general, ensemble methods reached a higher performance, but explainable methods such as K neighbors still reached almost 80% accuracy. If explainability was the focus of the project, we would therefore recommend choosing K neighbors as the main model. However, in our case, performance is critical, and we therefore arbitrarily chose the extra trees model over the random forest one. This model comes with a downside: Due to the high number of estimators it uses, its size is close to 200 MB.

Figure 4 shows the confusion matrix for the extra trees classifier. Globally, the model performs well, but issues occur with some vowel pairs such as (a, ɔ), (o, u) and (i, y).

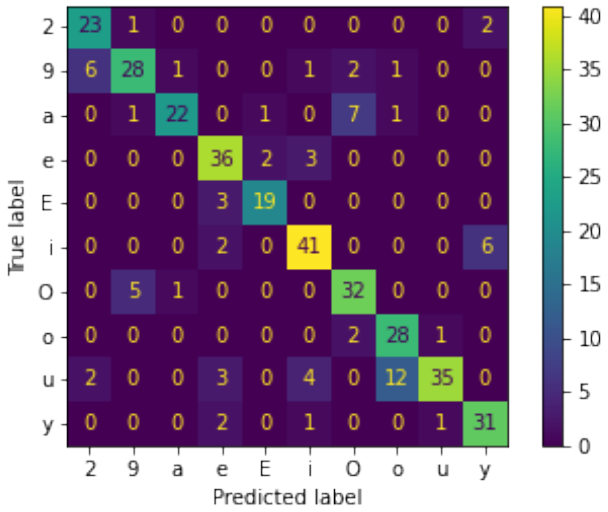

Figure 4: The confusion matrix for the extra trees classifier on the All Vowels test set.

3.3.3 Neural approaches

It is difficult to quantitatively estimate the performance of the vowel extractor. Qualitatively, we found that the model performed extremely well in virtually all situations in the All Vowel test set and real-life situations, with errors only occurring when the input audio had a low volume or the vowel duration was very short. The final total mean square error on the test set is 0.69371, which is fairly low. Furthermore, out of the 356 oral vowels in the test set, only 17 (less than 5%) had a mean boundary error of more than 10%, and only two of those had significantly wrong values. We therefore decided to keep this model.

The neural classifier obtained a final accuracy of 94.5946% on the All Vowels test set, an improvement of more than 10 percentage points over the linguistic classifier. Nevertheless, we chose to retain both classifiers for our final application, for several reasons:

- Test set performance is not indicative of real-life performance. Indeed, users wishing to train their French pronunciations are mostly non-native speakers, which are not represented at all in the All Vowels dataset. Furthermore, recording conditions are likely to be significantly worse in most use cases of our tool, since users do not typically have access to a professional recording room. This could translate into different noise levels and overall audio quality.
- Despite the relative simplicity of our model, the All Vowels dataset is fairly small. It is hard to say whether the neural model is prone to overfitting before testing it on in-domain data.

The confusion matrix of the neural model is given in Figure 5. The model performs almost flawlessly, but sometimes confuses /u/ and /ɔ/ as well as /e/ and /ɛ/.

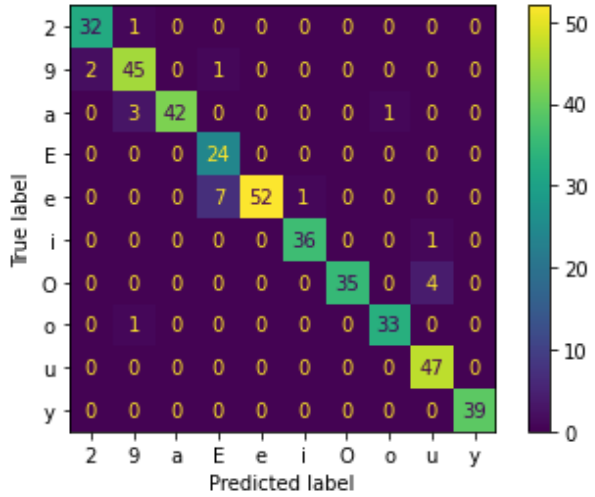

Figure 5: The confusion matrix for the neural classifier on the All Vowels test set.

4 Application

Our application, Vowel Tuner, allows the user to choose a given vowel they want to train on, and then to record themselves pronouncing a monosyllabic word chosen by our system. The audio record is saved in wave file format and used as the input for the linguistic and neural models. This communication is done through the server. After the vowel is predicted, the model sends through the server the results of the prediction to the user's browser as a feedback. A visualization of this process can be see in Figure 6. Moreover, our application is fully GDPR-compliant.

4.1 Application architecture

The application architecture is composed by two sides: client-side and server-side.

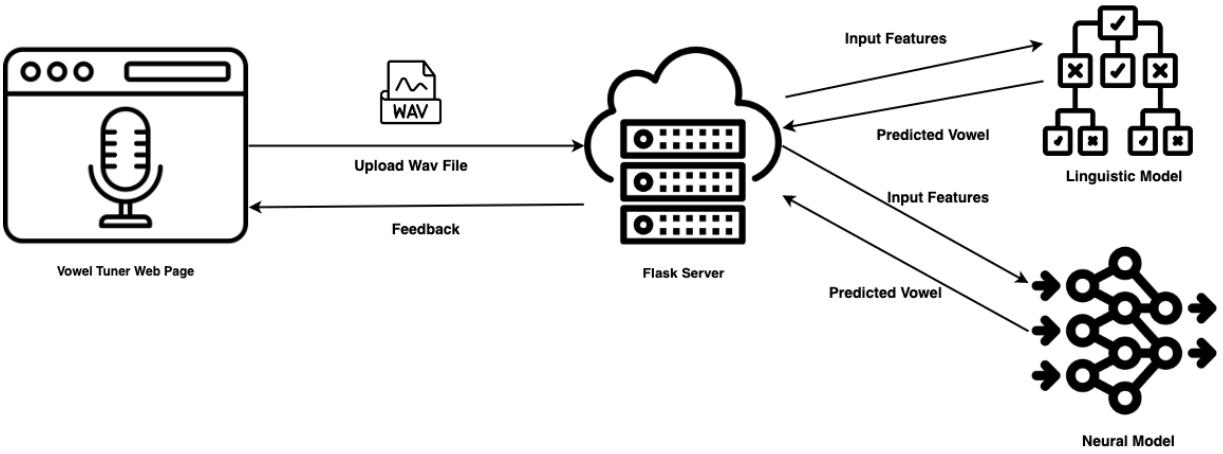

Figure 6: Vowel Tuner Application Diagram

4.1.1 Client-side

On the client side, the user can open the application in their web browser and record themselves with the microphone of their device. The recording and feedback process was done using JavaScript. In order to record the audio from the browser and save it as a wave (.wav) file, we used the Recorder.js [7] library.

In addition, we used HTML and Bootstrap 4 [8] to develop the web interface of Vowel Tuner.

The interface consists of several pages:

- A home page, where the user is given basic information about the application and asked to input their gender
- A vowel selection page, where the user can see all 10 French oral vowels and sample words, as well as how many of their pronunciation attempts have been recognized as the correct vowel so far
- A recording/prediction page, where a randomly chosen word is displayed for the chosen vowel and the user is prompted to record said word
- An output/feedback page, where the user is told whether their vowel was successfully recognized along with our model's confidence, and custom feedback in textual, audio and video form is given to help the user improve their pronunciation.

4.1.2 Server-side

For the server side development, we used Flask [9], which is a Python framework for web development. This framework allows the communication between the web app interface and the scripts which contains the implementation of the linguistic and neural approaches.

Once the prediction of the vowel is made, it is compared with the desired vowel and the output prediction, along with custom feedback, are returned to the user's browser.

4.2 Feedback

To effectively provide feedback for pronunciation is a complicated undertaking. A common method is to include cross-section visuals in order to illustrate correct tongue and lip posture, however, seeing a visual and applying the information to one's own vocal tract is not necessarily intuitive to the learner [17]. Therefore, for our tool, we decided to provide the following kinds of feedback:

[7] `https://github.com/mattdiamond/Recorderjs`
[8] `https://getbootstrap.com/`
[9] `https://flask.palletsprojects.com/en/2.2.x/`

- Audio comparison
- Visual aid
- Written instruction

First, the user can listen to their own recording as well as a recording of a native speaker for reference. Second, the user can watch a video of a native speaker pronouncing the vowel in order to get an idea of jaw and lip movement. Lastly, personalized written feedback is provided comparing the target vowel with the vowel perceived by the model. For example, if the user attempts to pronounce /y/ but the model detects /i/, the instruction would read: "Round your lips".

In order to provide written feedback, each vowel was tagged with four attributes: *openness*, *frontness*, *lip rounding* and *nasality*. The two latter were marked as either being present or not, whereas a discrete scoring was applied to *openness* and *frontness*. The scoring was based on the IPA chart provided by the International Phonetic Association as seen in Figure 7. *Openness* was scored on a 7-point scale ranging from close to open, while *frontness* was scored on a 5-point scale ranging from from back to front. This seemingly simplistic vowel representation was chosen in order to provide simple practical feedback. A more exact representation would not necessarily be more beneficial to the learner, since too precise instructions risk causing more confusion than clarity.

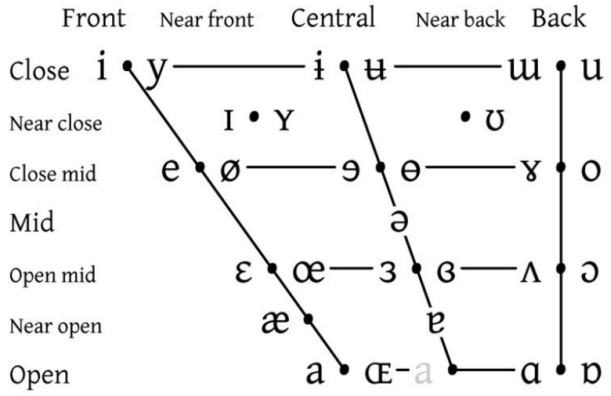

Figure 7: IPA vowel chart, courtesy of the International Phonetic Association.[10]

4.3 Experimental setup

For testing purposes, an evaluation module was added to our application. This module provides users with a simplified interface in which, after selecting their gender, the user was tasked with pronouncing a series of ten randomly sampled words from a list (given in Appendix D), with one word per oral vowel. Words were displayed in a random order to avoid systematic issues where the user's intonation might be falling in the last word of a series. For each word, users chose when to start and stop the recording, and could rerecord it as many times as wanted until they were satisfied. At the end of a series, users were prompted to choose whether to end the experiment or to record another series, up to a maximum total of 3 series (30 words) per user. Audio recordings were not kept, and upon ending the experiment, the set of input words, predictions and vowel probabilities for each model (the linguistic one and the neural one) were displayed in an easy to parse format and added to the final dataset.

During the recording, one or two annotators were present and manually annotated the vowel they perceived the user to pronounce. For example, some users tended to pronounce "se" as /sø/ while others pronounced it as /sœ/. The resulting dataset was then corrected accordingly, in order to not penalize or benefit the model.

In order to check whether the users spoke Parisian French, they were also asked "In which country or region would you say you grew up in?".

46 speakers took part in the recording, of which 27 were female and 19 were male. 14 of the speakers recorded 10 words, 20 speakers recorded 20 words, and 12 speakers recorded 30 words, summing up to a total dataset size of 900 words. The origin of the speakers is detailed in Figure 8. Of all recorded speakers, three participants (from Ukraine and Lebanon) considered themselves to be non-native French speakers. They were nonetheless kept as part of the final dataset for consistency and accuracy.

[10] `https://www.internationalphoneticassociation.org/content/full-ipa-chart#ipachartkiel`

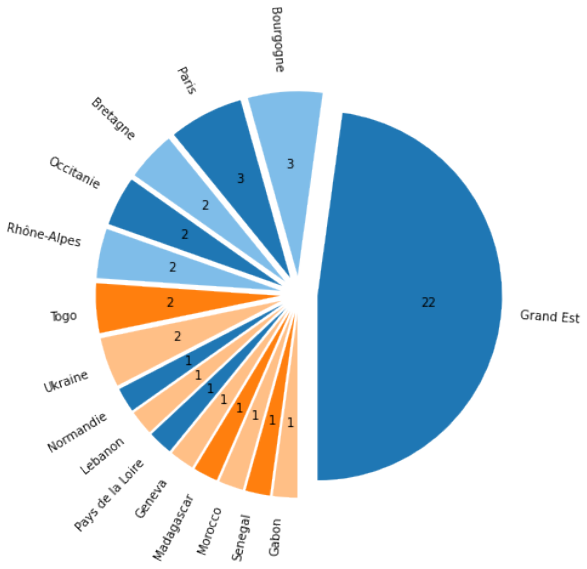

Figure 8: The origin of the speakers who tested the system. Blue sectors represent regions of France, while orange sectors represent foreign countries.

The experiment was performed on a laptop, placed in a relatively noisy room containing other people talking, and using a simple headset microphone. This was purposely done to replicate real recording conditions of a typical user of the application.

5 Results and discussion

5.1 Overall results

Table 6 describes the overall results of the classifiers.

Accuracy	Male speakers	Female speakers	All speakers
Neural model	50.00%	53.59%	51.56%
Linguistic model	**60.39%**	**77.69%**	**67.89%**

Table 6: Accuracy of the models depending on the speaker's gender.

A quick overview shows that while the neural model obtained better performance on the All Vowels test set, the linguistic model largely outperforms it on real data. The overall accuracy of both models is lower than on the All Vowels test set. There is also a strong discrepancy between the performance on male and female speakers in the case of the linguistic model.

The relatively low performance of the neural model can be explained by several factors. It is possible that it is due to overfitting, although it is not likely given that the performance on the All Vowels test set. We believe that it is more probable that the neural model is less robust to variations in the input, such as noise and sound quality. On the contrary, the linguistic model only relies on the values of the formants, which can be reliably extracted even when the input recording contains a high level of noise.

The discrepancy between the linguistic model's performance with respect to speaker gender is harder to explain, considering the gender parity in the training set. Furthermore, we found late in the project that the 5,000 Hz ceiling chosen for formant extraction is suitable for male speakers, but that a higher value of 5,500 Hz should be chosen for female speakers. The performance should therefore be lower than expected for female speakers, but it does not appear to be the case.

5.2 Per-vowel accuracy

Figure 9 shows accuracy of the models on a per-vowel basis.

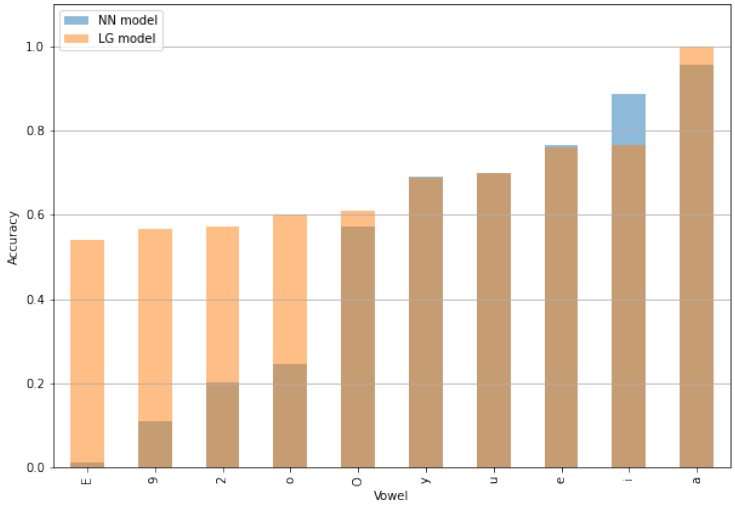

Figure 9: Accuracy of the neural (NN) and linguistic (LG) models for a each true vowel in the dataset.

We notice that the neural model performs significantly better for /i/, and that in general, both models perform similarly for cardinal vowels as well as vowels who are close to the edge of the (F1, F2) formant space. On the remaining vowels, however, the neural model's performance plummets to unacceptable levels, while the linguistic model's accuracy never drops below 50%. By splitting these statistics on a gender basis (see Appendix E), one can see that the neural model's performance gain on /i/ is only present for male speakers. We also see that the linguistic model's performance drops sharply for the central vowels in male speakers, but remains relatively consistent for female speakers.

Figure 10 shows the confusion matrices of both of the models.

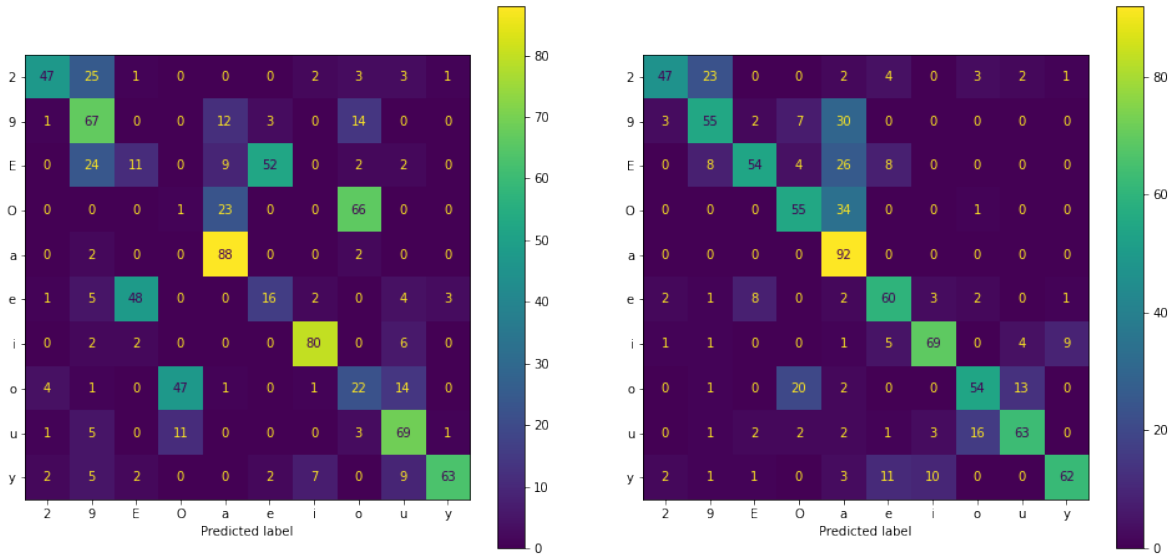

Figure 10: Confusion matrix of the neural (left) and linguistic (right) model.

The most common types of errors are not the same for both models. The linguistic model tends to over-predict /a/, while the neural model seems to mainly confuse two vowel pairs: (o, ɔ) and (e, ɛ). Interestingly, in each of these, the second vowel is present in a complementary distribution of the first one, found in closed syllables of the CVC form. This can be seen in the words “lot” /lo/ and “lors” /lɔʁ/. The classifier does not have access to the phoneme following the vowel and should therefore not have any bias in that regard.

5.3 Speaker distribution

Figure 11 shows how the model’s accuracy varied depending on the speaker.

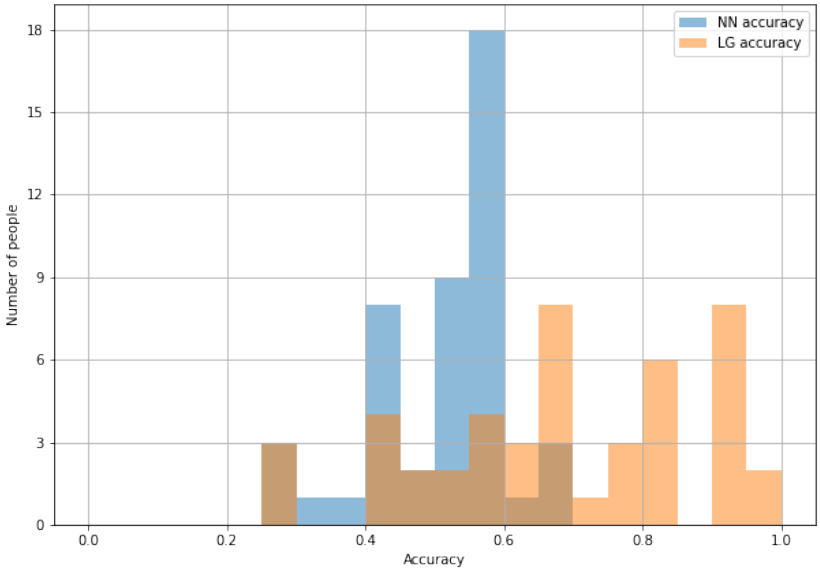

Figure 11: Distribution of the accuracy of the neural and linguistic models accuracy among speakers.

The neural model’s accuracy is very limited in range, and does never go above 60% for a given speaker, severely limiting its usefulness in practice. On the other hand, the linguistic classifier’s accuracy has a larger range and was able to achieve over 90% accuracy on a significant number of speakers.

Many factors may explain the variation between speakers, such as the volume of the recording as well as different types of voice and speaking speed. Indeed, errors due to a low recording volume consistently occurred for some of the speakers in the testing stage, while others were never confronted to this problem. It is not clear while the neural model’s accuracy seems to be capped at 60%, but it could be due to its consistently poor performance on some of the vowels.

A deeper look at the data showed that the linguistic model performed worse on female speakers as well as speakers who stated that they grew up outside of France. It performed especially well for speakers originating from Grand Est, which may be due to a majority of speakers from that region in the All Vowels dataset since it was recorded at the LORIA. However, we are not able confirm this hypothesis, as the identity and origin of the speakers were not available in the dataset.

We found no noticeable patterns in the accuracy of the neural network based on speaker gender or origin.

5.4 Model confidence and accuracy

In this last part, we wish to see whether there is a correlation between the confidence outputted by each model and its measured accuracy. Since each model outputs a probability distribution over the 10 target vowels, we can easily retrieve the highest confidence (corresponding to the predicted vowel) and second-highest confidence values.

Figure 12 shows how the model’s accuracy varies depending on the highest and second-highest probability returned for each vowel prediction.

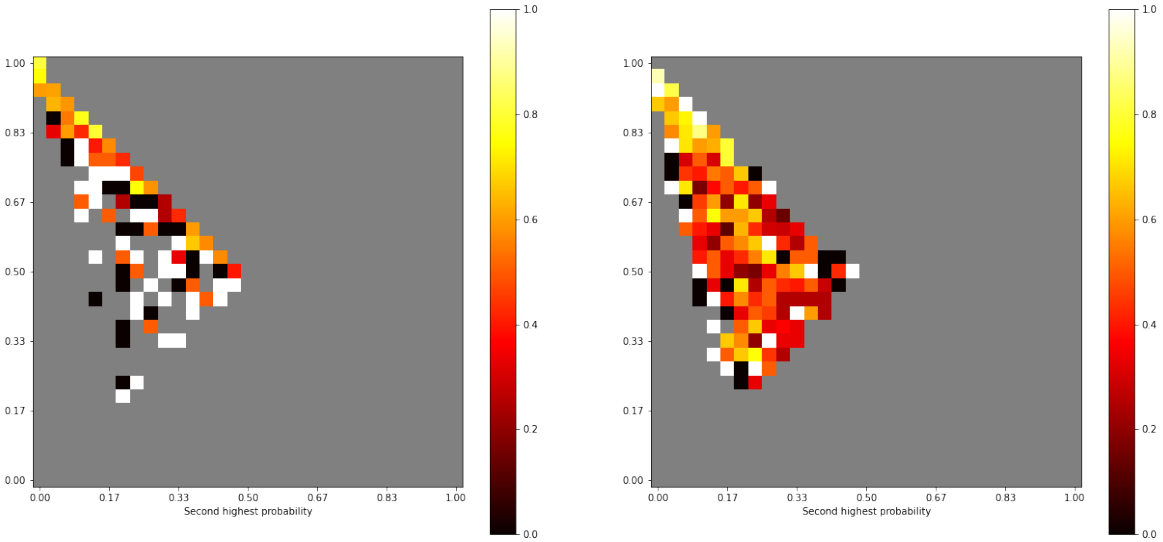

Figure 12: Accuracy of the linguistic (left) and neural (right) models depending on the value of the highest and second-highest probability returned.

For the neural model, one can see that having a high confidence value for the outputted vowel and a low confidence for all remaining vowels tends to correspond to a higher accuracy. For the linguistic model, the data is a lot sparser: for most predictions, the probability density was almost entirely concentrated in the top two values (represented by the diagonal in the figure). Values under the diagonal typically have a support lower than 3 and are not significant enough to be interpreted. On the diagonal, however, it seems that the model's accuracy increases with the value of the highest output probability.

Figure 13 shows the influence of the highest probability alone on model accuracy.

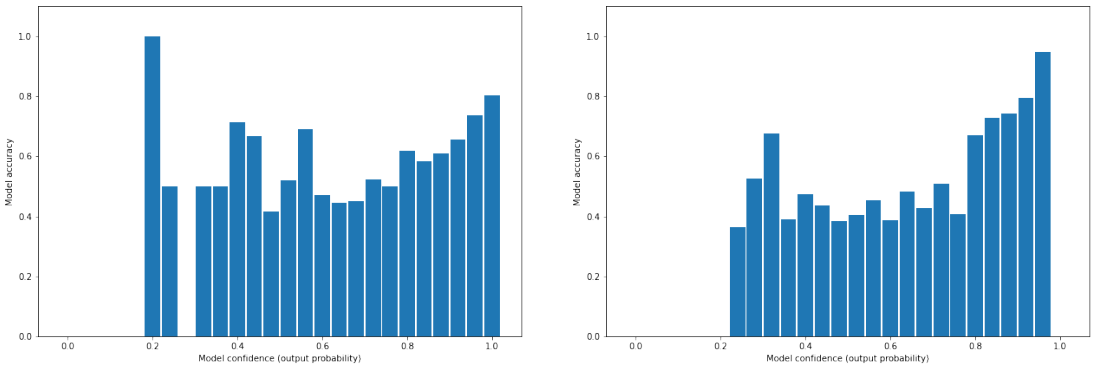

Figure 13: Accuracy of the linguistic (left) and neural (right) models depending on the value of the highest probability returned.

As one can see, the neural model's confidence is positively correlated with accuracy. For the linguistic model, however, this is less true: there are instances of the model having low confidence but being very accurate. In general, it seems that the neural model's confidence is more stable than the linguistic model's. This is confirmed by the data in Table 7, where it can be seen that the neural model makes better use of the probability space.

Model	Top-1 accuracy	Top-2 accuracy	Delta
Neural model	51.56%	64.56%	+13.00%
Linguistic model	67.89%	75.11	+7.22%

Table 7: Accuracy of the models when taking into account one or two of the highest probability values.

6 Conclusion and future work

In conclusion, the neural classifier performed remarkably well on the All Vowels test set, with an accuracy of 94.59%. However, it performed comparably poorly when tested in real-life conditions, with an accuracy of 51.56%, which can most likely be attributed to lack of robustness to background noise and sound quality. The model's confidence seems to correspond to the accuracy, where incorrect vowel predictions had a low confidence value.

The linguistic classifier achieved a high accuracy of 82.93% on the All Vowels test set using the extra trees model. In real-life conditions, the linguistic classifier outperformed the neural network with an accuracy of 67.89 %. However, the model oftentimes provided low confidence scores for correct predictions.

We firmly believe that despite our relatively good results, the Vowel Tuner project can be enhanced in several ways. First of all, we would like to support more words at both training and testing stage. In particular, in the training dataset, /ʁ/ was only present after some vowels. It would be better to also include it after other vowels for control purposes.

We believe that the neural model can also be improved. Since the high accuracy in the All Vowels test set did not translate into good performance at testing stage, we propose augmenting the existing dataset by adding noise to the audio recordings in order to increase model robustness. This would also increase the size of our training dataset, but might require architecture change in the neural model itself. Collecting more data in general might be beneficial to the neural model, but requires updating our privacy policy as we are not currently storing the recordings of any of the users.

The datasets we used also included nasal vowels (/ɛ̃/, /ɔ̃/, /ɑ̃/) that we did not use in our classifiers. It would be fairly easy to add those vowels into the pipeline and to re-train our existing models to include them, although the final accuracy is likely to further drop. More generally, we would be interested in adding support for arbitrary words and consonants to our project, but this is a hard task.

Our application currently requires the user to input their gender. For ethical purposes, we would like to remove our models' reliance on this information. We would also like to give custom feedback based on the speaker's native language, which can be useful in many ways according to language transfer theory. Finally, we would like to be able to support more varieties of French and not only the standard Parisian accent.

At the moment, our application is able to distinguish vowels in most of the case, especially for pairs of distant vowels. To be more useful to French learners, however, we believe that a minimal accuracy of 90% is required in all cases. In particular, we noticed during the testing stage that when our models showed a low final accuracy, speakers tended to trust the system and to assume that their pronunciation was wrong (even native speakers)! This underlines our responsibility, as NLP students, to ensure that our systems do not mislead users who may be likely to place high confidence in them.

Lastly, in our endeavor to provide a practical tool for French learners, it would be useful to receive feedback on the application from users, to see whether the given instructions were successful in helping the user achieve proper pronunciation of French vowels. Based on those results, the feedback could be tweaked to maximize the benefit of the user.

7 Environmental impact

NLP models can have a large environmental impact due to the energy costs associated with model training and inference, as well as web hosting. To evaluate the impact of our models, we therefore computed the associated energy consumption.

All of our models were trained and evaluated on a laptop equipped with a RTX 3060 graphical processing unit with a maximum Total Graphics Power (TGP) of 115 W.

We did not rigorously track the total training and inference time, but provide a generous upper bound of 10 hours. This corresponds to a maximum energy consumption of 1,150 Wh. Recent data[11] suggests that a typical French citizen consumes 2,220 kWh per year, or 6,082 Wh per day. The energy consumption of our models was therefore the equivalent of about 4.5h of a typical French citizen.

[11] `https://www.hellowatt.fr/suivi-consommation-energie/consommation-electrique/moyenne`

This low value can be explained by the reliance on computationally-light models for the linguistic classifier, and the small size of our neural model and of the dataset it was trained on.

8 Acknowledgements

The authors would like to thank professors Miguel COUCEIRO, Yves LAPRIE, Esteban MARQUER and Ajinkya KULKARNI for their feedback and advice. They also thank the students of the second year of the NLP Master of the IDMC for their contribution to the system, especially Karolin BOCZOŃ and Mathilde AGUIAR, as well as the students and professors who volunteered to test the application.

References

[1] Jessica S Miller. Teaching french pronunciation with phonetics in a college-level beginner french course. *NECTFL Review*, 69:47–68, 2012.

[2] Nancy F Chen and Haizhou Li. Computer-assisted pronunciation training: From pronunciation scoring towards spoken language learning. In *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 1–7. IEEE, 2016.

[3] Reconnaissance de phonèmes par analyse formantique dans le cas de transitions voyelle-consonne. - PDF Free Download.

[4] Cornelius Glackin, Julie Wall, Gerard Chollet, Nazim Dugan, and Nigel Cannings. Convolutional neural networks for phoneme recognition. pages 190–195, 01 2018.

[5] Juan Vasquez, Philipp Klumpp, Juan Rafael Orozco, and Elmar Noeth. Phonet: A tool based on gated recurrent neural networks to extract phonological posteriors from speech. pages 549–553, 09 2019.

[6] Kanishka Rao, Fuchun Peng, and Françoise Beaufays. Automatic pronunciation verification for speech recognition. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5162–5166, 2015.

[7] Kobayashi, Aozora and Wilson, Ian. Using deep learning to classify english native pronunciation level from acoustic information. *SHS Web Conf.*, 77:02004, 2020.

[8] M. Huzaifah. Comparison of time-frequency representations for environmental sound classification using convolutional neural networks, 2017.

[9] Lawrence Kuiper. Perception is reality: Parisian and provençal perceptions of regional varieties of french 1. *Journal of Sociolinguistics*, 9(1):28–52, 2005.

[10] Zsuzsanna Fagyal, Douglas Kibbee, and Frederic Jenkins. *French: A Linguistic Introduction*. Cambridge University Press, 2006.

[11] Bernard Rochet. Douglas c. walker. pronunciation of canadian french. ottawa: University of ottawa press. 1984. pp. xxii 185. $15.00 (softcover). *Canadian Journal of Linguistics/Revue canadienne de linguistique*, 32(1):101–107, 1987.

[12] B. Collins and I.M. Mees. *Practical Phonetics and Phonology: A Resource Book for Students*. Routledge English language introductions. Routledge, 2013.

[13] Raymond Kent and Houri Vorperian. Static measurements of vowel formant frequencies and bandwidths: A review. *Journal of Communication Disorders*, 74, 06 2018.

[14] Ayşe Enise Göker, İmran Aydoğdu, Ziya Saltürk, Güler Berkiten, Yavuz Atar, Tolgar Lütfi Kumral, and Yavuz Uyar. Comparison of voice quality between patients who underwent inferior turbinoplasty or radiofrequency cauterization. *Journal of Voice*, 31(1):121.e17–121.e21, 2017.

[15] Maurová Paillereau, Nikola. Do isolated vowels represent vowel targets in french? an acoustic study on coarticulation. *SHS Web of Conferences*, 27:09003, 2016.

[16] Laurianne Georgeton, Nikola Paillereau, Simon Landron, Jiayin Gao, and Takeki Kamiyama. Analyse formantique des voyelles orales du français en contexte isolé: à la recherche d'une référence pour les apprenants de FLE (formant analysis of French oral vowels in isolation: in search of a reference for learners of French as a foreign language) [in French]. In *Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 1: JEP*, pages 145–152, Grenoble, France, June 2012. ATALA/AFCP.

[17] Silke Witt. Automatic error detection in pronunciation training: Where we are and where we need to go. 06 2012.

Appendix A X-SAMPA reference table

Table 8 contains the translation table between IPA and X-SAMPA symbols for vowels used in this document.

IPA symbol	X-SAMPA symbol	Example
a	a	l**a**
e	e	p**eu**
ɛ	E	r**ê**ve
i	i	l**i**t
o	o	m**o**t
ɔ	O	p**o**rt
u	u	s**ou**s
y	y	v**u**
ø	2	d**eu**x
œ	9	n**eu**f
ɑ̃	A	l**en**t
ɔ̃	O	s**on**
ɛ̃	E	v**in**

Table 8: Matching table between IPA and X-SAMPA symbols for Parisian French vowels.

Appendix B All Vowels corpus detail

The list of French words pronounced by speakers in the recordings is as follows:

pi, lit, mi, si, ti, ti, pas, la, ma, sa, ta, ta, pou, loup, mou, sous, tout, tout, pé, les, mes, ses, tes, tes, paix, lait, mais, sait, taie, taie, port, lors, mort, sort, tort, tort, peau, lot, mot, seau, tôt, tôt, pu, lu, mu, su, tu, tu, peux, le, meut, ceux, te, te, peur, leur, meurs, sœur, teur, teur, pont, long, mon, son, ton, ton, pain, lin, main, saint, tain, tain, pan, lent, ment, sans, tant, tant, pan, lent, ment, sans, tant, tant.

Appendix C Linguistic classifiers

C.1 Confusion matrices

Figures 14 and 15 show the confusion matrices of the trained linguistic classifiers.

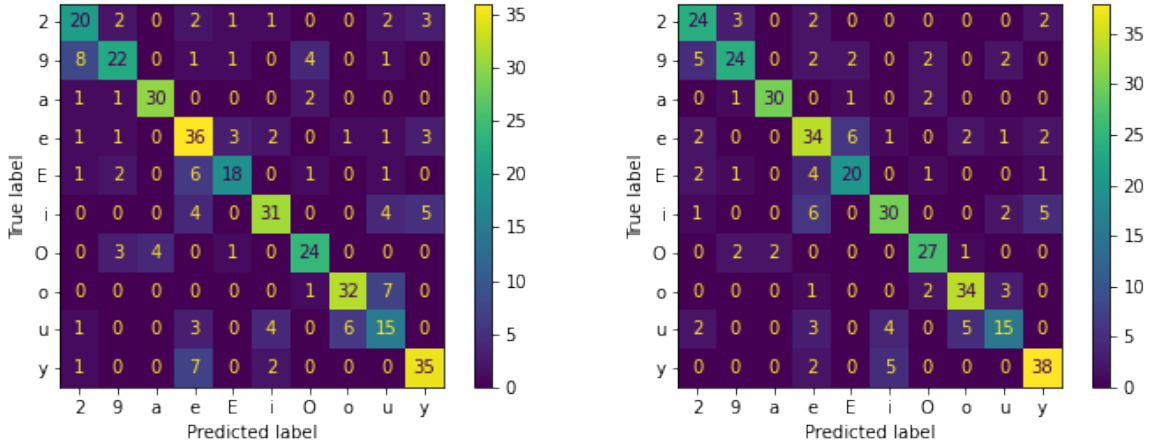

Figure 14: Confusion matrices for linguistic classifiers: decision trees (left) and k neighbors (right).

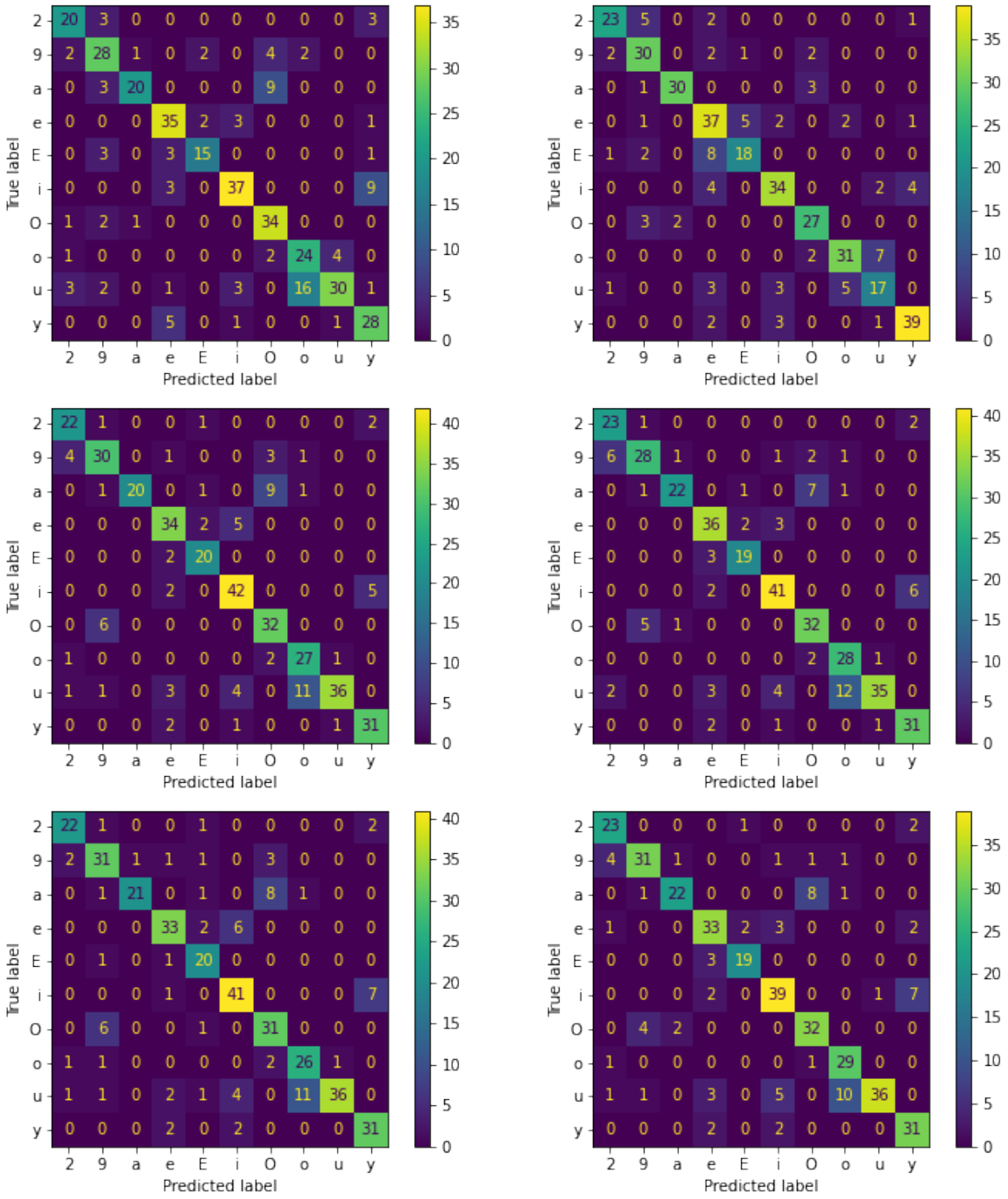

Figure 15: Confusion matrices for linguistic classifiers (from left to right and top to bottom: logistic regression, multilayer perceptron, random forest, extra trees, bagging and stacking).

C.2 Decision tree exploration

Figure 16 shows the first 4 decisions taken by the final trained decision tree classifier.

As expected, the values of formants, and particularly F1, and F2, are the most discriminating features for vowel distinction. Other features, such as the speaker's gender and the previous phoneme, are used deeper in the tree (not visible here).

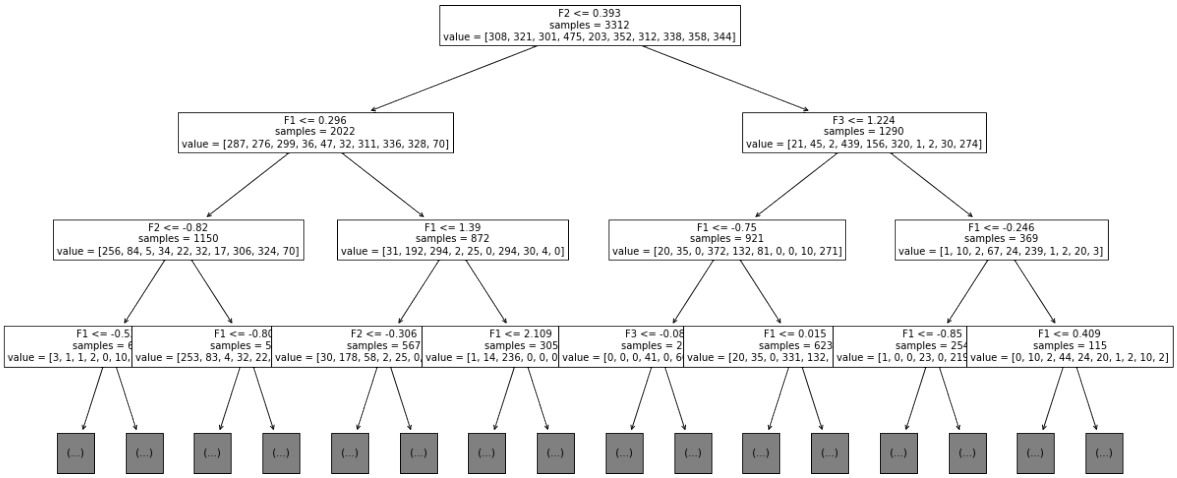

Figure 16: The final trained decision tree, up to a depth of 4.

C.3 Multilayer perceptron exploration

Figure 17 shows the 2-dimensional T-SNE projection of the input features (formants, speaker gender and previous phoneme) of each audio file in the dataset before and after being fed into the multilayer perceptron classifier. The second image is generated from the final hidden states of the classifier, before the last classification layer is applied.

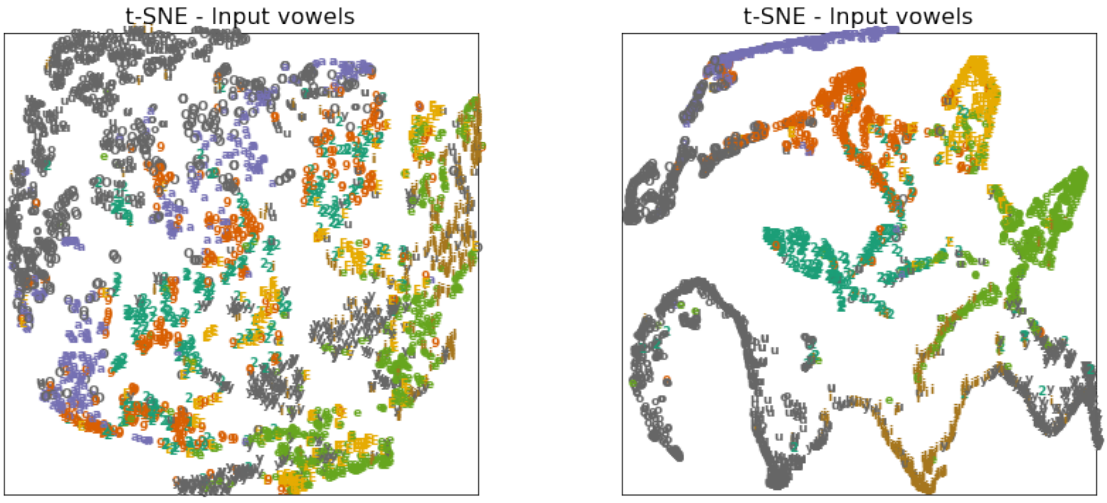

Figure 17: 2-dimensional T-SNE of the input dataset in its raw form (left) and before the final classification layer of the multilayer perceptron (right).

One can see how the classifier has learned to turn the input dataset into a more easily separable one. One can also notice problematic areas where mistakes are likely to occur.

Appendix D Words used for testing

Figure 9 gives the list of words that was selected for the final experiment. For each vowel, one word was sampled randomly, and the list of words was randomly shuffled to avoid order bias. Selected words follow the CV(ʁ) structure from the All Vowels corpus, where C is one of /l/, /m/, /p/, /s/ or /t/.

Vowel	Words
a	la, ma, pa, sa, ta
i	li, mi, pi, si, ti
u	loup, mou, pou, sous, tout
ɛ	l'air, mer, père, serre, terre
o	lot, mot, pot, seau, tôt
y	lu, mu, pu, su, tu
ɔ	lors, mort, porc, sort, tort
e	les, mes, pé, ses, tes
ø	le, me, peu, se, te
œ	leur, meurt, peur, sœur, -teur

Table 9: The list of available words in the application and evaluation module.

Appendix E Per-vowel model accuracy

Figures 18 and 19 show how the accuracy of the models for each type of true vowel in the dataset varies with respect to the gender of the speaker.

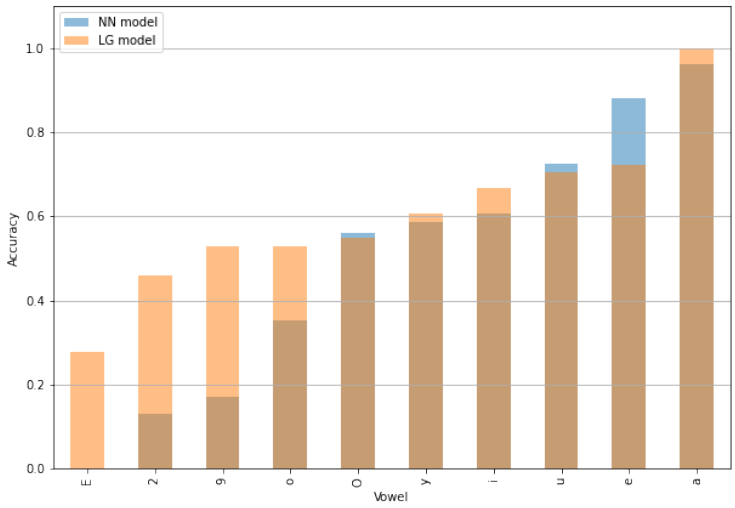

Figure 18: Accuracy of the neural (NN) and linguistic (LG) models for each true vowel in the real-life dataset (male speakers only).

The accuracy of the neural model is consistent across gender, and sometimes outperforms the linguistic model for male speakers.

The linguistic model performs better on recordings of female speakers for all vowels, and its accuracy stays above 60% in all cases, while it drops to unacceptable levels for the more central vowels in the case of male speakers.

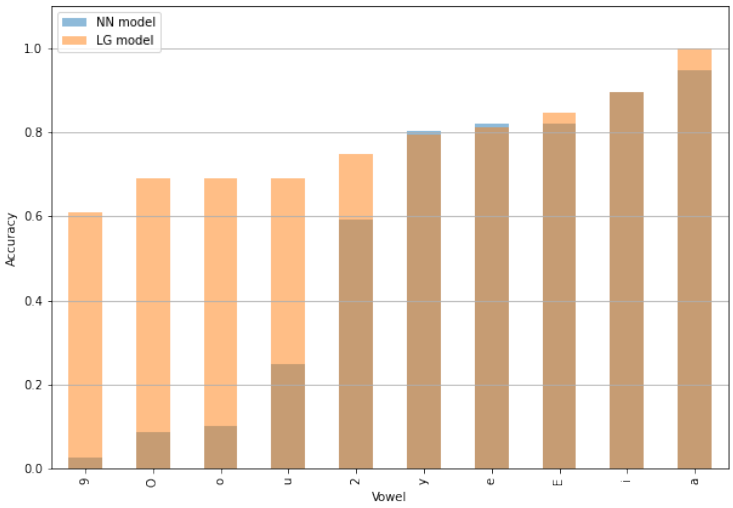

Figure 19: Accuracy of the neural (NN) and linguistic (LG) models for each true vowel in the real-life dataset (female speakers only).

Appendix F Vowel extractor experiments

Tables 10 and 11 describe the results of experiments for the neural regression model.

Convolutional layers	Max pooling layers	Dense layers	# Params.	Loss	Optimizer	Learning rate	Epochs	Validation loss
2, 3×3 kernel, 16/32 filters	2, 2×2 window	2 (64, 2)	374,146	MSE	SGD	0.01	15	0.4025
2, 3×3 kernel, 16/32 filters, **dropout 0.3**	2, 2×2 window	2 (64, 2)	374,146	MSE	SGD	0.01	15	0.4023
2, 3×3 kernel, 16/32 filters, dropout 0.3	2, 2×2 window	2 (64, 2)	374,146	MSE	SGD + **Plateau**	0.01	15	0.4023
2, 3×3 kernel, 16/32 filters, dropout 0.3	2, 2×2 window	2 (64, 2)	374,146	MSE	SGD + Plateau, **batch size 32**	0.01	15	0.9352
2, 3×3 kernel, 16/32 filters, dropout 0.3	2, 2×2 window	2 (64, 2)	374,146	MSE	SGD + Plateau, batch size 32, **normalized**	0.01	15	0.9477
2, 3×3 kernel, 16/32 filters, dropout 0.3	2, 2×2 window	2 (64, 2)	374,146	MSE	SGD + Plateau, batch size 32, normalized	0.01	**50**	0.6100

Table 10: The set of experiments that have been performed on the neural regression model. Bold values represent changes from the previous experiment or baseline.

Convolutional layers	Max pooling layers	Dense layers	# Params.	Loss	Optimizer	Learning rate	Epochs	Validation loss
2, 3×3 kernel, 16/32 filters, dropout 0.3	2, 2×2 window	1 (64, 2)	374,146	MSE	SGD + Plateau, **batch size 128**, normalized	0.01	200	0.2690
2, 3×3 kernel, 16/32 filters, dropout **0.5**	2, 2×2 window	1 (64, 2)	374,146	MSE	SGD + Plateau	0.01	50	0.2700
2, 3×3 kernel, 16/32 filters, dropout **0.1, 0.2**	2, 2×2 window	1 (64, 2), dropout 0.5	374,146	MSE	SGD + Plateau	0.01	50	0.2663
2, 3×3 kernel, 16/32 filters, dropout 0.1, 0.2	2, 2×2 window	1 (**32**, 2), dropout 0.5	189,570	MSE	SGD + Plateau	0.01	50	0.2693
2, 3×3 kernel, 16/32 filters, dropout 0.1, 0.2	2, 2×2 window	1 (**128**, 2), dropout 0.5	743,586	MSE	SGD + Plateau	0.01	50	0.2612
2, 3×3 kernel, 16/32 filters, dropout 0.1, 0.2	2, 2×2 window	**2 (64, 32, 2)**, dropout 0.5	376,258	MSE	SGD + Plateau	0.01	50	0.2467
2, 3×3 kernel, 16/32 filters, dropout 0.1, 0.2	2, 2×2 window	2 (64, **16**, 2), dropout 0.5	375,186	MSE	SGD + Plateau	0.01	50	0.2546
3, 3×3 kernel, 16/32/64 filters, dropout 0.1, 0.15, 0.2	2, 2×2 window	2 (64, 16, 2), dropout 0.5	140,930	MSE	SGD + Plateau	0.01	50	0.2774
2, 3×3 kernel, 16/32 filters, dropout 0.1, 0.2, **same padding**	2, 2×2 window	2 (64, 32, 2), dropout 0.5	466,370	MSE	SGD + Plateau	0.01	50	0.2459
2, 3×3 kernel, 16/32 filters, dropout 0.1, 0.2, **valid padding**	2, 2×2 window	2 (64, 32, 2), dropout 0.5	376,258	MSE	SGD + Plateau	0.01	50	**0.2414** (total test MSE: 1.00)
2, 3×3 kernel, 16/32 filters, dropout 0.1, 0.2, valid padding	2, 2×2 window	2 (64, 32, 2), dropout 0.5	376,258	**BCE**	SGD + Plateau	0.01	50	34.28 (total test MSE: 0.85)
2, 3×3 kernel, 16/32 filters, dropout 0.1, 0.2, valid padding	2, 2×2 window	2 (64, 32, 2), dropout 0.5	376,258	BCE	**AdamW** + Plateau	0.001	50	34.21 (total test MSE: **0.69**)

Table 11: The set of experiments that have been performed on the neural regression model. Bold values represent changes from the previous experiment or baseline (continued).

The last architecture also performed best for the neural classifier, with a parameter count of 192,202, a final training loss of 0.0783 and a test accuracy of 94.5946%.