GIPFA: Generating IPA Pronunciation from Audio

Xavier Marjou

Lannion, Brittany, France
E-mail: xavier.marjou@gmail.com

Abstract

Transcribing spoken audio samples into the International Phonetic Alphabet (IPA) has long been reserved for experts. In this study, we examine the use of an Artificial Neural Network (ANN) model to automatically extract the IPA phonemic pronunciation of a word based on its audio pronunciation, hence its name Generating IPA Pronunciation From Audio (GIPFA). Based on the French Wikimedia dictionary, we trained our model which then correctly predicted 75% of the IPA pronunciations tested. Interestingly, by studying inference errors, the model made it possible to highlight possible errors in the dataset as well as to identify the closest phonemes in French.

Keywords: audio; transcription; phonemes; Artificial Neural Network; sataset

1. Introduction

Some dictionaries such as Wiktionary offer a choice of both listening to words spoken by real users and reading phonemic pronunciations in the form of the International Phonetic Alphabet (IPA).

However, in the case of the French Wiktionary, the phonemic IPA transcripts are subject to a small percentage of errors. Several reasons can explain these errors. First, Wiktionary contributors may not be IPA experts; second, even IPA experts sometimes may make careless mistakes; third, the audio may be inconsistent because it is generally recorded independently without taking IPA pronunciation into account, which can lead to important discrepancies; fourth, some sounds such as /o/ and /ɔ/ may be very close to each other and can depend on the speaker.

This article examines whether such errors could be avoided by using a Natural Language Processing (NLP) tool to automatically extract phonemic IPA pronunciation from audio pronunciation.

For this purpose, we made use of Automatic Speech Recognition (ASR), which has already been the subject of in-depth studies. In particular, many recent implementation approaches have successfully used a deep Artificial Neural Network (ANN), such as in Han et al. (2020) and Das et al. (2019), hence our choice to design a new ANN called Generating IPA Pronunciation From Audio (GIPFA). In order to train and test it, we also assembled a new experimental dataset based on 80400 samples from the French Wiktionary.

Despite a dataset containing an unknown percentage of erroneous data samples, our GIPFA model succeeded in providing reasonable accuracy. Although it failed to replace IPA experts, it nevertheless proved to be particularly useful in identifying the biggest errors in the dataset.

2. Methodology

In order to predict the IPA pronunciation of a word, two main steps were necessary: identifying a relevant dataset and designing an ANN model capable of inferring an IPA pronunciation from an audio pronunciation.

2.1 Dataset

Word	Audio filename	IPA pronunciation
bonjour	LL-Q150 (fra)-LoquaxFR-bonjour.wav	bɔ̃ʒuʁ

Table 1: Dataset

Our dataset came from a Wikimedia dump[1] containing all pages and articles of the French Wiktionary. In this dump, each page generally contains three essential features: one *word* along with n main *IPA pronunciations* and m examples of *audio pronunciations* recorded by several speakers.

- A word is a text string containing Unicode characters. The *word* terminology has to be taken in the broad sense as a Wiktionary word contains common names, proper names words, abbreviations, numbers, and even sayings. Although our ANN did not use it, we kept the word in our dataset for debugging purposes, in order to have the possibility to again find the Wiktionary page containing the pronunciations.
- An audio pronunciation refers to an audio file generally recorded in a Waveform Audio File (WAV) format containing the pronounced word. Wiktionary pages can contain one or more audio pronunciations for the same word. When an audio file is generated with LinguaLibre (LL)[2] software, it benefits from three useful features: the audio file is under the Creative Commons sharing license[3]; the file can be fetched from Wikimedia Commons[4] based on its audio filename; the audio filename also contains a label representing a user name which can be used to identify audio files generated by users.
- An IPA pronunciation is a text string containing IPA symbols. For learning purposes, each audio pronunciation of a word should ideally be associated with a single IPA pronunciation transcribing this precise audio content; a ranking of the most common pronunciations might also be calculated and indicated in the page describing the word. However, most words have a single IPA pronunciation (i.e. $n = 1$) even when multiple audio pronunciations are available. Although some words have multiple IPA pronunciations (e.g. *coût*), a Wiktionary page rarely indicates which of these pronunciations corresponds to an audio file.

For our purposes, we restricted our dataset to samples containing:

- words in the French Wiktionary[5];
- French words, given that each Wiktionary describes words of several languages;
- words with a single IPA pronunciation, given that multiple IPA per audio sample introduce ambiguities;

[1] https://dumps.wikimedia.org/frwiktionary/20200501/
[2] https://lingualibre.org
[3] https://creativecommons.org/licenses/by-sa/4.0/
[4] https://commons.wikimedia.org/
[5] https://fr.wiktionary.org/

- IPA pronunciation containing symbols making part of the 37 traditional French phonemes (i.e. 'i', 'e', 'ɛ', 'a', 'ɑ', 'ɔ', 'o', 'u', 'y', 'ø', 'œ', 'ə', 'ɛ̃', 'ɑ̃', 'ɔ̃', 'œ̃', 'j', 'w', 'ɥ', 'p', 'k', 't', 'b', 'd', 'g', 'f', 's', 'ʃ', 'v', 'z', 'ʒ', 'l', 'ʁ', 'm', 'n', 'ɲ', 'ŋ');
- IPA pronunciation containing less than 20 phonemes, in order to keep our ANN model reasonable in size regarding our resources;
- audio files recorded with LL, in order to easily fetch audio files.

We also discarded 9 symbols that appear as optional in the IPA pronunciation of the French Wiktionary ('‿', '.', ' ', '‿', '' and 'ː', '(', ')', '-').

The resulting dataset contained 80200 samples from 102 different speakers. As depicted in Table 1, each sample contained three features: a *word*, an *audio filename* and an *IPA pronunciation*.

In addition, we also preprocessed the WAV files to have a fixed length of 2 seconds, and then converted them into a Mel-Frequency Cepstral Coefficients (MFCC) format so that they could serve as direct inputs into our model. Although processing audio files under a WAV format would be possible as in Sainath et al. (2015), it requires significant RAM memory, hence our choice to transpose them into an MFCC format, as usually performed in many studies, such as in Alcaraz Meseguer (2009) and Nahid et al. (2017).

2.2 Experiments

2.2.1 Model architecture

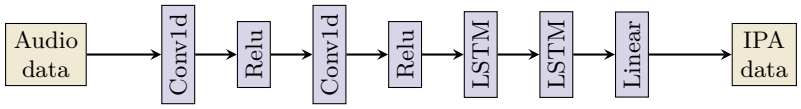

Figure 1: The GIPFA ANN model used for transcribing audio samples into IPA samples.

We modelled our GIPFA ANN as depicted in Figure 1. It contains typical components found in many ANN models used for ASR. However, given that we only had to translate a single word per sample, we did not use any Transformer component (Vaswani et al., 2017). Each audio input sample (MFCC data) first traversed a stack of two 1D convolution layer (Conv1D) layers to extract the shape of the MFCC data; followed by two Long Short Term Memory (LSTM) filters (Hochreiter & Schmidhuber, 1997) to extract temporal sequences; and finally followed by a linear layer in order to allow a Connectionist Temporal Classification (CTC) loss calculation (Graves, 2012). We did not allow the succession of two identical phonemes because this is rare in French words. In addition, we used an AdamW optimiser (Loshchilov & Hutter, 2017) with a learning rate of 1×10^{-4}.

2.2.2 Hyperparameters

We used Ray Tune (Moritz et al., 2018) for fine-tuning our hyperparameters with respect to accuracy results. This led us to identify a set of best values among a larger set of experimented values as summarised in Table 2. The resulting model contained 9,609,558 trainable parameters. Slight variations in the best values did not lead to significant

improvement. Although it is believed that a wider network may lead to better results (Nakkiran et al., 2019), we limited our model to these $10M$ parameters due to our limited computing resources.

Hyperparameter	Tested values	Best value
mfcc_coefficients	40	40
conv1d_activ	none, relu	relu
conv1d_layers	0, 1, 2, 3	2
conv1d_units	32, 64, 128	128
conv1d_bn	False, True	True
lstm_layers	0, 1, 2	2
lstm_units	128, 256, 512	512
lstm_dropout	0.1, 0.25, 0.5	0.5
lstm_bidir	False, True	True
lstm_bn	False, True	True
optimizer	Adam, AdamW	AdamW
lr	1e-3, 1e-4	1e-4

Table 2: GIPFA hyperparameters values

2.2.3 Training

For the training step, we used 79,326 samples distributed over 3,966 batches of 20 samples (3,927 training batches and 39 evaluation batches). During a preprocessing step, all audio samples were standardised with the mean (-11.48) and standard deviation (80.30) pre-observed with regard to the dataset.

Before each run, the data samples were randomly shuffled. Each training run took approximately 10 epochs of 3 minutes each on a single GPU (GeForce RTX 2080, 8 GB).

2.2.4 Test

For the testing step, we used 1,000 unseen samples to evaluate the performances of the GIPFA ANN.

2.2.5 Accuracy

Since solving the translation problem requires correct inference of the entire IPA pronunciation, we simply set for each tested sample an accuracy of 1 when our model predicted an IPA pronunciation equal to the tested target IPA pronunciation, or 0 otherwise. After each training run, we then calculated the average accuracy across all samples (i.e. an average accuracy between 0.0 and 1.0).

We performed 11 runs (with one training step and one test step for each) to allow reasonable confidence in the average accuracy results. We finally computed the mean accuracy and the associated standard deviation (std) for the 11 tests.

Since the dataset had not been studied further, there was unfortunately no baseline reference to challenge our results.

2.2.6 Further details on errors

To our knowledge, no study has examined the exactness and coherence of the audio files and IPA pronunciations of the French Wiktionary, meaning that the dataset may contain errors, making it difficult to assess whether a prediction error comes from the dataset or from the ANN.

In order to obtain more in-depth information on errors, we therefore also calculated three other metrics related to the 80000 samples in the dataset:

- At the word level
  - *Edit distance error*: the Levenshtein distance (Levenshtein, 1965) between the predicted IPA pronunciation and the target IPA pronunciation, in order to estimate how far the prediction was from the target.
- At the phoneme level
  - *Average phoneme accuracy*: the percentage of correct translations for each phoneme;
  - *Error pair percentage*: Since each of the 37 target phonemes can be incorrectly translated as one of the other 36 phonemes, the results can contain up to 37 * 36 categories of error pairs. To assess the representativeness of each pair, we calculated its number of occurrences divided by the number of phonemic errors.

The code is available on Github [6].

3. Results

In this section, we describe two different results: first, the accuracy of the model, then a more detailed observation of errors at phoneme level and at word level.

3.1 Accuracy

Table 3 presents the accuracy results which were consistent across the 11 runs; our GIPFA ANN model successfully predicted around 75 IPA pronunciations out of 100 audio samples.

Correctly inferred pronunciations had a mean length of 7.51, whereas incorrectly inferred pronunciations had a mean length of 8.65, thus indicating a slightly higher probability of error as the length of the IPA pronunciation increased.

[6] Code available at https://github.com/marxav/gipfa

Training samples	Tested samples	Pronunciation accuracy (mean)	Pronunciation accuracy (std)
79326	1000	0.75	0.02

Table 3: Pronunciation accuracy

3.2 Insight into the errors

Performing inferences on 80,000 samples of the dataset enabled a better understanding of the reasons for the errors.

3.2.1 Phoneme accuracy

Table 4 reports the translation accuracy of each phoneme. One phoneme (/ɑ/) had poor accuracy (less than 50%), five phonemes (/o/, /ŋ/, /œ̃/, /ɲ/ and /oe/) had moderate accuracy (between 65% and 89%), while the remaining thirty-one phonemes had high accuracy (over 90%).

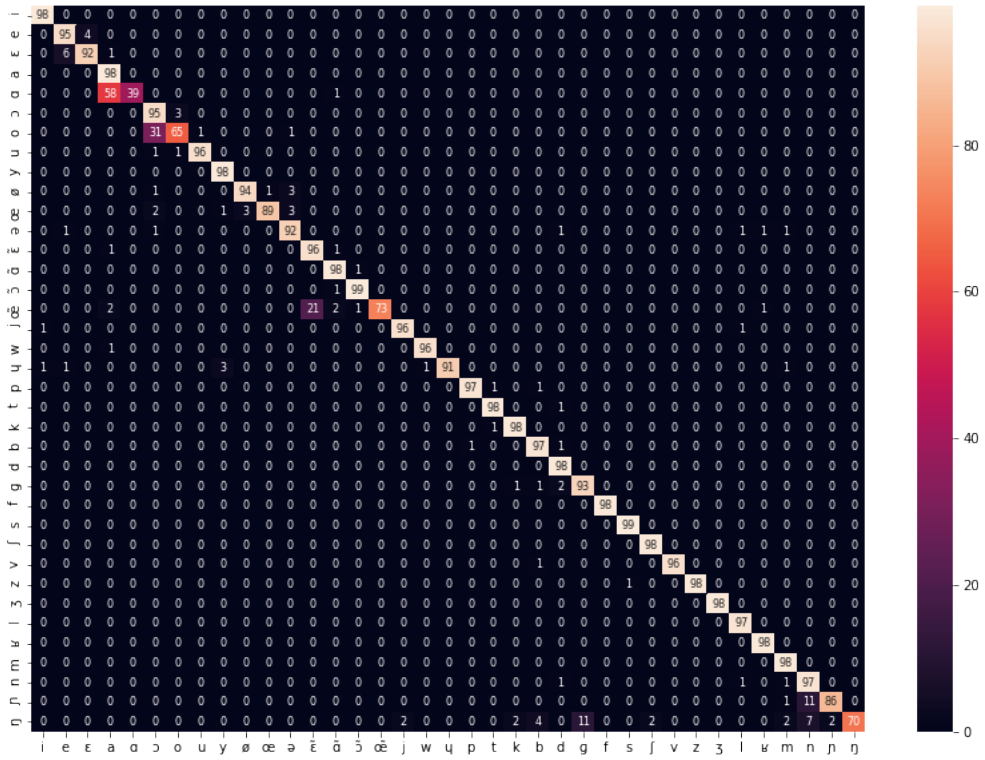

Figure 2: Confusion Matrix

To better observe the details, we also detailed these phoneme translation errors in a confusion matrix, as shown in Figure 2. Each row in the matrix represented a target phoneme while each column represented the distribution of the predicted phonemes. For instance, it turned out that the target phoneme /ɛ/ was predicted to be /e/ 6% of the

Target phoneme	Correct translation	Incorrect translation	Average accuracy
ɑ	392	605	0.39
o	4,615	2,485	0.65
ŋ	40	17	0.70
œ̃	241	89	0.73
ɲ	697	110	0.86
œ	2,459	301	0.89
ɥ	1,185	113	0.91
ɛ	15,859	1,472	0.92
ə	7,918	732	0.92
g	5,911	427	0.93
ø	2,587	169	0.94
ɔ	18,655	1,074	0.95
e	30,018	1,608	0.95
w	4,357	159	0.96
v	7,469	282	0.96
u	6,712	250	0.96
ɛ̃	4,527	192	0.96
j	12,567	547	0.96
b	12,753	434	0.97
n	13,165	472	0.97
p	14,845	464	0.97
l	23,181	684	0.97
ɑ̃	13,704	226	0.98
f	9,632	225	0.98
y	8,235	183	0.98
z	7,730	146	0.98
i	34,772	664	0.98
d	15,975	323	0.98
k	23,159	503	0.98
ʃ	4,407	92	0.98
a	44,575	707	0.98
m	17,334	313	0.98
ʁ	47,221	799	0.98
ʒ	5,552	137	0.98
t	29,691	713	0.98
ɔ̃	9,258	129	0.99
s	30,018	400	0.99

Table 4: Average accuracy of each phoneme

time, /ɛ/ 92% of the time, and /a/ 1%. Notable outliers were four large numbers outside the diagonal: 58% of /ɑ/ seemed to be poorly predicted as an /a/; 31% of /o/ as /ɔ/; 21% of /œ̃/ as /ɛ̃/; and 11% of /ŋ/ as /g/; It turned out that, like humans, the ANN had difficulties in differentiating close elementary sounds.

3.2.2 Error pair percentage

Table 5 represents the proportion of the error associated with each phoneme pair compared to the total errors of all pairs of phonemes. Interestingly, only three pairs of phonemes generated 31% of all errors: (/o/, /ɔ/) (15% of all errors), (/e/, /ɛ/) (12% of all errors), and (/a/, /ɑ/) (4% of all errors).

Target phoneme	Predicted phoneme	Percentage of all errors
o	ɔ	12.03%
e	ɛ	6.51%
ɛ	e	5.46%
ɑ	a	3.16%
ɔ	o	3.07%
t	d	1.25%
ɛ	a	1.04%
a	ɑ	0.83%

Table 5: Most encountered error pairs

3.2.3 Word-level distance error

Computed Levenshtein distance	
samples	mean, std
80000	0.31, 0.66

Table 6: Levenshtein distance

Table 6 reports a small mean Levenshtein distance and gives assurance that there is strong consistency between the audio content and the IPA pronunciation for the samples in the dataset studied.

However, Table 7 focuses on the most extreme outliers by reporting the 10 samples with the highest Levenshtein distance. Upon investigation, it was found that all of these 10 samples contained either an error in the audio sample (e.g. bad word pronunciation or no word spoken at all) or an error in the target IPA pronunciation, which meant that all these errors were in the dataset itself. These results therefore suggest that data samples whose pronunciations have a high Levenshtein distance probably contain an error.

Additional work would be required to identify the best threshold distance to identify possible errors in the dataset.

Word	IPA Target	IPA Prediction	Levenshtein distance
1337	/lit/	/mitasɑ̃tʁɑ̃mzɔt/	13
agent innervant	/aʒɑ̃inɛʁvɑ̃/	/gɔ/	11
brut de décoffrage	/bʁytdədekɔfʁaʒ/	/sbɔʁdedtɔʁ/	10
Michel	/miʃɛl/	/stɛ̃dəsɑ̃mʃɛl/	10
phalange proximale	/falɑ̃ʒpʁɔksimal/	/falɑ̃ʒ/	9
analyse calorimétrique	/analɔgʃimik/	/analiskalɔʁimetik/	9
àtha	/atɔ̃nœblavi/	/ata/	9
Wikitionnaire	/gazaefɛdəsfɛʁ/	/gɔʒifisølɛʁ/	9
arrondir par défaut	/aʁɔ̃diʁpaʁdefo/	/aʁɑ̃diʁ/	8
Luxembourg	/lyksɑ̃buʁ/	/yseʁzɔnb/	8

Table 7: Top-10 pronunciations with the highest Levenshtein distance

4. Discussion and Conclusion

Previous work has documented the effectiveness of the ANN model for ASR. However most studies have focused on the direct translation of audio samples into words.

In this study, we focused instead on the translation of audio samples into phonemes. We first proposed an ANN predicting with 75% accuracy the French pronunciations of the French Wiktionary.

Since to our knowledge no existing work has been done on this specific task and dataset, there was no basis for comparison or assurance as to the accuracy and consistency of the data.

We have shown that the translations of certain phonemes were more problematic since some phonemes are close elementary sounds (/o/ and /ɔ/; /ɛ/ and /e/; /ɑ/ and /a/) and thus difficult to distinguish. Future work may consider carefully checking the audio samples and IPA pronunciations containing these close phonemes, which would in turn enhance the efficiency of the ANN. In addition, future work could also involve synthesised audio examples and use them as additional samples to reinforce training data.

However, we have also shown that the Levenshtein distance between our GIPFA prediction and the target (as it exists in the dataset and therefore in the Wiktionary) can highlight the most suspect samples in the dataset. Such results therefore suggest that our GIPFA ANN would be a valuable tool to help verify the consistency of Wiktionary regarding pronunciation.

Therefore, integrating it into a tool like LL should be useful in order to suggest an IPA transcription. It could even be used to suggest an IPA transcription associated with each recorded audio sample, since having one IPA transcription per audio file should further improve the performances of the ANN.

Finally, we believe this method should be applicable to other languages provided that a sufficient number of training samples are available.

Acknowledgements

We thank all Wiktionary and LinguaLibre contributors for their contributions, without which there would be no wonderful free dictionary nor a free dataset.

5. References

Alcaraz Meseguer, N. (2009). *Speech analysis for automatic speech recognition*. Master's thesis, Institutt for elektronikk og telekommunikasjon.

Das, A., Li, J., Ye, G., Zhao, R. & Gong, Y. (2019). Advancing Acoustic-to-Word CTC Model with Attention and Mixed-Units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(12), pp. 1880–1892.

Graves, A. (2012). Connectionist temporal classification. In *Supervised Sequence Labelling with Recurrent Neural Networks*. Springer, pp. 61–93.

Han, W., Zhang, Z., Zhang, Y., Yu, J., Chiu, C.C., Qin, J., Gulati, A., Pang, R. & Wu, Y. (2020). ContextNet: Improving Convolutional Neural Networks for Automatic Speech Recognition with Global Context. *arXiv preprint arXiv:2005.03191*.

Hochreiter, S. & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), pp. 1735–1780.

Levenshtein, V.I. (1965). Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. *Reports of the USSR Academy of Sciences*, 163(4), pp. 845–848.

Loshchilov, I. & Hutter, F. (2017). Decoupled Weight Decay Regularization. 1711.05101.

Moritz, P., Nishihara, R., Wang, S., Tumanov, A., Liaw, R., Liang, E., Elibol, M., Yang, Z., Paul, W., Jordan, M.I. et al. (2018). Ray: A distributed framework for emerging {AI} applications. In *13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18)*, pp. 561–577.

Nahid, M.M.H., Purkaystha, B. & Islam, M.S. (2017). Bengali speech recognition: A double layered LSTM-RNN approach. In *2017 20th International Conference of Computer and Information Technology (ICCIT)*. IEEE, pp. 1–6.

Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B. & Sutskever, I. (2019). Deep double descent: Where bigger models and more data hurt. *arXiv preprint arXiv:1912.02292*.

Sainath, T., Weiss, R.J., Wilson, K., Senior, A.W. & Vinyals, O. (2015). Learning the Speech Front-end with Raw Waveform CLDNNs. In *Interspeech*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008.