Inference of pronunciation difficulty from non-native data

João Pedro Sousa Correia

Instituto Superior Técnico, Universidade de Lisboa, Portugal

Abstract— ELSA Corp. developed a CAPT (Computer Aided Pronunciation Training) system that assists its users to improve their American English accent. In order to develop exercises, appropriate for the level of its users, it is important to have a metric capable of assessing the difficulty of their exercises, according to the user's proficiency level. Therefore, the objective of this thesis is to develop a system capable of determining the pronunciation difficulty associated with a certain utterance and its phonemes, for a Vietnamese student of English. Our model uses a Neural Network in order to forecast the probabilities associated to how competently the user pronounce each of the utterance's phonemes. Then, using these probabilities, the system computes the difficulty score associated to the phoneme and the difficulty score associated to the utterance. In the end, we have a system able to receive as input an utterance and the proficiency level of the user. Then, the system outputs difficulty scores for the utterance and its phonemes.

***Index Terms*— Pronunciation Difficulty, Phonemes, Neural Networks, Computer Aided Pronunciation Training**

I. Introduction

NOWADAYS, one of the most spoken languages in the world is English. Therefore, more and more people around the world are trying to learn this language.

One of the most difficult, yet sometimes overlooked, aspects of learning English is mastering its pronunciation and even students with a high theoretical knowledge tend to struggle in this area. This is particularly truthful for students that have a native language that is substantially different from English. In these cases, they usually have a harder time mastering the target language accent. In some situations, these difficulties are so extreme that even if the students can read and write competently, they still struggle to be understood by native speakers due to their foreign accent.

To solve these problems, ELSA Corp. provides a CAPT (Computer Aided Pronunciation Training) system capable of helping students of English around the world improve their American English accent.

In order for their users to practice their American English accent, they designed multiple exercises.

In these exercises, a user says an utterance, which the system records. Then, the system automatically evaluates it according to how close the user was from the American English accent. After the evaluation, the system provides a score representative of the user performance and gives a feedback in how to improve his accent.

To provide appropriate exercises for their students' proficiency level, it is important to understand the difficulty of a specific exercise. Therefore, having a system capable of evaluating the difficulty of a certain utterance would be helpful to design and assign exercises to their users.

In this thesis, we developed a system capable of automatically evaluating an utterance according to the difficulty that a Vietnamese student of English, with a certain proficiency level, would have pronouncing that sentence. In this document, we will present this system.

This report is divided in five chapters: Introduction, Related Work, Methods, Results and Conclusions.

In Related Work, we will present some research works that we used as a base to develop our system. In Methods, we present step by step how our model functions and how we evaluated it. In the Final Results Chapter, we present the results obtained after evaluating the model. Finally, in the Conclusions chapter we summarize the method used, analyze the results obtained, discuss the limitations of our model and specify some aspects that could be improved in future work.

II. Related Work

Most of the bibliography in the theme of pronunciation difficulty for Vietnamese students of English, concentrates in the usual difficulties they experience.

In these research papers, the authors usually compare the English and Vietnamese phonology and identify phonemes that are more difficult to pronounce. In these situations, the authors often highlight the positions and contexts in which these phonemes cause more difficulties for the students. These research papers usually also discuss how the suprasegmental aspects (not related to a particular phoneme) influence the pronunciation of Vietnamese students of English. [1]

Although, these research papers are important and useful to understand the pronunciation difficulties Vietnamese students experience. They do not have a similar objective to ours,

because they identify the difficulties that the students have, but they do not assess a score for a certain input text, according to its pronunciation difficulty.

Although this problem is not extensively explored there are some research in this area. In [2], the authors had to automatically assess the difficulty associated to a certain input text for Korean students of English. In their model, the authors trained a Support Vector Machine, using as input, features related to: the length of the input text, the number of phonemes difficult to pronounce (according to the bibliography on the topic), the number of consonant clusters ….

Due to the reduce number of research work on this area, we focus on investigating an area that is similar (but not equal) to our problem. This area was readability assessment.

In readability assessment, the objective is to obtain a score representative of the difficulty that a person will have to comprehend a certain input text.

Initially, the problem was solved using formulas to compute a readability score. These formulas used characteristics of the sentence related to: the length of the sentence, the length of the word, the number of syllables …. [3]

As the research on readability improved other characteristics were tested and new formulas emerged. [4]

More recently, machine learning has also been used to assess the readability of text.

One of the methods that uses machine learning, extracts certain features from the text (similar features to the ones used in readability formulas) and then trains the model using these features as input. [5],[6]

More recently, due to the success that Neural Networks demonstrated in numerous fields, they also have been used in this area. Using Neural Networks, we do not have to define the features for the text. Instead, we can input the raw data in the Neural Network and allow it to learn from the data. This method has the advantage that the model can find patterns in the data that were not specifically encoded. An example of this method is Vec2Read [7]. Vec2Read is a model that takes as input a certain text and uses an attention mechanism in order to focus on certain areas of the text. Based on these areas, the system can assess a readability score for the text.

III. Methods

In this chapter, we will first define the problem of our thesis and indicate how we divided our database. Then, we will present the methods used to solve the problem and how we evaluated the model.

A. Problem Definition

In this project, we developed a system that receives as input an utterance and the proficiency level of a user. Then, the system outputs a score representative of the pronunciation difficulty of that utterance, for a user with that specific proficiency level. Additionally, the system outputs a similar score for each of the utterance's phonemes.

To train the model, we have a database with multiple utterances and the evaluation for multiple users, effectuated by the ELSA system.

It is to be noted that the scores present in the database do not express the inherent difficulty of the sentence. Instead, the scores reflect the performance of a certain app user.

This problem is analogous to assessing the difficulty of a school exam because we cannot determine the difficulty of an exam based on a single student result. Instead, we should look at all the results of the students to obtain an idea of the exams difficulty.

In the same way, we cannot determine the difficulty of a certain utterance (or its phonemes) just by observing a single user evaluation. Instead, we should observe all the results the users obtained. So, in our system we will first forecast the probability distribution function (pdf) of the sentence's phonemes and then forecast the pdf associated to the utterance's scores.

After obtaining the pdfs for the phonemes and the pdf (or a computationally generated sample set representative of the pdf) for the sentence we obtain the difficulty score using the appropriate formula for both cases.

B. Database Division

We divided the database in two main sets: Usable Dataset and Test Set A. The Test Set A will be used exclusively for testing the overall algorithm. The Usable set will be subdivided in three other subsets: Train Set, Validation Set and Test Set B, these subsets will be used to train the Neural Network that we used for a component of our system.

Furthermore, from the Usable Dataset, we extracted the n-grams of phonemes (p_i) and their corresponding score sample sets ($s_i{}'$ =[$s_{i1}{}'$,$s_{i2}{}'$,$s_{i3}{}'$, …., $s_{iN}{}'$]), where i corresponds to a phoneme n-gram and $s_{i1}{}'$ to a score, extracted from the dataset, that evaluates the performance of a user.

Using this data, we prepared a hash-table (H_1), which receives an n-gram (p_i) and the user's proficiency level as input and outputs the corresponding sample set (s_i). This associative array will be used for our system.

In the next subsections we will describe in detail the workflow of our system.

C. Pre-Process input data

First, we remove all the punctuation and transform all the letters in the text into lower case letters.

Secondly, we resort to a hash-table to obtain the phonetic transcription of the utterance. This transcription has all the phonemes in the utterance. Furthermore, it also indicates the stress of the vowels and the position of each phoneme in the word.

Then, we use the sliding window algorithm in order to obtain multiple n-gram composed of an impar number of phonemes (in our case we selected 3 phonemes). In the end, we will have the same number of n-grams and phonemes.

D. Forecast phonemes sample set

After obtaining the n-grams of the sentence, we compute the sample set of the n-gram's central phoneme.

As we can observe in equation (1), this sample set can be obtained in two different ways: if the n-gram is in the associative array (H_1), we simply input the n-gram and obtain the sample set, if the n-gram is not in the associative array we obtain it from a Neural Network (H_2), that we previously trained. In this section, we will focus mainly in the second form of obtaining the n-gram's sample set.

$$s'_i = \begin{cases} H_1(p_i, l), & if\ p_i \in H_1 \\ H_2(p_i, l), & if\ p_i \notin H_1 \end{cases} \quad (1)$$

In equation (1), p_i corresponds to the n-gram, l corresponds to the proficiency level of the user and s'_i corresponds to the scores sample set.

The user can have four proficiency levels: low, mid-low, mid-high and high.

To train the Neural Network H_2, we used the data sets: Train Set, Dev Set and Test Set B. The input of this Neural Network (NN) was the n-gram. To encode the n-gram we encoded each of its phonemes using one-hot-encoding and concatenating it with: the code associated with the position of the phoneme in the word, its stress (in case the phoneme is a vowel) and the proficiency level of the user. The output of the NN corresponds to the quantized pdf associated to the central phoneme when he is in that specific n-gram.

The Neural Network used to train the model was a Feed Forward Neural Network. The only particularity about the Network was its output layer. In this layer, each of the output neurons represented the probability of obtaining a score in a certain interval. In our case, we opted to use five output neurons. Each of these neurons corresponded to the following score intervals:

- Output Neuron 1: Represents the probability of obtaining a score in the interval: [0, 0.02).
- Output Neuron 2: Represents the probability of obtaining a score in the interval: [0.02, 0.07).
- Output Neuron 3: Represents the probability of obtaining a score in the interval: [0.07, 0.2).
- Output Neuron 4: Represents the probability of obtaining a score in the interval: [0.2, 0.995).
- Output Neuron 5: Represents the probability of obtaining a score in the interval: [0.995, 1].

Furthermore, since the output of the network was a quantized pdf, the sum of the values in the array had to equal one, so we had to use an output layer with a softmax activation function.

Because, the output of the NN is a pdf and our objective is to obtain a sample set representative of that pdf, we still have to generate its sample set. In order to generate the sample set, we considered the quantized pdf as a sum of five uniform distributions, which their area corresponds to the probability of each of the output neurons. Therefore, we used the following algorithm in order to obtain the sample set from the quantized pdf:

1. Initialize an array of length M (it has to be sufficiently large), with values from 1 to 5 that represent the output of the Neural Network. Each value of the array is chosen randomly in which: the probability of giving an element the value 1 is equal to the probability determined by the output neuron 1, and the same logic is applied for values 2, 3, 4 and 5.

2. Substitute the values of 1 to 5 in the array for phoneme scores. In order to choose the new value in the array we first check the previous value that corresponds to the number of an output neuron interval. Then, we sample a number from an uniform distribution with the same limits as the corresponding output neuron interval. The sampled number is the new value of the array. We do this step for the M elements of the array.

After following this method, we obtain the scores sample set of the n-gram's central phoneme.

E. Forecast sentence sample set

In the dataset, the score obtained by a user can be computed using a particular formula (g). This formula takes as input: the sentence and the scores that the user obtained for each of its phonemes. Then, the formula outputs the score that the user obtained for the utterance.

As we previously mentioned, we will follow a similar approach. In this case, we will use the phonemes scores sample set to compute the sample set of the sentence.

Therefore, the method we used takes as input: the utterance, the sample set of the phonemes and the formula (g) and outputs the sentence sample set.

The method we selected was the Monte Carlo Simulation. In our version, we have multiple phonemes' scores sample sets (s_i' = [s_{i1}',s_{i2}',s_{i3}', …., s_{iN}']) and we randomly select one sample (s_{ik}') from each of the n-grams' score sample set. Then, we compute a sentence score using the formula:

$$y'_j = g(utterance, s'_1, s'_2, \dots, s'_N) \quad (2)$$

Where N corresponds to the number of phonemes in the utterance.

After repeating this process multiple times and storing the values in a vector, we obtain $y' = [y'_1, y'_2, \dots, y'_L]$ which represents the scores sample set for the utterance.

F. Compute Difficulty Score for the phonemes and utterance

As previously mentioned, the difficulty score should be representative of the scores sample set.

Therefore, once we obtain the phonemes and utterance sample sets, we can compute the difficulty score for the phonemes and utterance, respectively.

If we observe the histograms of the phonemes scores, we verify that they are very polarized. Which means, they tend to concentrate in the maximum and minimum scores and less in

the intermediate ones. So, we defined the Difficulty score as the probability of the phoneme being well pronounced. Which in practice has the following formula:

$$s_i = \frac{number\ of\ correct\ samples}{total\ number\ of\ samples} \qquad (3)$$

We defined as a correct sample, a score in the interval [0,0.02), which is the same interval as the output neuron 1 from the Neural Network (H_2). In other words, the first output neuron of H_2 computes the difficulty score for the phoneme.

For the utterances, we also defined a difficulty score that describes the utterance's pdf. In this case, the utterance histograms are not as polarized, therefore we use the Expected Value to express the difficulty of the utterance (y):

$$y = \frac{1}{N_u}\sum_{i=1}^{N_u} y_i' \qquad (4)$$

In (4), N_u represents the number of samples in the scores sample set, obtained from the Monte Carlo Simulation.

G. Evaluation of the Method

The model has two outputs: the difficulty score for the phonemes and the difficulty score for the sentence. Therefore, we need to evaluate the model for both of these outputs.

To evaluate how the model assesses the difficulty score for the phonemes, we evaluate how the Neural Network (H_2) forecasts the quantized pdf and how it predicts the interval: [0,0.02), that corresponds to the phoneme difficulty score. As the ground truth, we will use the Test Set B and we will compare the quantized pdf obtained from the Test Set B and the quantized pdf obtained from H_2.

To evaluate how the model computes the Difficulty Score for the utterance, we use the Test Set A. This set has multiple utterances and their respective scores sample set, for multiple proficiency levels. Therefore, to evaluate the model we compute the Difficulty Score (y_{pred}) for every utterance, using our model. As ground truth, we extract the score samples sets from Test Set A and we compute the Difficulty Score (y_{true}), also using equation (4).

Once we obtain the output for the utterance and its phonemes, we use the following metrics to evaluate the model:

1. Mean Absolute Error (MAE): Computes the average absolute value of the error. It is a scale dependent metric.

$$MAE = \frac{1}{N_T}\sum_{n=1}^{N_T}\left|y_{true_n} - y_{pred_n}\right| \qquad (5)$$

2. R2-Score: It is a scale independent metric. That can have negative numbers but has a maximum of 1, which represents a perfect prediction.

$$R2 - Score = 1 - \frac{1}{N_T}\sum_{n=1}^{N_T}\frac{\left(y_{true_n} - y_{pred_n}\right)^2}{\left(y_{true_n} - \bar{y}\right)^2} \qquad (6)$$

Where N_T corresponds to the size of the respective Test Set and $\bar{y}$ corresponds to the mean of y_{true}.

For the output of the utterance, we just need to apply the metrics above to the difficulty scores obtained.

For the output of the phonemes, we evaluate how the Neural Network predicts the quantized pdf. We also evaluate this using the metrics above, but for each of the output neurons of the NN. To evaluate overall the Neural Network, we compute the average of the MAE and R2-Score for these output neurons.

IV. Final Results

In this section we will present the final results for our model, according to the evaluation metrics previously mentioned. The model has two outputs, so we must present the results for both outputs.

Considering that the NN is where the error associated to the phonemes' difficulty score occurs, first we will evaluate the NN which forecasts the quantized pdf.

In the Table 1, we present the results for each of the output neurons, according to the metrics and methods presented in Chapter III, Section G.

TABLE 1
RESULTS OBTAINED USING THE NEURAL NETWORKS.

Output Neuron Number - Score Interval	MAE	R2-Score
1 – [0, 0.02)	**0.077**	**0.81**
2– [0.02, 0.07)	0.068	0.4
3 – [0.07, 0.2)	0.054	0.51
4 – [0.2, 0.995)	0.057	0.66
5 – [0.995, 1]	0.042	0.64

The highlighted line in Table 1 corresponds to the evaluation of the phonemes' difficulty score, predicted by the NN. As we can observe, the model is able to efficiently predict the difficulty of the phonemes, according to its context and the proficiency level of the user.

To evaluate how the model predicts the utterance difficulty score we used the Test Set A, as described in Chapter III, Section G. In Table 2, we present the results obtained comparing the difficulty score generated from the algorithm and computed from the Test Set A. In this table, we also present the result, in the case we only use H_2 (in the model we use H_1 and H_2, as mentioned in equation (1)) to generate the n-gram's score sample set.

TABLE 2
RESULTS OBTAINED FOR THE OVERALL SYSTEM.

	MAE	R2-Score
Results when we only use the NN (H_2) to predict the n-gram score's sample set	6.35	0.55
Results when we use the algorithm as described in Methods.	6.3	0.55

As we can observe from Table 2, the model predicts similarly with the hash-table H_1 and without the hash-table H_1. That is the case, because Test Set A does not have a lot of n-grams present in the hash-table. But in the hypothetical case in which there was a lot of n-grams in common between the hash-table and the Test Set A utterances, the overall results of the model would have improved using H_1.

Furthermore, we can observe from this results that the model can predict the difficulty for the utterance. Although, not as accurately as it can for the phonemes' difficulty.

V. Conclusions

In this thesis, we developed a system capable of automatically assessing the inherent difficulty that a person, with a certain proficiency level, would have pronouncing a certain utterance. This system receives as input an utterance and the proficiency level of the user and outputs the difficulty scores for the utterance and for all its phonemes.

To develop this system, we had a database with multiple utterances. For each sentence, we had multiple scores that indicated how well the users were able to pronounce the sentence and each of its phonemes.

In the approach that we used to model our system, we obtained sample sets representative of the pdfs for both the sentence and its phonemes. Then, from those sample sets we extrapolated the difficulty scores for the sentence and its phonemes using the appropriate methods presented in this thesis.

In the end, we obtained a system that is capable to automatically assess the difficulty of the utterance and its phonemes.

This system predicts more accurately the difficulty scores associated with the phonemes than the difficulty scores associated to the utterances, because the NN could not predict the values of the output neurons 2 to 5 as well as the value for the output neuron 1.

Observing the histogram from the n-grams' scores sample sets, we can observe that the majority (by a great amount) of the samples are in the interval [0, 0.02), which makes our ground truth more reliable for this interval. In the other intervals, because there are much less data a simple deviation causes a compromise in the ground truth, affecting the prediction of the NN for these intervals.

Other problem, that might had an impact in the evaluation of the model was how we evaluated the sentences' difficulty. Perhaps, if there was a better way of extrapolating the difficulty score from the utterances' score sample sets, we could had better results in this area.

In future work, we could try to improve the results from the NN. One way of obtaining better results, it would be to better incorporate in the phonemes encoding, external features to the raw input such as: the position of the phoneme in the word, stress, syllable markers… .Other way of obtaining better results it would be to increase the number of n-grams in the hash-table H_1.

References

[1] Deborah Hwa-Froelich, Barbara W Hodson, and Harold T Edwards. Characteristics of Vietnamese phonology. *American Journal of Speech-Language Pathology*, 2002.

[2] Jeesoo Bang and Gary Geunbae Lee. Determining sentence pronunciation difficulty for non-native speakers. In *Speech and Language Technology in Education*, 2013

[3] Scott A Crossley, David B Allen, and Danielle S McNamara. Text readability and intuitive simplification: A comparison of readability formulas. *Reading in a foreign language*, 23(1):84–101, 2011

[4] Scott A Crossley, Jerry Greenfield, and Danielle S McNamara. Assessing text readability using cognitively based indices. *Tesol Quarterly*, 42(3):475–493, 2008

[5] Orphee De Clercq and Veronique Hoste. All mixed up? finding the optimal feature set for general readability prediction and its application to English and Dutch. *Computational Linguistics*, 42(3):457–490, 2016.

[6] Kevyn Collins-Thompson and James P Callan. A language modeling approach to predicting reading difficulty. *In Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 193–200, 2004

[7] Ion Madrazo Azpiazu and Maria Soledad Pera. Multiattentive recurrent neural network architecture for multilingual readability assessment. *Transactions of the Association for Computational Linguistics*, 7:421–436, 2019