Assessing the performance of the TDNN-BLSTM architecture for phoneme recognition of English speech

Irene Klom

June 27, 2021

Abstract

This research studies the Projected Bidirectional Long Short-Term Memory Time Delayed Neural Network (TDNN-BLSTM) model for English phoneme recognition. It contributes to the field of phoneme recognition by analyzing the performance of the TDNN-BLSTM model based on the TIMIT corpus and the Buckeye corpus, respectively containing read speech and spontaneous speech. The TIMIT corpus can be used as benchmark to make comparisons between architectures. The Buckeye corpus is used to better understand how the TDNN-BLSTM architecture would perform on recorded informal conversations. Parameter values are taken from literature and are optimized. Using the improved parameters, the results show Phoneme Error Rates (PER) for read speech to be 31.78% and for spontaneous speech to be 54.03%. Related work shows PER scores for read speech to be 14.9% and for spontaneous speech to be 23.4%. This indicates that the TDNN-BLSTM architecture does not perform as well as other acoustic models for both spontaneous and read speech.

1 Introduction

Automatic Speech Recognition (ASR) translates speech into words. It needs to be able to cope with a large diversity in voices and accents. Every voice is unique and subject to change, for example when a sore throat affects the vocal chords [1].

Phoneme recognition is a form of ASR. While most ASR systems translate speech into words, Phoneme Recognition (PR) systems translate speech into phonemes. A phoneme is "the [smallest] contrasting unit of sound which can be used to change the meaning" of a word [2]. This change in meaning can be illustrated by the words 'hat' and 'cat' and their phonemic difference: /hat/ and /kat/. The change of /h/ into /k/ changes the meaning of the word, therefore /h/ and /k/ are phonemes.

Heteronyms are words that are written the same, but have different meanings and pronunciations. The output of a Word Recognition (WR) system cannot show which pronunciation is used, but this information remains traceable when using a PR system. An example of this is the word 'sake', which can either be pronounced as /ˈseɪk/, meaning 'benefit', or as /ˈsɑːkiː/, meaning 'rice wine'.

WR is different from PR in the sense that instead of having a single phoneme, a word usually consists of multiple phonemes. A lexicon is the dictionary of words the WR system can output. A WR system uses the words in the lexicon to find what word is most likely to be formed by the phoneme sequence. When comparing PR systems to WR systems, it can be established that both have their own advantages and disadvantages. WR systems make

use of a lexicon, but they are therefore also limited by the lexicon available to the model. Although WR research shows word error rates "as low as 2%-3% on standard datasets" [3], WR implementations, such as digital assistants, are not robust yet [3].

The benefits of PR systems can be applied, for example, in online language training tools. A study by Dlaska and Krekeler shows that second language (L2) learners benefit from "individual corrective feedback" [4]. This means that a student's language learning process is impacted when there is no teacher available to give such feedback. An application that makes use of a PR system that can evaluate a student's pronunciation could therefore be beneficial to L2 learners.

A benefit of PR systems is that they are, unlike WR systems, not limited by a lexicon and can therefore recognize all combinations of phonemes. However, PR systems are still bound by the characteristics of the language it is trained on.

The acoustic model is trained for each phoneme in a certain language on large amounts of speech data. The set of phonemes that appears in a language can be different per language. An example of languages with more phonemic differences are tonal languages, where pitch is tied to the meaning of a word. In Mandarin, for example, two words with a different meaning can consist of the same sequence of phonemes: pronouncing these words with a different pitch is what causes the change in the meaning.

1.1 The working of an automatic phoneme recognition system

The automatic phoneme recognition system used in this research translates audio files of speech into phonemes. First, the speech signals are processed into Mel-Frequency Cepstral Coefficients (MFCC) by splitting the audio into frames of about ten milliseconds and extracting the characteristics of the speech in these frames into feature vectors. These feature vectors are the input for training the acoustic model and the language model. The acoustic model learns the probability of observing the acoustics given a sequence of phonemes, i.e. $P(O|phoneme_1, phoneme_2, ...phoneme_n)$. The language model contains the probability of generating a phoneme sequence in the language, i.e. $P(phoneme_1, phoneme_2, ..., phoneme_n)$. Multiplying these probabilities gives the probability of these audio frames being that sequence of phonemes: $P(phoneme_1, phoneme_2, ..., phoneme_n|O)$. The full formula is shown below in Equation 1. In order to recognize new speech, the speech signals are first turned into feature vectors. Then, the trained acoustic and language model are used to compute the most probable sequence of phonemes. The language model captures what phoneme sequences are likely to exist, based on the phoneme sequences that are used as training input. When trained on one language, the language model will capture the probability of a phoneme sequence to appear in that language.

$$P(ph_1, ph_2, ..., ph_n|O) = P(O|ph_1, ph_2, ...ph_n) * P(ph_1, ph_2, ..., ph_n) \quad (1)$$

1.2 TDNN-BLSTM model

This research applies the Projected Bidirectional Long Short-Term Memory Time Delayed Neural Network (TDNN-BLSTM) model on phoneme recognition of American English speech. It aims to evaluate the performance of the model both quantitatively and qualitatively. This evaluation can be used to assess whether the TDNN-BLSTM has the potential to work well for English speech recognition. Two databases are used to analyze the performance of TDNN-BLSTM architecture: The TIMIT Acoustic-Phonetic Continuous Speech Corpus

and The Buckeye Corpus of Conversational Speech [5, 6]. These corpora contain respectively read speech and spontaneous speech. In the rest of this paper, these will be referred to as the TIMIT corpus and the Buckeye corpus.

The TDNN-BLSTM architecture is used to train the acoustic model. The TDNN-BLSTM is a hybrid acoustic model consisting of TDNN layers and BLSTM layers. The BLSTM model is a bidirectional model, which means that it takes previous, current, and future input into account. The TDNN is a network that takes the current and a predetermined number of previous and future inputs into account. The TDNN model is especially good at capturing the short-term context of an input, whilst the BLSTM model is good at capturing long term temporal dependencies [7, 8]. Combining these models gives the TDNN-BLSTM model the ability to balance the importance of long short-term context and short-term context [8].

1.3 Related work

This research builds upon recent work done by Levenbach on Dutch PR [9]. In Levenbach's research, the performance of four acoustic models on two different Dutch corpora was compared. It showed that the TDNN-BLSTM model was performing well. The TDNN-BLSTM model reached a Phoneme Error Rate (PER) of 5.71% on a Dutch clear speech corpus and a PER of 23.42% on a Dutch spontaneous speech corpus [9].

To be able to compare the results of the TDNN-BLSTM architecture to other architectures, these architectures have to be discussed. The first ASR systems were designed in the 1950s and were made to only predict a handful of different speech sounds [10]. Using the TIMIT corpus as a benchmark, a comparison between the TDNN-BLSTM architecture and other architectures can be drawn [11]. The Buckeye corpus, on the other hand, helps us gain a better understanding of how the TDNN-BLSTM architecture would perform on recorded informal conversations, but fewer comparative results are available. Currently, the Li-GRU architecture can achieve a PER score of 14.9% on the TIMIT corpus [12]. This is, to this researcher's knowledge, the lowest PER score on the TIMIT corpus to this date. Although the Buckeye corpus contains spontaneous speech, which is closer to situations outside of a controlled environment, this corpus is used less often for assessing the accuracy of phoneme recognition systems than the TIMIT corpus.

In a research done by Qader et al. [13] a 23.4% PER score is achieved. Since the Buckeye corpus does not have a standard way of preparing data, like the TIMIT corpus does, the data preparation, and thus the results, differs per research. This makes comparisons more difficult.

2 Research question

This research aims to answer the following question, in order to compare the TDNN-BLSTM architecture with other phoneme recognition architectures:
How does the TDNN-BLSTM architecture perform on English read and spontaneous speech?
The following sub-questions are answered in order to answer the main research question:

- What PER can be achieved with training and testing the TDNN-BLSTM model on the TIMIT corpus?
- What PER can be achieved with training and testing the TDNN-BLSTM model on the Buckeye corpus?

- What phonemes have a large PER difference between read and spontaneous speech?

By answering these research questions, a better understanding of the effectiveness of the TDNN-BLSTM architecture for phoneme recognition can be achieved. One question this research aims to answer is whether the TDNN-BLSTM model is generally a good architecture or whether it performs well specifically on Dutch. The combination of the results of the TDNN-BLSTM evaluation on Dutch and English are not enough to generalize any claims about phoneme recognition in all languages. It contributes to the generalization of claims about the network, but does not provide answers to this question.

3 Methodology

This research focuses on the preparation of the corpora and the configuration of the TDNN-BLSTM model. The preparation of the corpus data has been done in collaboration with colleague Georgi Genkov. The tuning of the configuration focused on layer dimensions, on the number of epochs, and on the learning rate of the model. At the beginning, five runs were done with the same configuration in order to estimate the variance. After the same configuration had been run five times, the PER scores were all within 0.58% of each other. Whenever PER scores were close to the minimum PER score of that tuning step, the configurations were rerun to see what PER score is the lowest over two separate runs.

3.1 Corpora

To analyze the performance of TDNN-BLSTM, two databases were used. The TIMIT Acoustic-Phonetic Continuous Speech Corpus and The Buckeye corpus of conversational speech were chosen to represent speech [5, 6]. These corpora respectively contain read speech and spontaneous speech.

The TIMIT corpus is an American English continuous speech corpus with 6300 sentences [5]. These sentences come from 630 speakers who each spoke ten sentences. The data in the TIMIT corpus has a predetermined training and testing division. This research used this advised division in order to ensure that TIMIT can be used as a benchmark and to maximize the validity of the comparisons. The training set consists of 462 speakers of whom 70% are male and 30% are female. The core test set contains 24 speakers of whom 67% are male and 33% are female. The set contains speakers from eight dialect regions in the United States. Two male and one female participant from each region make up this test set. These 24 speakers each spoke eight sentences. All phonemes appear both in the training set and in the test set. Instead of having a continuous flow of input data, the audio is split into chunks. For TIMIT, each sentence is a separate chunk.

The Buckeye corpus is a 307.000-word spontaneous speech corpus [14]. There are 40 speakers who had 30 to 60 minute interviews. All speakers were native to Central Ohio and were middle-class Caucasians. There was an even distribution in gender. The two interviewers, one male and one female, each conducted half of the interviews. Each interviewer spoke with half of the male and half of the female participants. This corpus has no predetermined training and test subdivision. For this research, speakers one to six have been selected. Speakers four and five form the test set, the other speakers form the training set. The training set contains a young male, an old male, a young female and an old female. The test set contains a young female and an old female. Together, these four speakers of the

training set spoke 2876 utterances. The audio and phoneme transcriptions had to be cut into chunks of less than ten seconds in order to let the acoustic model process it effectively. This study has defined these chunks based on non-phonetic audio markers. Non-phonetic audio markers are, for example, silences or vocal noises such as sighing, sniffing, or clearing the throat.

Both corpora have been labeled using the TIMIT labeling guidelines and the Defense Advanced Research Projects Agency (DARPA) phonetic alphabet. Furthermore, it is important to note that although the audio has been carefully transcribed into phonemes and verified by comparing multiple transcriptions, these corpora can still contain transcription errors.

3.2 Network configuration

The TDNN-BLSTM model contains many parameters that can be tuned to alter the outcome. The dimensions of the layers, as well as the number of epochs and the learning rate, have been tuned to reach the lowest possible PER.

The TDNN-BLSTM network is made up out of four TDNN layers, then a BLSTM layer and afterwards two more TDNN layers. This is visualized in Figure 1.

Because this research uses of smaller datasets than the training set consisting of 200.000 utterance used by Levenbach, it focuses on reducing the complexity of the network. The main reductions within the network are done on the number of dimensions within both the non-linear and the linear layers. Furthermore, experimentation with the training epochs and the learning rate is done to see if improvements are possible.

Figure 1: TDNN-BLSTM model

3.3 Kaldi

For this research, the Kaldi speech recognition toolkit was used [15]. Kaldi was instrumental in preparing the data, training the TDNN-BLSTM model, and decoding the test data. It

offers scripts for most standard techniques used in ASR [15].

The TDNN-BLSTM model was trained and tested on the corpora using the TU Delft's student access to the HPC cluster.[1] For this research, the cluster gave access to two CPU cores and a maximum runtime of four hours.

3.4 Evaluation metric

To evaluate the performance of the model, the Phoneme Error Rate (PER) was used. The PER is a metric based on the Levenstein Distance metric, which determines the distance between two strings. It gives an error rate based on the phonemic difference between the ground truth provided by the corpus and the phonemes predicted by the model. The PER is calculated by taking the minimum number of insertions, deletions and substitutions needed to make the predicted phoneme sequence equal to the ground truth and dividing this by the total number of labeled phonemes in the ground truth. Lower PER scores are considered to be better than higher PER scores. The PER formula can be seen in Equation 2.

$$PER = \frac{\#Insertions_{total} + \#Deletions_{total} + \#Substitutions_{total}}{\#GroundTruth_{total}} \tag{2}$$

3.5 BLSTM parameters

This research made use of the same parameter values that Levenbach used in their research for initial testing of the model [9, p. 32].

- 3 BLSTM layers [7]
- 1024 cells per BLSTM layer [16]
- Equal [number] of cells per layer [17]
- Projected recurrence and non-projected recurrence: 256 [18]
- Dropout schedule: 0,0@0.20,0.3@0.50,0 [19]
- L2 regularization: 0.00005 [20]
- Mini-batch: 128 [17]
- 6 epochs [9]

[1] http://insy.ewi.tudelft.nl/content/hpc-cluster

4 Results

This section discusses the results for each tuning step. After the data was prepared, the TDNN-BLSTM model was run on the TIMIT and Buckeye corpora. The PER results of running the model will be given in this section. First, the PER results of the optimization of respectively the TIMIT and the Buckeye corpora are discussed. Then, the confusion matrices of the best runs are provided, in order to analyze how TDNN-BLSTM performs on read versus spontaneous speech.

4.1 Optimization of TDNN-BLSTM for TIMIT

In order to be able to compare the PER scores of TIMIT with Dutch speech, this research first evaluates the BLSTM configuration. Using the BLSTM configurations of Levenbach as shown before on the TIMIT corpus yielded a PER of 35.25%. This is comparable to the 35.82% PER score of BLSTM on Dutch [9, p. 38]. Since the TIMIT corpus contains read speech and Levenbach did not study read speech, no direct comparisons can be made. It can be seen that BLSTM configurations for TIMIT (35.25%) yield a higher PER than BLSTM on Dutch clear speech (11.75%), but a lower PER than BLSTM on Dutch spontaneous speech (43.12%). This is plausible since read speech is considered more clear than spontaneous speech, but less clear than clear speech.

Then, the TDNN-BLSTM architecture is evaluated. The optimizations for the TDNN-BLSTM network were done on the TIMIT model as described in subsection 3.2. All changes made to this model are discussed below.

Table 1 shows the PER scores for the tuning with different layer dimensions. The linear layer dimensions range from 16 to 64 and the non-linear layer dimensions range from 32 to 256. Only powers of two are considered as layer dimensions. The X in the figure means that the configuration is not possible, due to the non-linear layer dimension needing to be at least two times the linear layer dimension. Table 1 shows that the original configuration of a linear layer dimension of 64 and a non-linear layer dimension of 256 is already close to optimal. Slightly reducing the dimensions helps, but larger reductions do result in higher PER scores. The table shows that having 32 linear layers and 128 non-linear layers gives a PER score of 33.44%. The tuning continues with a linear layer dimension of 32 and a non-linear layer dimension of 128.

Lin. Lay. Dim. \ Lay. Dim.	32	64	128	256
16	37.93	35.30	33.65	34.04
32	X	33.94	33.44	33.90
64	X	X	33.47	33.64

Table 1: (TIMIT) PER results in percentages with different (linear) layer dimensions

Table 2 shows the influence of the number of epochs used while training the model. It contains the PER scores for runs with three to seven training epochs. The number of iterations was automatically adjusted. The original value was six epochs and in Table 2 it can be seen that this gives a PER score of 33.44%. This is also the lowest PER score in this tuning step. Increasing the number of epochs to seven leads to the same PER score

as for six epochs, but increases the training time. Therefore, the tuning continues with six training epochs.

Train Epochs	PER
3	35.07
4	34.54
5	33.87
6	33.44
7	33.44

Table 2: (TIMIT) PER results in percentages with different number of epochs

The final tuning is done with the learning rates. The learning rates are split into an initial learning rate and a final learning rate. The initial learning rate is larger than the final learning rate to make large improvements early on and to approach the local minimum more precisely. The outcome of the tuning can be found in Table 3. It shows the PER scores for initial learning rates varying from 0.0005 to 0.1 and final learning rates varying from 0.00005 to 0.05. Since it is not desirable to have a final learning rate larger than the initial learning rate, these cells have no results and are marked with an X. The run with the initial learning rate 0.01 and final learning rate 0.05 was run accidentally. Nevertheless, this result was still included, for the sake of completeness, as X/33.10. The lowest reached PER score was 31.78%, which was achieved with an initial learning rate of 0.1 and a final learning rate of 0.0005.

Inital lr. \ Final lr.	0.00005	0.00010	0.00050	0.00100	0.00500	0.00100	0.05000
0.0005	33.44	33.18	32.67	X	X	X	X
0.0050	32.72	32.67	32.47	32.07	32.61	X	X
0.0100	32.27	32.46	32.77	31.91	33.33	33.39	X/33.10
0.0500	32.10	32.50	32.49	32.24	32.81	32.75	32.39
0.1000	32.22	32.42	31.78	32.64	31.95	32.60	32.71

Table 3: (TIMIT) PER results in percentages with different learning rates

4.2 Optimization of TDNN-BLSTM for Buckeye

Using the BLSTM configurations of Levenbach as shown before on the Buckeye corpus yielded a PER of 57.26%. This is higher than the PER 43.12% for BLSTM on Dutch spontaneous speech [9, p. 38]. A reason for the higher PER score can be the different sizes of the training sets: Levenbach used a training set of 200.000 utterances [9, 37], whereas this research used 2876 utterances for training to reduce training time.

The TDNN-BLSTM configurations are tuned to get a lower PER. In Table 4 the results for the layer dimension optimization can be seen. The same configurations as for the TIMIT layer dimension optimization are used. The combination of 16 linear layers and 32 non-linear layers timed-out two times after four hours. This is indicated in Table 4 with XX. Since the results of 64 linear layers and 256 non-linear layers was very close to the result of 16 linear layers and 256 non-linear layers, these configurations have been run a second time.

The mean of these two runs is used in the table and is indicated with an orange cell. The tuning continues with 16 linear layers and 256 non-linear layers.

Lin. Lay. Dim. \ Lay. Dim.	32	64	128	256
16	XX	56.27	55.41	54.37
32	X	55.37	55.37	55.64
64	X	X	57.26	54.73

Table 4: (Buckeye) PER results in percentages with different (linear) layer dimensions

For the step of tuning the number of epochs, the values three to seven were chosen. From this, seven epochs gave the best result, with a PER score of 54.05. Therefore, one more epoch was added to see if the accuracy would further increase. Training with eight epochs turned out to yield a PER score of 54.03% and tuning therefore continued with eight epochs. The results for this tuning step can be seen in Table 5.

Train Epochs	PER
3	55.23
4	54.56
5	54.14
6	54.73
7	54.05
8	54.03

Table 5: (Buckeye) PER results in percentages with different number of epochs

Since the training and decoding of Buckeye takes longer than that of TIMIT, it is attempted to do an epoch reduction for the decoding step. The number of test epochs is reduced from 12 to eight, six and four. The result of this optimization is that four epochs is enough for the decoding step. Table 6 shows the full results. The lowest reached error rate remains at 54.03%.

Test Epochs	PER
4	54.03
6	54.03
8	54.03
12	54.03

Table 6: (Buckeye) PER results in percentages with different number of epochs

4.3 Individual PER differences between TIMIT and Buckeye

For the run with lowest PER scores per corpus, a confusion matrix is made to illustrate the errors that are made. The matrices in Appendix A and Appendix B show how many times the actual phoneme was deleted, had to be inserted, or was confused with other phonemes. Each row represents an actual phoneme and the columns represent the phoneme that was

predicted. When looking at the individual contributions to the error rates for the TIMIT corpus, it can be discovered that 430 out of the 2293 errors are deletions of silences or substitutions with silences. Furthermore, 80 errors are related to 'epi', which stands for epenthetic silence. This means that 22% of the phoneme errors relate to silences.

Overall, the recognition of TIMIT is better than that of Buckeye. When comparing the correct predictions of TIMIT to the correct predictions of Buckeye, the phoneme recognition percentages for TIMIT are, apart from those for 'ih' and 'sh', all higher.

The 'ih' phoneme has a TIMIT recognition value of 60% and a Buckeye recognition value of 60%. The higher value for Buckeye comes from the difference between phoneme sets between the corpora. The Buckeye corpus does not contain the 'ix' sound, whilst for TIMIT 16% of the errors for 'ih' comes from the confusion with 'ix'.

The 'sh' phoneme has a TIMIT recognition value of 85% and a Buckeye recognition value of 87%. This does not have an apparent reason, but it does have the highest percentage of insertions.

Furthermore, there are confusions of phonemes that are quite similar. For example the phonemes 'z' and 's'. For TIMIT 6% of 'z' phonemes were recognized as 's'. For Buckeye, this number is 12%. Buckeye has a higher percentage of confusions, which is to be expected, since spontaneous speech differs more often from the original phonemes in a word than read speech.

5 Responsible Research

During this research, two corpora were used: TIMIT and Buckeye. For both corpora, the TU Delft licence was used. The Buckeye corpus is free for non-commercial use and the TIMIT corpus has paid licences available. The TIMIT corpus dates to 1993 and the second release of the Buckeye corpus is from 2007.

When studying the inclusiveness of these corpora, it is important to take into account that both corpora do limit the variety of voices that is being trained on. All participants were over 18 years old, which means that the voices of children and teenagers are not represented in the training corpus and can therefore cause higher error rates when tested on.

The TIMIT corpus divided its participants up into eight dialect regions and it gives information about gender of the participants. Additional information about the participants is not available. This makes it hard to determine any potential bias towards elderly people or to determine whether there is any bias relating to socio-economic background of participants.

It must be noted that the TIMIT corpus has 30% female participants and 70% male participants. However, since some research has shown that ASR systems are better at recognizing female voices than male voices [21, 22], this inequality in percentages would be beneficial to lowering the difference between error rates. It must also be pointed out that not all studies agree upon this matter and some research shows increased error rates for females when comparing male and female error rates [23].

The Buckeye corpus also divides its participants by gender and makes a division between 'young' and 'old' participants. The researchers used speakers between 18 and 40 years old to make up the group of 'young' participants and speakers over 40 years old to make up the group of 'old' participants. There is no information about the upper bound of the age of 'old' participants and a more precise indication of the ages is not given. While the Buckeye corpus is balanced in terms of gender of participants and some balance is shown in terms of age of participants, the corpus is not diverse in socio-economic background and dialect region. The Buckeye corpus has only middle-class Caucasian participants who are native to

Columbus, Ohio [14]. This causes speech recognition on speech outside of the middle-class Caucasian subgroup to become a scientific gap. More diverse corpora are advised.

While both corpora have made decisions to ensure diversity between speakers, these corpora on their own are still limited in representation. During this research, the aim is to find a minimum PER for the TDNN-BLSTM architecture to see how well the model performs. It must not be forgotten that a low PER for the entire corpus does not automatically mean that the architecture performs equally well when subdividing data by gender, age, socio-economic background or dialect region. When aiming for an overall minimum PER, it can happen that a PER for a specific subgroups turns out to be higher. For any use of phoneme recognition systems, it should be determined whether to aim for a minimum PER globally or for a balanced PER when looking at results of subgroups. For research with large real-world implications it is advised to use a diverse training set and to aim for a balanced PER rather than a minimum PER globally. Since this research has a limited real-world influence, the choice was made to aim for a minimum PER globally without taking differences between results of subgroups into account.

The results of this paper should be reproducible using the procedures described in the methodology and experimental setup. Of course, the results can differ slightly due to the non-deterministic character of the TDNN-BLSTM training process. Before tuning, the same configuration was run five times to see how much variance there is, the PER scores were all within 0.58% of each other. When values were close to the lowest PER value for that tuning step, these configurations were rerun to improve the accuracy of the result. The mean of these runs is used in the result section.

6 Discussion

The results found for TDNN-BLSTM on TIMIT are not near those for the current best architecture. Li-GRU outperforms TDNN-BLSTM by 16.88%. The model can be further optimized, but it is unlikely that these optimizations will result in a PER score below 14.9%. Using TIMIT as benchmark, it can be established that there are at least 15 architectures that perform better on read speech. The results for spontaneous speech are also lower than the 23.4% PER reached by Qader et al. on Buckeye [13].

When comparing the results to the parallel research carried out by Genkov [24], the TDNN-BLSTM model is slightly outperformed by the TDNN-OPGRU model. The PER score of TDNN-OPGRU is 30.82% for TIMIT and 51.1% for Buckeye.

Two comparable studies for TDNN-BLSTM and TDNN-OPGRU have been performed by Chiroşca and Van der Tang on two Mandarin corpora [25, 26]. These studies combined show that TDNN-OPGRU outperforms TDNN-BLSTM when trained and tested on Mandarin corpora.

Because of the limited time and computational power available during this project, the choice was made to train and test on only a portion of the Buckeye corpus instead of on the full corpus. As a consequence, the parameters shown in this paper could lead to different error rates when applied to the full corpus.

In Appendix B it can be seen that there are more phonemes than appear in the DARPA phonetic alphabet. These phonemes do not appear in the set of expected phonemes, but do appear in the set of predicted phonemes. Since most of these phonemes seem to be a variation to phonemes within the DARPA phonetic alphabet, e.g. ‘aa’ and ‘iy’ are in the DARPA phonetic alphabet and ‘aan’ and ‘iyn’ are not, a merge of these phonemes with

their variation might be possible. The Buckeye corpus documentation did not elaborate on the meaning of these variations and the validity of this merge can therefore not be assessed. Therefore, they are not removed in this research.

6.1 Future work

Future work should include lowering the number of errors made with silences, since these kind of errors were present in high numbers during this research. If the full potential of the TDNN-BLSTM network is to be analyzed, future work should include further optimization of the model's parameters. Tuning with different numbers of TDNN or BLSTM layers or different order of layers is not investigated in this study, but it would be interesting to see what influence this can have on the PER score. As the section about responsible research pointed out, the current corpora are limited in representation. Therefore, it would be interesting to research the validity of the current PER scores on corpora with a more diverse group of participants.

7 Conclusion

In conclusion, this research studied the performance of the TDNN-BLSTM architecture on English read and spontaneous speech. By preparing the TIMIT and Buckeye corpora and optimizing the TDNN-BLSTM configuration, the performance of the TDNN-BLSTM model is assessed. The results show that the TDNN-BLSTM achieves a PER score of 31.78% on read speech and a PER score of 54.03% on spontaneous speech. These PER results are not as low as those of current state-of-the-art architectures. The TDNN-BLSTM architecture currently seems to be the best model for Dutch phoneme recognition, but not for English phoneme recognition. When comparing the results to the research done by Genkov [24], Chiroşca [25] and Van der Tang [26], it can be concluded that the TDNN-OPGRU architecture performs better than the TDNN-BLSTM network on both spontaneous and read speech in English and Mandarin.

8 Acknowledgement

I would like to thank Dr. Odette Scharenborg and Dr. Siyuan Feng for their feedback and support during this process. I would like to thank Jitske van der Laan for her support during the last years, but especially during the last month. I thank my parents for always believing in me. Lastly, I want to express my gratitude towards Elise Klom and Rover van der Noort for supporting me and for reviewing my drafts.

References

[1] M. Islam, "Frequency domain linear prediction-based robust text-dependent speaker identification," 2016.

[2] B. Collins and I. M. Mees, *The Phonetics of English and Dutch*. Koninklijke Brill NV, 2003.

[3] P. Szymański, P. Żelasko, M. Morzy, A. Szymczak, M. Żyła Hoppe, J. Banaszczak, L. Augustyniak, J. Mizgajski, and Y. Carmiel, "WER we are and WER we think we are," 2020.

[4] A. Dlaska and C. Krekeler, "Self-assessment of pronunciation," *System*, vol. 36, pp. 506–516, Dec. 2008.

[5] J.S. Garofolo, L.F. Lamel, W. M. Fisher, D. S. Pallett, N.L. Dahlgren, V. Zue, and J.G. Fiscus, "Timit acoustic-phonetic continuous speech corpus," 1993.

[6] M.A. Pitt, L. Dilley, S. K. Johnson, W. Raymond, E. Hume, and E. Fosler-Lussier, "Buckeye corpus of conversational speech (2nd release)," 2007.

[7] G. Cheng, D. Povey, L. Huang, J. Xu, S. Khudanpur, and Y. Yan, "Output-gate projected gated recurrent unit for speech recognition," in *Interspeech 2018*, pp. 1793–1797, ISCA, Sept. 2018.

[8] S. Feng and T. Lee, "Improving cross-lingual knowledge transferability using multilingual tdnn-blstm with language-dependent pre-final layer," in *Proc. Interspeech 2018*, pp. 2439–2443, 2018.

[9] R. Levenbach, "Phon times: Improving dutch phoneme recognition," Master's thesis, 2021.

[10] B. Juang and L. Rabiner, "Automatic speech recognition - a brief history of the technology development," 2005.

[11] H. van Geffen, M. Smit, A. Warners, F. Warners, and T. Yarally, "A review of deep neural network-based phoneme recognition systems." 2020.

[12] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, "Light gated recurrent units for speech recognition," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, pp. 92–102, Apr. 2018.

[13] R. Qader, G. Lecorvé, D. Lolive, and P. Sébillot, "Probabilistic Speaker Pronunciation Adaptation for Spontaneous Speech Synthesis Using Linguistic Features," in *International Conference on Statistical Language and Speech Processing (SLSP)*, (Budapest, Hungary), pp. 229–241, Nov. 2015.

[14] S. Kiesling, L. Dilley, and W.D. Raymond, *The Variation in Conversation (ViC) Project: Creation of the Buckeye Corpus of Conversational Speech*. Ohio State University, 2006.

[15] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hanneman, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," 2011.

[16] T. Niedek and T. Heskes, "Phonetic classification in tensorflow." 2016.

[17] A. Mohamed, G.E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, pp. 14–22, Jan. 2012.

[18] G. Cheng, D. Povey, L. Huang, J. Xu, S. Khudanpur, and Y. Yan, "Output-gate projected gated recurrent unit for speech recognition," in *Proc. Interspeech 2018*, pp. 1793–1797, 2018.

[19] G. Cheng, V. Peddinti, D. Povey, V. Manohar, S. Khudanpur, and Y. Yan, "An exploration of dropout with LSTMs," in *Interspeech 2017*, ISCA, Aug. 2017.

[20] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," 2014.

[21] M. Adda-Decker and L. Lamel, "Do speech recognizers prefer female speakers?," pp. 2205–2208, Jan 2005.

[22] S. Feng, O. Kudina, B. M. Halpern, and O. Scharenborg, "Quantifying bias in automatic speech recognition," 2021.

[23] R. Tatman, "Gender and dialect bias in YouTube's automatic captions," in *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, (Valencia, Spain), pp. 53–59, Association for Computational Linguistics, Apr. 2017.

[24] G. Genkov, "Training and testing the TDNN-OPRGU acoustic model on English read and spontaneous speech." 2021.

[25] M. Chiroşca, "Evaluating the performance of the TDNN-BLSTM on Mandarin read and spontaneous speech." 2021.

[26] J. van der Tang, "Evaluation of phoneme recognition through TDNN-OPGRU on Mandarin speech." 2021.

A Normalized confusion matrix of TDNN-BLSTM on TIMIT

	aa	ae	ah	ao	aw	ax	ay	b	ch	cl	d	dh	dx	eh	el	en	er	ey	f	g	hh	ih	ix	iy	jh	k	l	m	n	ng	ow	oy	p	r	s	sh	t	th	uh	uw	v	w	y	z	zh	epi	vcl	sil	del	ins
aa	0.75													0.01														0.03		0.02				0.01																0.12	0.02
ae	0.04	0.8			0.05	0.07		0.01	0.06					0.06														0.01																						0.01	0.01	0.06	0.01
ah	0.05	0.01	0.6	0.01		0.1	0.01			0.01				0.08													0.01	0.01				0.06		0.01				0.01	0.01								0.01	0.01	0.03	0.01
ao	0.12		0.03	0.7	0.01	0.02				0.01							0.01	0.01									0.01					0.02		0.02	0.03												0.09	0.03
aw	0.05	0.05	0.03	0.03	0.68	0.03																					0.11							0.03														
ax			0.04			0.55	0.01						0.03	0.02	0.01	0.02						0.01	0.21	0.01			0.01			0.02				0.01		0.03	0.01	0.01									0.01	0.01	0.14	0.08
ay	0.07	0.01	0.02	0.01			0.78						0.01										0.01	0.01			0.01	0.01	0.01		0.01	0.01		0.01												0.06	0.03
b								0.92		0.01																																						0.01	0.03	0.01
ch									0.74																0.05							0.03		0.05	0.07	0.02											0.02	0.05	0.02
cl			0.0				0.01			0.9	0.0																																				0.0	0.04	0.03	0.06	0.02
d											0.72	0.05	0.05		0.01			0.0		0.0	0.0	0.0	0.01	0.01	0.0	0.01				0.01		0.01	0.0												0.01	0.05	0.1	0.07
dh	0.01										0.04	0.79	0.01					0.02		0.01	0.01		0.01			0.04	0.01	0.01							0.03											0.01	0.11	0.02
dx	0.01										0.03	0.04	0.8							0.01	0.02		0.07	0.04			0.01	0.01		0.06				0.01												0.02	0.05	0.03
eh		0.07	0.05		0.01	0.01				0.01				0.69																																0.01	0.11	0.01
el			0.02			0.05									0.68												0.14		0.02			0.02														0.07	0.05	
en											0.04					0.52							0.04																							0.16	0.08	
er		0.0	0.01		0.03								0.01		0.0	0.78	0.0					0.02	0.04						0.01		0.04	0.16			0.05	0.0		0.0	0.01					0.04		0.03	0.04	
ey		0.02				0.02				0.03			0.03			0.01	0.81	0.0				0.02	0.01	0.03						0.01														0.01	0.01	0.03	
f											0.01		0.01			0.01	0.0	0.87				0.01	0.01																								0.01	0.01	0.02	
g												0.06	0.02					0.01	0.81	0.02					0.06																							0.02	0.12	0.03
hh	0.02	0.02											0.02						0.01	0.87	0.02											0.02																0.05	0.1	0.03
ih			0.01		0.02					0.01				0.07		0.01	0.02			0.01	0.01	0.6	0.02																									0.08	0.02
ix		0.01	0.01		0.09					0.01	0.0			0.01	0.0	0.03	0.0	0.0		0.0	0.0	0.1	0.68	0.03				0.01	0.01	0.01		0.01	0.0	0.01											0.01	0.0	0.12	0.03
iy														0.0		0.0	0.01				0.0	0.02	0.04	0.88																							0.0	0.0	0.03	0.0
jh								0.05			0.01		0.01							0.01	0.01	0.01			0.95																							0.03		
k										0.0	0.0	0.01														0.9	0.8	0.01	0.01			0.01	0.01	0.01		0.01										0.03	0.08	0.03
l				0.0	0.0					0.0	0.01			0.07		0.0	0.0				0.01	0.01	0.0	0.03			0.02	0.86	0.05	0.01																	0.05	0.1	0.03
m					0.01						0.01					0.01											0.02	0.86	0.05	0.01																	0.08	0.02
n				0.0						0.01	0.01	0.01				0.02	0.0				0.0	0.02	0.04	0.88			0.0	0.04	0.85	0.02																	0.02	0.11	0.01
ng										0.02				0.02		0.02					0.01	0.0						0.17	0.67																		0.02	0.07	0.0
ow		0.05	0.05	0.01	0.03		0.01						0.06				0.02				0.02		0.07			0.07			0.61	0.01	0.01	0.01														0.04	0.04	
oy																														0.07	0.86																0.01	0.09	
p										0.0						0.08		0.02								0.01	0.02	0.01				0.9	0.01															0.05	0.01
r										0.0												0.01	0.01	0.01	0.0								0.88															0.17	0.03
s			0.0							0.0	0.0		0.0	0.0	0.0					0.0		0.02												0.84	0.01											0.03	0.01	0.01
sh									0.06		0.01	0.05										0.01				0.02	0.02								0.85											0.02	0.02	0.12
t									0.01		0.03	0.06																								0.83	0.02									0.01	0.07	0.01
th		0.03																								0.09	0.07										0.6									0.09	0.09	0.06
uh			0.07		0.07							0.07						0.06									0.02											0.71								0.07	
uw																						0.05	0.05	0.02			0.02											0.83							0.02	0.08	0.02
v							0.01				0.05		0.01				0.02	0.05				0.01					0.02	0.02	0.01											0.75	0.01					0.01	0.09	0.01
w		0.01					0.03																				0.02	0.01			0.01									0.92					0.01	0.08	0.02
y											0.03			0.03			0.03			0.05	0.03		0.05											0.06							0.8				0.03	0.3	0.07
z											0.01	0.01																														0.85			0.08	0.04	
zh																																												0.12	0.75		0.12		
epi		0.01	0.0		0.01		0.02		0.01	0.0	0.01	0.0	0.01																																0.11	0.0	0.72	0.08	0.09
vcl		0.01								0.07	0.0	0.01		0.0	0.0		0.01			0.02	0.0	0.0	0.01	0.0		0.0	0.01	0.01	0.01	0.01		0.0	0.01	0.01	0.01	0.0	0.01	0.0	0.01	0.0	0.0	0.01	0.0	0.0	0.0	0.01	0.82	0.04	0.1	0.03

B Normalized confusion matrix of TDNN-BLSTM on Buckeye