Developing a speech recognition system for isiXhosa

Yuanyue You

Master of Science
School of Informatics
University of Edinburgh
2021

Abstract

The main purpose of this project is to build an Automatic Speech Recognition(asr) system to recognize isiXhosa, a low-resource South African language. This project compares the recognition performance of isiXhosa by a variety of different speech recognition systems, which mainly includes two sections: GMM training section and DNN training section. The GMM training includes the most basic mono-phone system, triphone system, lda-mllt system and SAT system. In order to further improve the model's recognition performance on isiXhosa, DNN network is also used to train the data, including monolingual system and multilingual system. In this report, this project is to apply a variety of different training models, analyze the decode results and try to find the most suitable model for recognizing the isiXhosa language.

Acknowledgements

In this report, I would like to thank my supervisor, Peter Bell, for his guidance and help. He helped me to have a better understanding of the entire project process. With his help, I can work with my project more smoothly. He mentioned that I need a baseline system, through which the system can be optimized and modified. The main process of this project is carried out in accordance with this idea.

Secondly, I would like to thank Electra WALLINGTON, a PhD student of my supervisor. She helped me a lot in the details of the project. Some program running problems and the preprocessing of raw data are all handled by her. This allows me more time to focus on applying more speech recognition systems and the comparison between different models.

With the help of Peter and Electra, I learned a lot of knowledge about speech recognition system and actual project experience from this project. And I also learned to use Kaldi tools to perform speech recognition work more proficiently. All in all, thanks to them, my project was able to proceed smoothly, and at the same time I was able to gain a lot of knowledge and project experience in speech recognition work.

Table of Contents

Chapter 1

Introduction

1.1 motivation

The current language recognition system is getting increasingly better performance for languages, and these language recognition functions are gradually being applied in our real life, such as the automatic translation function of yotube and adding subtitles on screen, and the voice recognition wizard siri of iphone. But these are for large languages such like English, Chinese, German etc, that is, languages with sufficient training data. But for low-resource languages, the training data and corresponding annotation will be more limited. In this paper, Peter proposed that the multilingual asr system can improve the recognition performance with the help of other languages, so here this project will verify whether the multilingual system has a good recognition performance for low-resource language, Xhosa.

1.2 objective

The language to be recognized in this project is Xhosa, a low-resource language in South Africa. In this project, a traditional GMM model will be built to recognize the Xhosa language, and several attempts were made to optimize the GMM model such as lda-mllt and SAT models. For the DNN model, the traditional TDNN system is also applied for test the performance of Xhosa[1]. Besides, the project will also apply the multilingual system with Xhosa+Zulu and Xhosa+English combination. The project will use Zulu[2] and English as the collocation languages of Xhosa to train the mul-

[1]https://en.wikipedia.org/wiki/Xhosa_language

[2]https://en.wikipedia.org/wiki/Zulu_language

tilingual language model. The reason for using these two languages is to determine whether the similarity of the two languages in the multilingual system will affect the final performance. Finally, through the recognition performance of several systems on the Xhosa language, try to find the best model for recognizing Xhosa, verify that the multilingual model has improved the ability to recognize low-resource languages, and verify whether the language similarity in the multilingual model will affect the recognition performance of multilingual model.

1.3 structure of this report

In this paper, I will first introduce some background theoretical knowledge about this project in the second part so that readers can better understand the whole process of this project. In the third chapter, I will introduce the specific implementation process of the project, including the data introduction, the project structure, and the experimental process. In Chapter 4, I will list the results of the experiment and analyze these results. And analysis whether these results meet expectations. Or if the results do not meet expectations, then try to find the possible reasons for the results. The final conclusion part includes some summary of the project and somethings to reminder if there is more time to improve the recognition ability of the whole system.

Chapter 2

Background and Related Work

2.1 Related Work

Generally speaking, under-resourced languages have limited training data which is difficult to support the model to learn the features of these languages well. However, in these years of research, there have been many researches on low-resourced languages and good progress has been made.

In 1999, Wet et al[5]. presented the first 'state-of-art' model for Afrikaans speech recognition. In this experiment, he first let 40 Xhosa natives record the pronunciation of 2000 words and use these recordings as training data. In addition, for these recordings, there will be a period of silence between each word. After collecting the training data, he used these data to train the basic word model and use the trained model to recognize the test data. The recognition result of the final trained model was very good. Even the recognition performance of male recordings has reached an word error rate(WER) of 6.8, and the recognition performance of female recording test data has WER of 10.1. However, this trained word model is actually not suitable for actual speech recognition, because people's normal speech will not deliberately separate each word with a short time interval. For the training data in this experiment, there will be a short silence between each word, which makes the recognition of the model easier. So at the end, Wet also mentioned in the paper that this was just a context independent model. For the data of normal people speaking, this model will not have good recognition results.

Regarding the multilingual model, Uebler et al.[17] established the bilingual language model earlier. In this experiment, he used Italian and German to train his bilingual language model. And this experiment was also adapted to the speaker's adaptation data with speaker adaptation training(SAT) method. The author used the MAP algorithm to make the bilingual model adapt to the data. In addition, the author also retrained the models specially for the speaker data to observe the improvement of the the model's performance trained with the training data of the specific speaker. This experiment also includes monolingual model, bilingual model, SAT model, speaker independent(SI) model and speaker dependent(SD) model. The SD model is what he retrained for specific speaker's training data. Through the final experimental results, it can be seen that the bilingual model has better recognition performance than that of monolingual model, and the SAT method will still improve the recognition performance of these two models. The best experimental result was got by the SD model trained by specific speaker's training data, which also shows that if there is enough training data, the SD system will have a very high recognition performance.

Gales et al.[8] built ASR and KWS systems for low-resourced languages. In this project, he first used the Tandem model, and tried to combine the tandem models of each language together to form a multilingual model. Throughout the experiment, the author also adjusted the training data, using sufficient training data (80 hours) and limited training data (10 hours) to train the model. Hence, he can observe what kind of influence the sufficient training dataset and limited training data will have on monolingual and multilingual models. From the experimental results, the recognition performance of monolingual model and the multilingual model both trained by sufficient data are very close. However, once the limited training data is used, the multilingual model will greatly improve the recognition performance of the monolingual model. The author used limited training data to simulate some low-resourced languages, then it can also prove that the multilingual model can improve the recognition ability of low-resource languages, even though these languages do not have enough training data.

The project of Zoltan et al.[16] is very similar to that of Mark. He only tried various improvement methods for 10 hours data to improve the recognition performance of the entire system. Among them, MLP was further optimized, and the sigmoid activation in each layer was replaced with a rectified linear layer. These optimization steps not only improve the recognition result of the monolingual model, but also greatly improve the recognition result of the multilingual model.

Peter et al.[3] published a crosslingual model based on German and English which has sufficient training data. This model first applied the multilingual model and used the source language and target language as the input of the multilingual model. And extract a hidden layer in the multilingual model as the bottle neck layer, and use this layer and the target language as the input of a new DNN model. The output of each dnn model is the alignment result of the corresponding language from GMM training. In the experiment, German was used as the target language and English was set as the source language. Experiments show that the crosslingual model has greatly improved the overall recognition performance.

Astik et al.[4] did a multilingual speech model on isiZulu and English. He also tried to find whether the combination of isiXhosa, Setswana and Sesotho languages with English can improve the recognition performance of the English-isiZulu code-switch model. And in this experiment, the author separately evaluated the DNN model and the TDNN-LSTM model. The final result shows that if the two languages are relatively similar and related, the ASR model will be greatly improved. The best model is a multilingual model trained by all 4 combined languages. It can be indicated from this experiment that the multilingual model can learn linguistic knowledge from other language pairs and gain improvement from it. If the language is similar, the learning effect is better. And the more languages, the more content will be learned. These two methods can both improve the recognition performance of the multilingual model.

Thomas et al.[13] did a Phonetic analysis of several South African languages which included Xhosa, Zulu and Afrikanns. And he also compared and analyzed these languages with English. He used AST databases and compared the monophone tokens that appear most frequently in each language, and tried to cover the other three languages using different languages. The result is that the monophone types of Zulu and Xhosa are very similar. As the number of monophone types in Xhosa or Zulu increases, their own fraction curve can well cover the fraction curve of the other, so it shows that at least in monophone types, Zulu and Xhosa are very similar. In addition, the author compared the Triphones of each language. From the statistical results, the triphones of Xhosa and Zulu are less, and the triphone fraction curves of the two have similar growth trends and even the shapes are very similar. Therefore, it can be concluded that the phone composition of both Xhosa and Zulu are very similar. Besides, he also compare the word types through the borrowed word tokens method which show that at the word level, Xhosa and Zulu still have very high similarity. Hence, Thomas compared the four languages in monophone, triphone and word level. And from the result, we can conclude that the Xhosa and Zulu has high similarity.

Through the above background knowledge and related work, it is obvious that if there is enough training data, then the monolingual model will also get a good recognition performance. But if there is not sufficient data, as mentioned in Gales et al.[8]

the monolingual language model will not get as good recognition result as the multilingual language model. This result also shows that it may be a good possible method to train multilingual language mode with low-resourced language. In addition, from Astik's[4] experiment, the more similar the languages are, the greater improvement the multilingual model will get. And in Thomas'[13] analysis, the most similar language of Xhosa is Zulu. Therefore, for this project, Xhosa is a low-resourced language which will have limited training data. Using the multilingual model with Xhosa and Zulu as input language may be a good method. In addition, some other methods can further improve the overall recognition performance, such as the SAT[17] method. They can still be applied into this project for better recognition performance.

Chapter 3

Conceptual Design work

3.1 Xhosa and Zulu

In Thomas' paper[13], the similarity of Afrikaans, Zulu, Xhosa, english is compared, and the number of monophone type, triphone type, word type, unigram language model and bigram language model are also compared. Among them, Xhosa and Zulu have a very high degree of similarity. In order to verify the influence of language similarity on the multilingual network, two languages were selected as the partner language of Xhosa. One of the selected language is Zulu, which is also a low resource language in South Africa and very similar to Xhosa. The other one is English which is not very similar to Xhosa but with sufficient training data. Through the final multilingual results, we can verify the effectiveness of the multilingual model for low-resource languages and also try to explore the influence of language similarity on multilingual.

3.2 ASR models: GMM and DNN

Because this project is to compare a variety of different methods to find the best model for recognize isiXhosa, in this section I will introduce two major aspects of background knowledge and introduce how these methods work in the project. The first section is about showing the conceptual design work of different GMM models, and the second is about DNN training. In the actual project process, the best GMM part result will be used as the label during DNN training, so that the procedure of the whole project is to use different models training is aligning again and again, and the model will gradually have better results.

3.2.1 Monophone and Triphone GMM models

This project uses monophone[15][6] and triphone[11] GMM models to build a basic GMM recognition system. The monophone model is in the GMM model, and each state corresponds to a phone. Generally speaking, the features we extract are 39-dimensional MFCC features, and each 39-dimensional feature vector represents that frame of audio in the original sound file. So in the monophone GMM model, what the model has to do is to align the vector of each frame with the corresponding state, and use these aligned states to iterate through the EM algorithm in the GMM model to find the most likely phones. The final result is to recognize the sound signal of each frame as the corresponding phone. The monophone model is a very basic speech recognition model, because this model has relatively large flaws. This model has a good performance on single word recognition, that is, there is a clear silence interval between each word. But for real speech, the usual speech will have continuous reading, and the interval between each word is not obvious, and the monophone model will have a relatively poor recognition performance.

Therefore, this project adopts the GMM model of triphone. Compared with the monophone model, triphone considers the association between phonemes. In the triphone model, each state represents three connected phonemes. This can better solve the problem of inconspicuous spacing between words in normal voice communication. So as to better improve the recognition performance.

3.2.2 LDA-MLLT

Through the triphone method, the model can also take the phonemes before and after a phoneme into consideration at the same time, so that the model can grasp more data information and the training result will be more accurate. In order to further improve the recognition performance of the overall model and enhance the robustness of audio features, it is necessary to extract feature vectors with strong distinguishing ability. Here the project use the Linear Discriminant Analysis(LDA) algorithm [12], and the feature vector of the training data becomes more discriminative through the LDA algorithm.

lda is actually a feature convert method, that is, a The process of transforming acoustic features into features of another frame through a certain algorithm. As for the converted feature vectors, we generally hope to have stronger robustness. For this project, the feature vector of each frame need to have strong distinction after feature

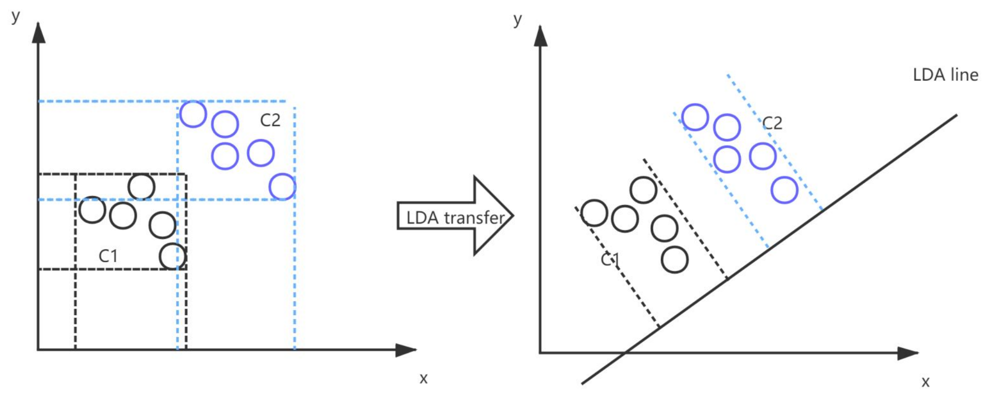


Figure 3.1: lda transfering

convert. lda is a linear discriminant analysis, a supervised learning dimensionality reduction algorithm that provides information through the labels of data categories so that the data points after dimensionality reduction can be distinguished as much as possible. The core idea of the lda algorithm is to find a low-dimension feature space, so that the intra-class variance after the original feature projection is the smallest and the inter-class variance is the largest. As shown in the Figure3.1 , the feature vectors c1 and c2 of the two categories are projected to the x-axis and y-axis, their projections will have overlapping parts, and the projection information cannot distinguish the two features well. If the lda change is applied, the low-dimensional line of lda is obtained, that is Xlda, after c1 and c2 in the figure are projected on the lda surface, the inter-class distance between the two types becomes larger, the intra-class distance becomes smaller.Hence there is more Good differentiation performance. All in all, the goal of lda dimensionality reduction is to project labeled data into a low-dimensional space, and at the same time, it must be satisfied to keep as much data sample information as possible to find the best projection space, and the projected samples are more discriminative . Such an operation method can increase the characterization of the data, and make it easier for the system to learn the information of the data, thereby improving the recognition performance.

However, the covariance matrix after LDA change cannot be diagonalized, so it is necessary to perform MLLT [7] on the obtained data for global feature change. For specific operations, please refer to this paper(cite reference). Through the MLLT, the feature vector obtained in this way can be diagonalized, which has a relatively high degree of discrimination and good robustness.

3.2.3 SAT

The previous training is all based on the data itself, without considering the speaker's information. In reality, due to the speaker's speaking style, accent, and recording environment, different training data will also influence the recognition performance. Assuming that when there is enough training data for a certain speaker, the speaker dependent model can be obtained by using traditional training methods for the data of this speaker, and this model reflects the characteristics of the current speaker well, so the recognition performance is usually will be better. However, in practice, the data of a certain speaker is often very small, so it is necessary to adapt a speaker adaptive method, so that a relatively large performance improvement can be obtained with a small amount of data. The system trained by the training data of many speakers used before is the speaker independent model. The speaker adaptive training (SAT) [1] is to adjust the SI system with adaptive data to meet the characteristics of the current speaker. That is to say, SAT is to shift the trained SI system to the SD system to improve the recognition of the current speaker data ability.

In this project, the feature-space MLLR (fMLLR)[18] method is used to achieve the SAT performance. fMLLR is the conversion matrix obtained by training, so that the adaptation data can obtain the maximum likelihood value in the current model.

fMLLR adapts both means and covariance of the Gaussians by applying an affine (linear) transform of mean parameters.

$$\hat{\mu} = A'\mu - b'$$

$$\hat{\Sigma} = A'\Sigma A'^T$$

The matrix A is estimated through training, and then the parameters of all gaussians are transformed into a speaker adaptive model to improve the recognition performance of a specific speaker.

Up to now, the GMM training part has been introduced. Through consideration of neighboring phonemes, triphone is applied in this project, in order to make the training data more discriminative, and make it easier for the system to learn the characteristics of the data. , Lda-mllt is also used in this project. Finally, the SAT method allows the system to perform offset training on the speaker, thereby further improving the recognition performance. However, the simulation effect of GMM is limited. Generally speaking, the best recognition performance (wer) of GMM is about 20%. If the system still need to improve, the dnn model is necessary to train the data.

3.2.4 TDNN

In this project, the TDNN model[14] is used to train the data. The structure of TDNN is very similar to CNN, and both adopt a hierarchical structure. For TDNN, the data of the current hidden layer is obtained by processing the results of the previous layer through the time context window. Because the lower hidden layer is obtained from a narrower context, it will have a narrower receptive field, and the higher layer will have a wider receptive field for the input. As shown in the figure below, the range of the receptive field increases as the hidden layer increases. In the final output layer, the result will be output through an affine layer. In this project, the input is the feature vector (MFCC vector) of the sound signal, and the output is the HMM state corresponding to the feature vector of each frame. Then after back propagation and constant iteration, the tdnn model is finally trained, and the function of this model is to map the feature vector of each frame to the corresponding HMM state. This model will be used to evaluate the performance of monoligual recognition.

3.2.5 Multilingual DNN Model

If the amount of training data is sufficient, the monolingual tdnn model can have a good recognition performance on the language, and the more complex tdnn structure has the stronger the learning ability. However, when the training data is insufficient, there is not enough training data to train the tdnn model. For example, in this project, a low-resource language isiXhosa is selected as the object language of recognition, but there is not enough training data. Then the tdnn model has no way to learn good features. Therefore, the multilingual model[9][10] is used in this project to solve this problem. And this project is also used to verify whether the multilingual model can solve the problem of lack of training data in small languages.

The Multilingual model is still a tdnn model, but the hidden layer in tdnn learns the expression of multiple languages. These hidden layers are shared by multiple languages, and the output layer of this tdnn is unique to a single language. As shown below. In this way, the hidden layer can be trained with more training data to make up for the lack of low-resource language training data. In the multilingual model, the more similar the language is trained, the better will be the model.

Therefore, this experiment will generally use all the above methods to find the best method for isiXhosa recognition, and verify whether these methods improve isiXhosa language recognition as the theory suggests.

And for evaluation method, this project will use word error rate(WER) to show the performance of the system.

Chapter 4

Experiment

4.1 Data Introduction

In this project, Xhosa is the language that needs to be recognized. At the same time, in the multilingual section, English and Zulu were selected as the partner languages of Xhosa. Therefore, in this project, there are a total of three languages of data to be collected. In this project, the Xhosa, Zulu and English of the NCHLT-clean dataset[1] are used. The detail information about the dataset can be found in this paper[2]. The amount of training data for each language and the various characteristics of these three languages are presented in the Table 4.1. These features include the number of male and female speakers, the total number of utterances, the total duration, word types, word tokens, the number of training sets and test sets.

language	speaker			word		utterance	duration(hh:mm)
	total	male	female	types	tokens		
Xhosa	209	107	103	29130	146904	46651	56:15
Zulu	210	98	112	25650	130866	44673	56:14
English	210	100	110	8351	222884	77412	56:25

Table 4.1: Dataset information for Xhosa, Zulu and English

The stats are shown in the Table 4.1 which include the training and test set. The test set comprise 8 speakers with equal males and females. For each dataset, they all include many audio files and associated nchlt_[language].xml files. The xml files include the information about the audio files including transcription, duration, speaker

[1]https://sites.google.com/site/nchltspeechcorpus/home

id, gender and relative path of audio files.

Based on these audio files and xml files, the project needs also to convert the xml files into kaldi type files which are text, wav.scp, spk2utt, utt2spk and other optional files. The text file include all the utterance-id and corresponding transcription with [utt-id] [transcription] format. The wav.scp include all the utterance-id and their actual path with [utt-id] [actual path] format. The utt2spk and spk2utt files map each utterance with their speaker-id. [utt-id] [speaker-id] format for utt2spk file and [speaker-id] [utt-id] format for spk2utt file.

The detail information about train and test data for each language are shown in the Table 4.2.

dataset	speaker			utterance
	total	male	female	
Xhosa train	201	103	99	43881
Xhosa test	8	4	4	2770
Zulu train	202	94	108	41871
Zulu test	8	4	4	2802
English train	202	96	104	74180
English test	8	4	4	3232

Table 4.2: The dataset information of training set and test set

The dict is also provided in NCHLT dataset which can be downloaded from here. The downloaded dict is only the raw dictionary, it is necessary to convert these dictionaries to the kaldi type dict which includes lexicon, extra_questions.txt, nonsilence_phones.txt, silence_phones.txt, optional_silence.txt and lexiconp.txt files.

4.2 Experiment Procedure

The steps of the whole project are the same as the structure introduced in the background and theoretical knowledge section. There are two parts. The first is the GMM training part, and the second is the DNN training part. This project is modified based on the recipes of babel and babel_multilang in the Kaldi tool. And the experiment procedure follows the pipeline shown in Figure 4.1.

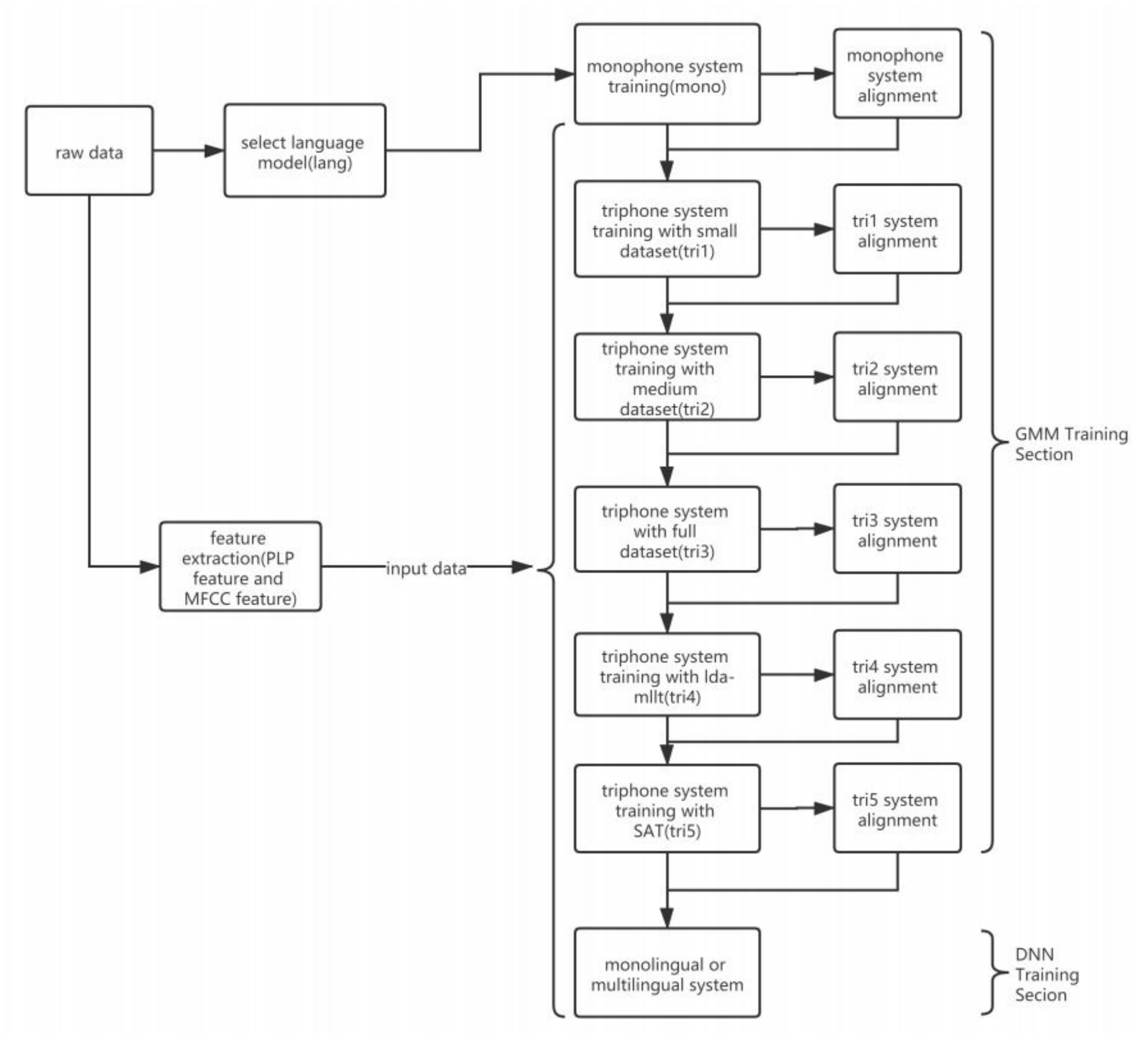

raw data
select language model(lang)
feature extraction(PLP feature and MFCC feature)
input data
monophone system training(mono)
monophone system alignment
triphone system training with small dataset(tri1)
tri1 system alignment
triphone system training with medium dataset(tri2)
tri2 system alignment
triphone system with full dataset(tri3)
tri3 system alignment
triphone system training with lda-mllt(tri4)
tri4 system alignment
triphone system training with SAT(tri5)
tri5 system alignment
monolingual or multilingual system
GMM Training Section
DNN Training Secion

Figure 4.1: Pipeline of the whole project

4.2.1 Feature Extraction and Train Language Model

Language Model
First, convert Xhosa's dict into lexicon and store it in the L.fst file in wfst format. At the same time, when generating lexicon, set the silent phoneme to sil and require all silent phonemes to use this symbol together. In addition, those phones which are not in the vocabulary will be replaced with [unk] symbol. Use such restrictions to generate lexicon. Then, use the srilm tool to calculate the language model of the text information in the training data and select the language model with the lowest perplexity. These language models include the language models of Good-Turing, Kneser-Ney and Maxent's 2gram, 3gram and 4gram. By comparing the perplexity scores of these language models, the language model with the lowest perplexity is selected as the optimal language model for the language, and this model will be used as the basis for the subsequent decoding steps of the system.

Feature Extraction
The process of feature extraction is to convert the original audio file into a frame by frame feature vectors through appropriate operations. Use a sample frequency of 16000 to sample the original audio file. In addition, use window length=25ms and frame shift=10ms to sample the original voice and generate plp feature vectors. In the process of feature extraction, the timbre of the speaker will also be sampled and used as part of the feature so the pitch vector is used to generate the final 16-dimensional feature (13-dimensional plp+3-dimensional ptich). These are just the most basic plp features. The complete PLP feature includes delta and delta-delta for the 13-dimensional basic features. Therefore, the 13-dimensional plp*3+1-dimensional energy+3-dimensional pitch consist of 43-dimensional final plp feature. The 43-dimensional final plp feature is the input dimension of the network in tdnn training.

Then standardize the entire feature, and perform the CMVN operation on it to normalize the mean/variance of each bit of the feature by using $F_k^{cmvn} = (F_k - \mu_k)/\delta_k$. So far, the feature extraction process of the training data is completed.

4.2.2 GMM training section

The entire training process of the speech recognition system is the continuous training, the alignment process. The training is carried out through the previous alignment information. So in order to make the subsequent training more efficient, in this step, the training data is divided into 3 parts, of which the first 5000 training data is used as train_sub1 to provide training data for monophone training. For the first triphone training (tri1), use first 10,000 training data(small data) train_sub2 to train the tri1 system. The second triphone training (tri2) uses the first 20,000 data trian_sub3(medium data) in the overall training data. The last and third triphone training (tri3) uses all training data(full data) for training. Through such data segmentation, the training and alignment process from time to time becomes more effective.

The overall training process is as shown in the flowchart. After each model is trained and aligned, the alignment information will be used as the basis for the next model training. In the entire training process, the parameters required by each model are displayed in the Table 4.3. In this project, for monophone system, only using Gaussian models will get good result. Hence, the number of Gaussian are set 1000 without using decision tree method. For the triphone system, if still only use the Gaussian models, the total triphone state will cost much larger number of Gaussian and cost so much computation. Hence, to reduce the complexity of the method, the decision-tree method is applied at triphone models. With the amount of training data increasing and model getting more complex, the number of leaves and Gaussian also increase shown in the Table 4.3. Use more number of leaves and Gaussian can improve the model's simulation ability so that improve the recognition performance. And after each model training is completed, the language model will be re-estimated based on the current data, language model, dictionary, and trained system. The re-evaluated language model will be used in the alignment of this system and the training of the next system. For example, after the tri2 model is trained, the system will save the results of the re-evaluated language model in the tri2 directory, and this tri2 directory will be used in the alignment(tri2_ali) of the tri2 system and the training process of the tri3 system. This approach can update the old language model after each training is completed, so that the updated language model is more suitable for the current system, and thus has a higher recognition accuracy rate.

When all the systems are trained and aligned, the final training information (tri5) and its information (tri5_ali) will be obtained, which also means that the entire system has experienced the monophone training, 3 triphone training, lda- mllt training and final SAT training. The best results obtained in this process will also be used in the DNN training process. In theory, after step-by-step optimization, tri5 and tri5_ali are the models with the best recognition performance for GMM section, so the two results of tri5 and tri5_ali will be used in DNN training.

Training Models	Train Dataset	NumLeaves	NumGaussian	Train iteration
Monophone	train_sub1		1000	40
tri1-small data	train_sub2	1000	10000	35
tri2-mid data	train_sub3	1000	20000	35
tri3-full data	full train data	6000	75000	35
tri4-lda-mllt	full train data	6000	75000	35
tri5-SAT	full train data	6000	75000	35

Table 4.3: Training Settings of GMM Models

4.2.3 DNN training section

Monolingual Model
Unlike GMM training, in monolingual, the entire system uses the mfcc feature vector of the original sound model as training data. Pitch is still used to represent the timbre information of the sound. Use the sampling rule of window length=25ms and frame shift=10ms to perform mfcc operation on the data within 25ms. The dimensions are the same as those of plp, both are 16 dimensions. After the feature extraction completed, the tdnn network structure is need to be built. The tdnn network structure is shown in Figure 4.2.

First of all, the input of the entire network is a 43-dimensional mfcc feature vector, and the final output dimension is determined by the alignment information in the GMM step. When training the triphone model, in order to save the calculation, Kaldi uses decision tree algorithm to train GMM model. So the output dimension here is the number of branches and leaves in the last alignment information in the GMM step. In this experiment, it is 2136, with a total of 2136 HMM states as the classification class of the entire network. In addition, this tdnn network has a total of 6 tdnn hidden layers, each of which has a dimension of 1024. After these 6 hidden layers, there is a 1024 fully connected layer. These hidden layers and fully connected layers all include relu and renorm layer. Finally, the fully connected layer is connected to the final output layer. For reduce the computation, the experiment also applied the sub-sampled TDNN models. The sub-sampled information and the detail anbout TDNN structure config is shown in Table 4.4. Mention that there are some blank at fully connected layers and output layer. This is because the fully connected layer receive all the inputs from the previous layers' outputs, so they do not need to represent feed-forward method such like the context window [-2,2] or sub-sampled tdnn method.

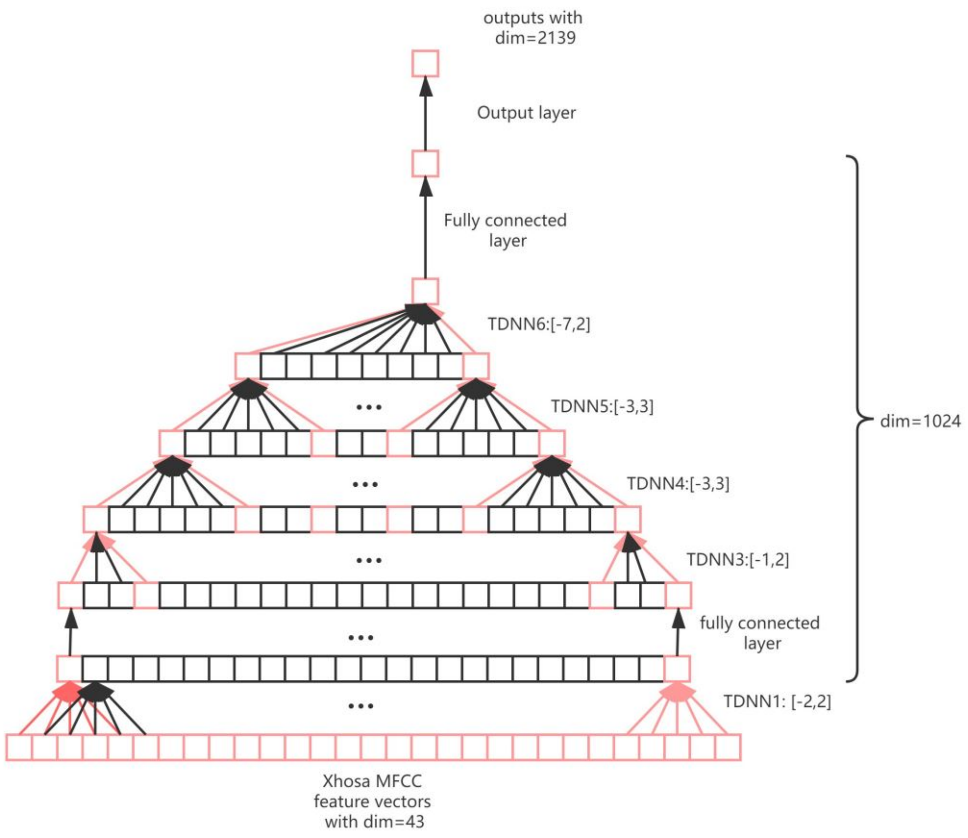


Figure 4.2: The monolingual TDNN structure

layer name	context	Sub-sampled context	input dim	output dim
input layer	[-2,2]	[-2,2]	feat-dim(43)	1024
fully connected layer			1024	1024
tdnn hidden layer1	[-1,2]	{-1,2}	1024	1024
tdnn hidden layer2	[-3,3]	{-3,3}	1024	1024
tdnn hidden layer3	[-3,3]	{-3,3}	1024	1024
tdnn hidden layer4	[-7,2]	{-7.2}	1024	1024
fully connected layer			1024	1024
output layer			1024	num target class(2136)

Table 4.4: TDNN Structure Settings

In the training process, total 2 epochs are trained, each iteration contains 400,000 samples, and a minibatch size of 512 is used to train the tdnn network. Finally, the transation model is applied and adjusted again. So far, the single-language tdnn model is trained.

Multilingual: Xhosa+Zulu

The training steps and settings of Multilingual are almost the same, except that the output layer needs to have two fully connected layers and output layers for the two languages. As shown in the Figure 4.3. And the weight of these two languages trained in the neural network is 1:1.

4.2.4 Decode and Check score

In the decoding process, use the previously prepared test data to test the trained tdnn model, and use the lattice method to score each possible decode result, so as to keep the best ones. At the same time, for the entire system, the WER method is used to evaluate the recognition performance of the entire system on Xhosa language.

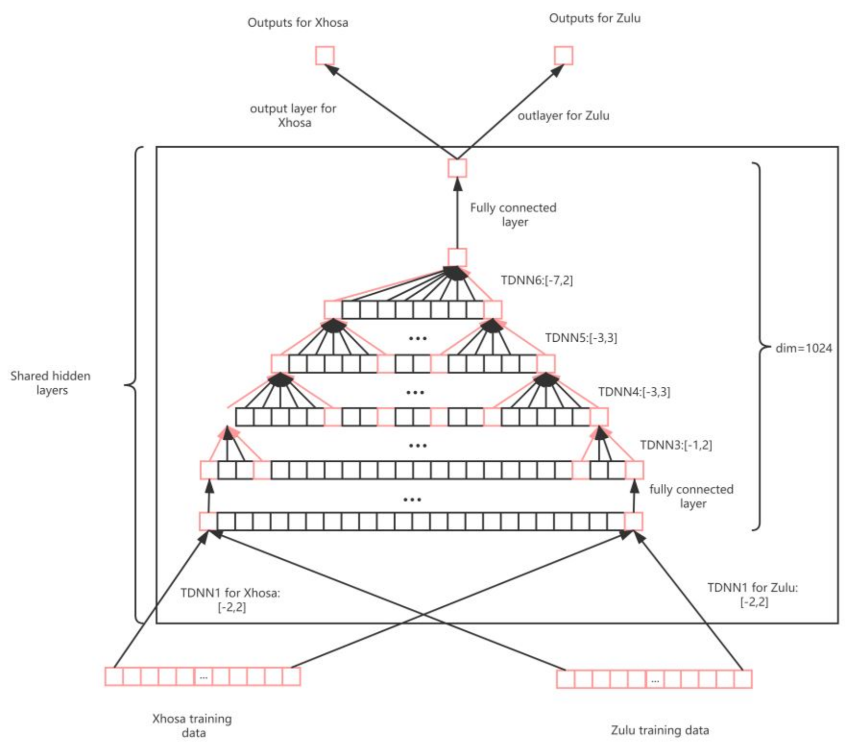


Figure 4.3: The Structure of Multilingual model with the Xhosa and Zulu as input data

Chapter 5

Result, Analysis and Improvement

5.1 GMM training Results and Analysis

All the experimental result is shown in the Table 5.1. For the GMM training part, this project uses the initial language model and the final language model to decode the entire test data set.

GMM Models	tri5_ali for test		lang for test	
	plp feature	mfcc feature	plp feature	mfcc feature
monophone	23.59	21.49	22.97	21.02
tri1-small data	21.02	20.66	20.87	20.41
tri2-mid data	20.71	20.58	20.64	20.34
tri3-full data	20.46	19.96	20.18	19.82
tri4-lda-mllt	21.75	21.53	21.39	21.14
tri5-sat	22.40	22.29	22.03	21.71

Table 5.1: The decode results of different GMM Models

Secondly, plp feature and mfcc feature are applied to train the GMM system respectively.

Among them, the leftmost column represents the various language models from GMM training, from top to bottom is the monophone language model(monophone), the triphone language model with small amount of data(tri1-small data), and the triphone language model with medium amount of data(tri2-mid data). Use all the data to train the triphone language model(tri3-full data), use the lda-mllt algorithm to process the triphone language model (tri4-lda-mllt), and finally the speaker-adaptive SAT tri-

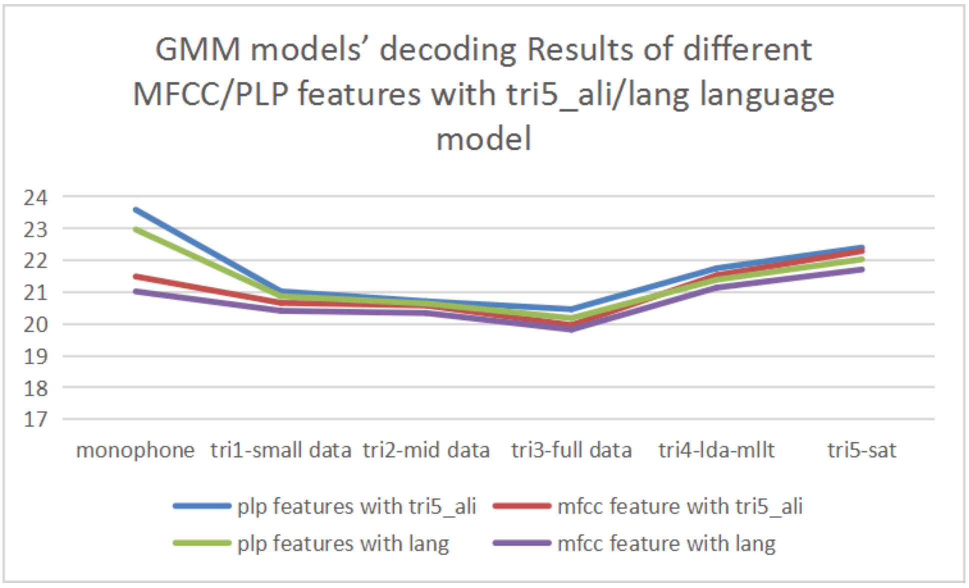


Figure 5.1: The GMM models' decoding results with using PLP/MFCC features and with tri5_ali/lang language model as decoding models

phone language model(tri5-sat). At the top of this tabel, "tri5_ali for test" represents the model will use of the final reestimate language model for decoding, while "lang for test" represents the model will use of the initial language model to evaluate the entire GMM model. To view the trend of the iterations training of models, the results are also shown in the Figure5.1. From the Figure 5.1, before the tri3, the ASR system keep decreasing the WER as expectation especially there is a dramatic drop from monophone system to tri1 system. This large drop also indicate that the triphone system can largely improve the monophone system's performance even though the triphone system only have limited training data(such like tri1). The drop of WER is especially obvious for PLP features trained monophone system. However, at the beginning of tri4 system, the whole ASR models increase the WER rapidly for all the four curves. Besides, at the tri5 system, the WER sill goes up and finally upto approximate 20%. From these findings, it can also be seen obviously from the Table 5.1 and Figure 5.1that there are mainly three points shown in table and figure. (1) First, when using mfcc features to train the GMM model, the performance of the model on Xhosa recognition will be better than the model trained with plp data, although they are all the same model. (2) The second point, in the background knowledge and theory, the algorithm should improve the overall model performance step by step, but the strange thing is that in the actual experiment, only the results before the tri3 step are as expected, the recognition ability of the model is gradually enhanced. But when it comes to tri4 and tri5 models, the recognition performance decrease. (3) The third point is that the system re-estimated the language model during the experiment to make the updated language model more suitable for the current system. Therefore, theoretically, the language model obtained by tri5_ali should be the best. Using tri5_ali to decode should have a better recognition performance. However, the actual result is not as expected. On the contrary, the original language model has better performance. For these findings, there are following analysis:

5.1.1 MFCC features are better than PLP features

For the first point, the model trained with MFCC features is better than the PLP model, this may because that PLP features are much more robustness about And this may be because PLP is more robust to noise. In babel's experiment, babel's data is some phonecall recordings, and the sampling frequency is 8000. Therefore, for babel's training data, the PLP features will be more suitable for these noisy training data, and it is also more suitable for babel's speech recognition model. However, for this experiment, the clean-NCHLT data is used. These data sets are not particularly noisy, and the these high-frequency recordings have 16000 sampling frequency. Therefore, the MFCC feature is more suitable for the data set of this experiment. And that is why the recognition performance of MFCC features' GMM models is better than that of PLP features' GMM models.

5.1.2 Decreasing Performance at tri4 and tri5

Regarding the second point, tri4 and tir5 showed a decline in recognition performance, there are several possible reasons.

* First, it may be that the tri4 and tri5 models overfit or underfit the training data. In order to verify this, I re-trained the tri4 with mfcc features with different epoch numbers of 30, 40, 55. The results are shown in the Table 5.2.

epoch30	epoch40	epoch55
23.25	21.09	21.1

Table 5.2: The tri4 model's decoding results of 30,40,55 training iterations

From the Table 5.2, it is found that the data from training to 30 rounds is obviously worse than the initial result, and as the number of training increases, the training performance of epoch40,50 is obviously better than the performance of epoch30, but it seems that it reached a saturation. There is no way to continue to improve. Modified the number of alignments again, from the default alignment of 10, 20, 30 to 10, 20, 30, 40 alignment. And tested the results of 45 rounds of training. However, the result is still 21.1. It seems that the training has reached a saturation state, but the recognition performance of this saturation state is worse than the previous model. And this may be because I are still not familiar with the training script, and some of the settings may still affect the training performance.

* Another possibility is that the number of Gaussians and leaves used in training are too small, which limits the learning ability of the model. Both lda-mllt and sat require the model to learn more features, so the structure of the model limits the learning ability of the model.

* At the same time, for the tri4 model, this model uses the lda-mllt algorithm. This algorithm reduces the dimensionality of the data and project the original data to a low-dimensional space. Therefore, during the mapping process, part of the information will inevitably miss which may also lead to the result of decreasing recognition performance.

5.1.3 Bad Decoding results when using tri5_ali

For the third point, the decode result obtained by using the final language model tri5_ali is worse than the original language model lang. This may be because tri5_ali is re-evaluated based on the tri5 model. However, in this experiment, the tri5 experimental result is not very good, the decode result is even close to that of monophone model. This bad trained model will also have a negative impact on the re-evaluated language model tri5_ali. Therefore, using the alignment information and language model generated by the model with poor recognition performance will have a negative impact on the entire decoding process.

Similar results are also reflected in the process of training English shown in Table 5.3. In training the GMM system to recognize English, this project uses the mfcc method to extract features, and uses the original language model “lang” for evaluation. The results obtained are as follows.

This result shows that the settings of these models are not suitable for English. Because these settings are based on the default settings of the babel language recogni-

GMM models for English	WER
monophone	86.35
tri1	84.78
tri2	86.46
tri3	79.80
tri4	79.60
tri5	82.35

Table 5.3: English decoding result of different GMM models

tion project in Kaldi. Therefore, the model training settings of the babel language are largely different from babel and also different from Xhosa. This is due to the differences and complexity of the language. The settings suitable for each language require constant attempts of the settings, train, and test the final results so that we can find the suitable settings that are close to the optimal settings.

Because the GMM training part is too bad for English recognition, in this project, English is no longer used as the partner language of Xhosa in multilingual tdnn system.The model's recognition result of Zulu is very similar to that of Xhosa, which is 23.39. It also shows that Zulu and Xhosa are very similar.

5.2 DNN result and Analysis

All dnn results are displayed in the following Table 5.4. The overall decoding results trend are also shown in the Figure5.2.

TDNN models	WER
monolingual tdnn with tri5_ali	20.83
multilingual tdnn with tri5_ali	19.61
multilingual tdnn with tri3_ali	18.48

Table 5.4: TDNN decoding results

For the monolingual model, compared to tri5, the error rate is very low, from 22.29 to 20.83, which is a 6% improvement. This also proves that tdnn has indeed improved model performance based on the GMM part. In the multilingual row, the default combination of Xhosa and Zulu is used, the left side uses the default tri5_ali and tri5 as the

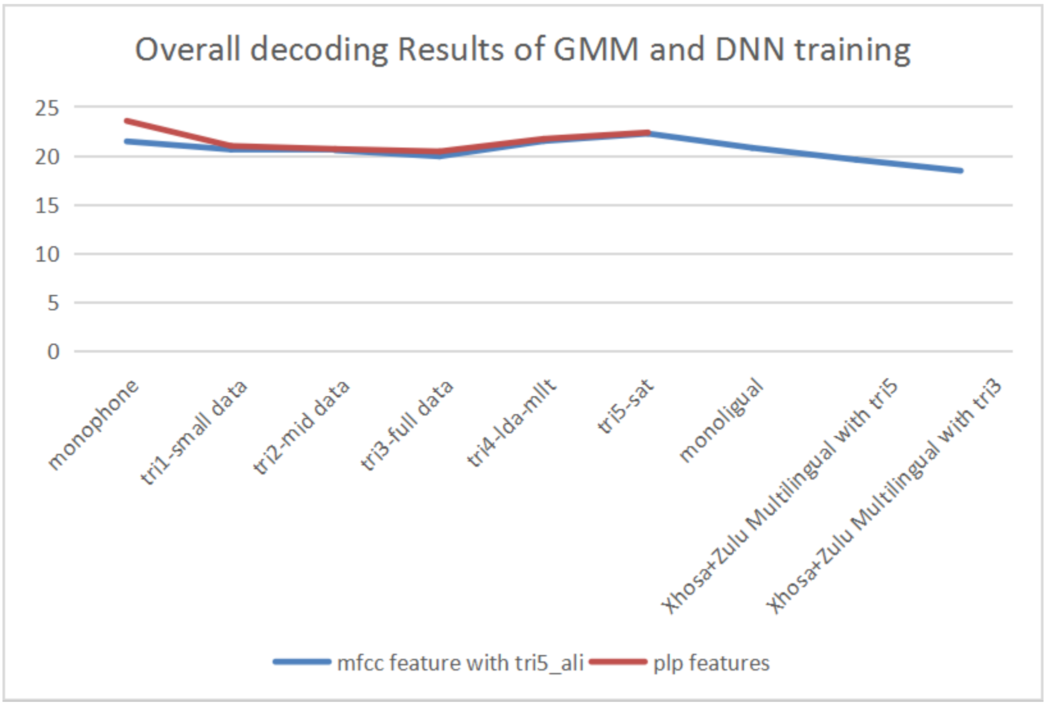


Figure 5.2: The overall decoding results of GMM section training and DNN section training

alignment information of tdnn, and the right side uses the alignment information of the tri3 model that has the best recognition performance in the GMM stage as the alignment information and labels of tdnn. From the results, it is obvious that whether using tri5 or tri3, the multilingual model greatly improves the recognition performance. For tri5, wer dropped from 22.29 in GMM to 19.61 in multilingual, which increased the performance by 12%. In addition, compared to the monolingual tdnn model, it still has a 5.8% improvement. And this also proves that, compared to the single language model, the multilingual model can make up for the lack of training data in low-resource languages to a certain extent, thereby improving the recognition performance of the entire system. As for the tri3 model, wer dropped from 19.96 to 18.48, which also has a 7.4% improvement. All these results prove that the multilingual model has a good performance in language recognition of low-resource languages.

Chapter 6

Conclusion

6.1 Conclusion

Through this project, it can be found that due to the different complexity and composition structure of different languages, it has a deep impact on the training of the model. It usually takes a lot of time to adjust some training parameters to find the model settings suitable for this language. In addition, although there are a lot of GMM models and optimization methods, the GMM model's ability to recognize language is still limited. When the GMM model can't be improved again, dnn needs to be added. The dnn model can learn more language features, thereby improving the overall recognition performance. For small languages, the lack of training data generally limits the dnn model's learning of the language.

Therefore, in order to solve the problem, try several languages with a higher degree of similarity to jointly train the dnn model will be a good idea, which is the multilingual model used in this project. Because if the two languages are very similar, compared to the features that the neural network needs to learn, double the training data can allow the neural network to learn these language features well. For example, the number of features of the two languages is 10, and the training data is also 10. In the monolingual model, 10 data are used for training, and the model needs to learn 10 features. In the multilingual model, if the two languages are very similar, for example, the similarity can reach 60%, then the entire language model uses 20 training data to learn 14 features (6 consistent features + 4 unique features of language 1 + 4 unique features of language 2). Therefore, these language features can be better learned. But if the two languages are completely dissimilar, then 20 training data is still used to learn 20 features, which is the same as the general tdnn. However, this is just an assumption. There are many situations to consider in the analysis of language similarity, such as the repetition of words, the structure of the grammar, the consistent number of n-gram words, and so on. But in general, for two similar low-resource languages, multilingual is a new way to further improve the recognition efficiency of the model, and at the same time solve the problem of insufficient training data for low-resource languages to a certain extent.

6.2 Reminder

The assumption mentioned above language similarity can affect the performance of the multilingual model still needs further experimental verification. Also the tri4 and tir5 models still need to get better results as expected. In the future, the number of different numGaussians and the number of leaves need to be adjusted for different languages to make these model settings more suitable with Xhosa, Zulu and English. For Xhosa and Zulu, look for model settings until tri5 becomes the best performing model, and then use tri5 to train the multilingual model again. In order to verify the influence of language similarity on the multilingual model, it is necessary to adjust the settings of the GMM model to improve the recognition performance of GMM on English to a certain extent(about decrease the WER to around 20 for GMM section). Then use the trained English GMM tri5 as the multilingual alignment information, which can be used as Xhosa's partner as the two training languages in the multilingual network. And the results obtained from Xhosa-English multilingual network are compared with the results of the Xhosa+Zulu collocation to verify the language similarity's Influence. Besides, to investigate more function of the multilingual system, in the future, more languages can be applied into the multilingual system so that to find how will the increasing number of languages influence the multilingual system's performance.

Bibliography

[1] Tasos Anastasakos, John McDonough, Richard Schwartz, and John Makhoul. A compact model for speaker-adaptive training. In *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96*, volume 2, pages 1137–1140. IEEE, 1996.

[2] Etienne Barnard, Marelie H Davel, Charl van Heerden, Febe De Wet, and Jaco Badenhorst. The nchlt speech corpus of the south african languages. Workshop Spoken Language Technologies for Under-resourced Languages (SLTU), 2014.

[3] Peter Bell, Joris Driesen, and Steve Renals. Cross-lingual adaptation with multi-task adaptive networks. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[4] Astik Biswas, Febe de Wet, Ewald van der Westhuizen, Emre Yilmaz, and Thomas Niesler. Multilingual neural network acoustic modelling for asr of under-resourced english-isizulu code-switched speech. In *INTERSPEECH*, pages 2603–2607, 2018.

[5] F de Wet and E. C Botha. Towards speech technology for south african languages: automatic speech recognition in xhosa. *South African journal of African languages*, 19(4):216–226, 1999.

[6] Mark Gales and Steve Young. The application of hidden markov models in speech recognition. 2008.

[7] Mark JF Gales. Semi-tied covariance matrices for hidden markov models. *IEEE transactions on speech and audio processing*, 7(3):272–281, 1999.

[8] M.J.F Gales, K.M Knill, A Ragni, and S.P Rath. Speech recognition and keyword spotting for low-resource languages : Babel project research at cued. International Speech Communication Association (ISCA), 2014.

[9] Arnab Ghoshal, Pawel Swietojanski, and Steve Renals. Multilingual training of deep neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 7319–7323. IEEE, 2013.

[10] Jui-Ting Huang, Jinyu Li, Dong Yu, Li Deng, and Yifan Gong. Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7304–7308. IEEE, 2013.

[11] Dan Jurafsky. *Speech and language processing Daniel Jurafsky, James H. Martin.* Always learning. Pearson Education, Harlow, second edition. edition, 2014.

[12] Sebastian Mika, Gunnar Ratsch, Jason Weston, Bernhard Scholkopf, and Klaus-Robert Mullers. Fisher discriminant analysis with kernels. In *Neural networks for signal processing IX: Proceedings of the 1999 IEEE signal processing society workshop (cat. no. 98th8468)*, pages 41–48. Ieee, 1999.

[13] Thomas Niesler, Philippa Louw, and Justus Roux. Phonetic analysis of afrikaans, english, xhosa and zulu using south african speech databases. *Southern African Linguistics and Applied Language Studies*, 23(4):459–474, 2005.

[14] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur. A time delay neural network architecture for efficient modeling of long temporal contexts. In *Sixteenth annual conference of the international speech communication association*, 2015.

[15] Ruhi Sarikaya and John HL Hansen. High resolution speech feature parametrization for monophone-based stressed speech recognition. *IEEE signal processing letters*, 7(7):182–185, 2000.

[16] Zoltán Tüske, Pavel Golik, David Nolden, Ralf Schlüter, and Hermann Ney. Data augmentation, feature combination, and multilingual neural networks to improve asr and kws performance for low-resource languages. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[17] Ulla Uebler, Michael Schüßler, and Heinrich Niemann. Bilingual and dialectal adaptation and retraining. In *Fifth International Conference on Spoken Language Processing*, 1998.

[18] Phil C Woodland. Speaker adaptation for continuous density hmms: A review. In *ISCA Tutorial and Research Workshop (ITRW) on Adaptation Methods for Speech Recognition*, 2001.

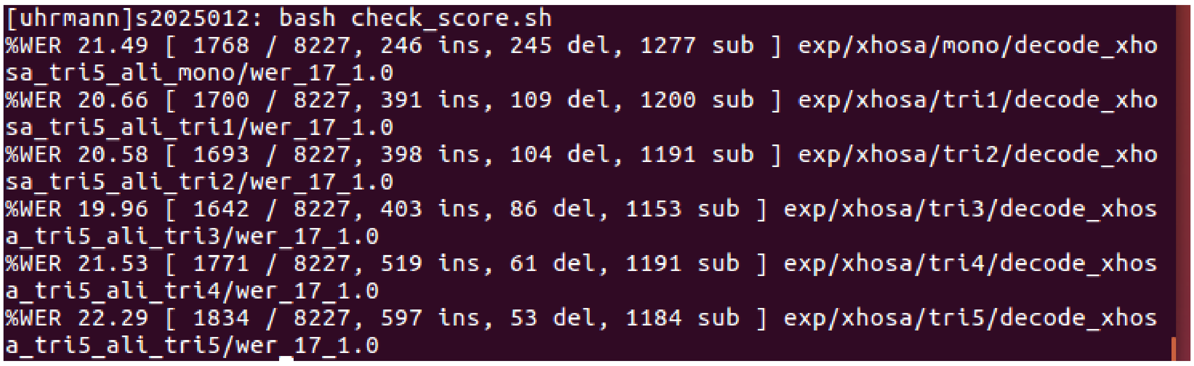
Chapter 7

Declaration

I have read and understood the University's plagiarism guidelines. I declare that this project does not break these rules. I have sourced all the materials that are not from myself. And cite proper reference in this dissertation.

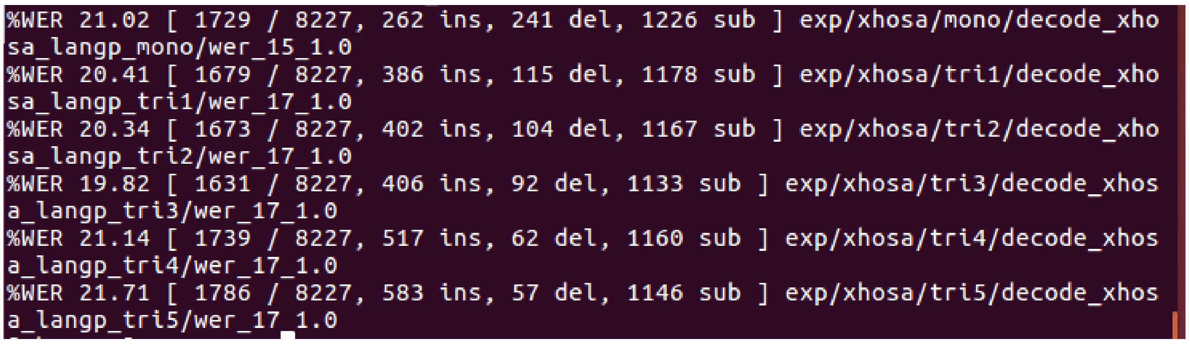
Appendix A

GMM section experiment results



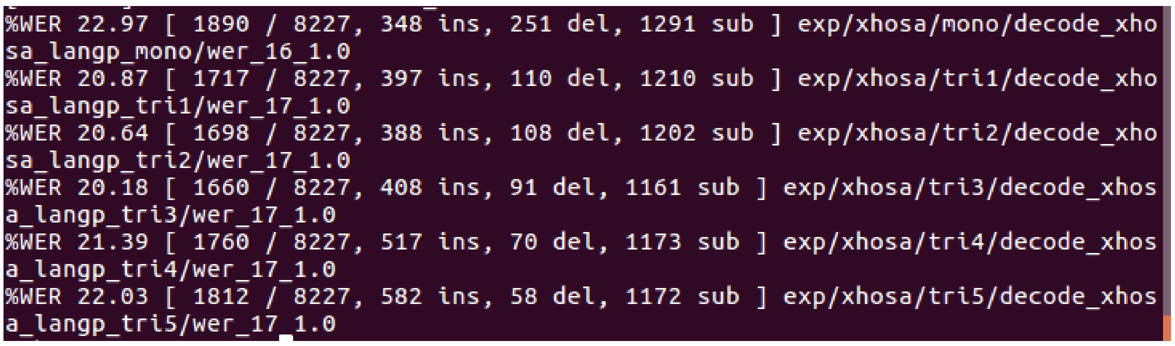
```
[uhrmann]s2025012: bash check_score.sh
%WER 21.49 [ 1768 / 8227, 246 ins, 245 del, 1277 sub ] exp/xhosa/mono/decode_xho
sa_tri5_ali_mono/wer_17_1.0
%WER 20.66 [ 1700 / 8227, 391 ins, 109 del, 1200 sub ] exp/xhosa/tri1/decode_xho
sa_tri5_ali_tri1/wer_17_1.0
%WER 20.58 [ 1693 / 8227, 398 ins, 104 del, 1191 sub ] exp/xhosa/tri2/decode_xho
sa_tri5_ali_tri2/wer_17_1.0
%WER 19.96 [ 1642 / 8227, 403 ins, 86 del, 1153 sub ] exp/xhosa/tri3/decode_xhos
a_tri5_ali_tri3/wer_17_1.0
%WER 21.53 [ 1771 / 8227, 519 ins, 61 del, 1191 sub ] exp/xhosa/tri4/decode_xhos
a_tri5_ali_tri4/wer_17_1.0
%WER 22.29 [ 1834 / 8227, 597 ins, 53 del, 1184 sub ] exp/xhosa/tri5/decode_xhos
a_tri5_ali_tri5/wer_17_1.0
```

Figure A.1: Xhosa decode results with using tri5_ali and mfcc feature



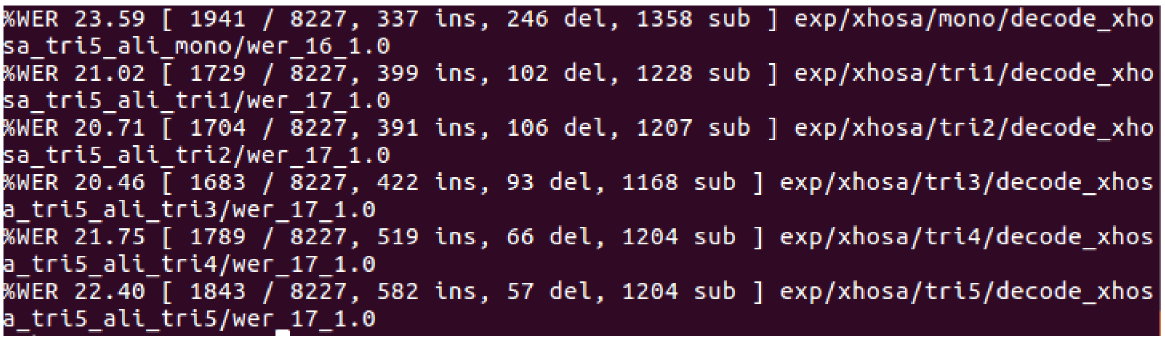
```
%WER 21.02 [ 1729 / 8227, 262 ins, 241 del, 1226 sub ] exp/xhosa/mono/decode_xho
sa_langp_mono/wer_15_1.0
%WER 20.41 [ 1679 / 8227, 386 ins, 115 del, 1178 sub ] exp/xhosa/tri1/decode_xho
sa_langp_tri1/wer_17_1.0
%WER 20.34 [ 1673 / 8227, 402 ins, 104 del, 1167 sub ] exp/xhosa/tri2/decode_xho
sa_langp_tri2/wer_17_1.0
%WER 19.82 [ 1631 / 8227, 406 ins, 92 del, 1133 sub ] exp/xhosa/tri3/decode_xhos
a_langp_tri3/wer_17_1.0
%WER 21.14 [ 1739 / 8227, 517 ins, 62 del, 1160 sub ] exp/xhosa/tri4/decode_xhos
a_langp_tri4/wer_17_1.0
%WER 21.71 [ 1786 / 8227, 583 ins, 57 del, 1146 sub ] exp/xhosa/tri5/decode_xhos
a_langp_tri5/wer_17_1.0
```

Figure A.2: Xhosa decode results with using lang and mfcc feature



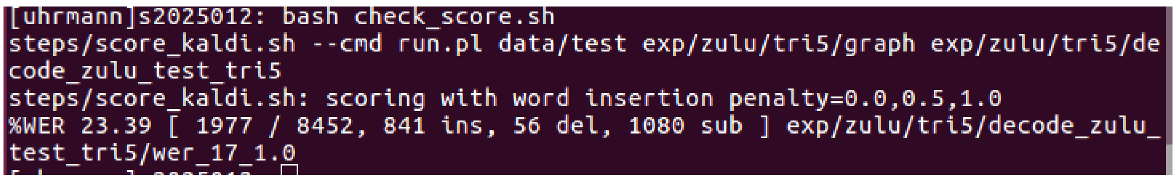
```
%WER 22.97 [ 1890 / 8227, 348 ins, 251 del, 1291 sub ] exp/xhosa/mono/decode_xho
sa_langp_mono/wer_16_1.0
%WER 20.87 [ 1717 / 8227, 397 ins, 110 del, 1210 sub ] exp/xhosa/tri1/decode_xho
sa_langp_tri1/wer_17_1.0
%WER 20.64 [ 1698 / 8227, 388 ins, 108 del, 1202 sub ] exp/xhosa/tri2/decode_xho
sa_langp_tri2/wer_17_1.0
%WER 20.18 [ 1660 / 8227, 408 ins, 91 del, 1161 sub ] exp/xhosa/tri3/decode_xhos
a_langp_tri3/wer_17_1.0
%WER 21.39 [ 1760 / 8227, 517 ins, 70 del, 1173 sub ] exp/xhosa/tri4/decode_xhos
a_langp_tri4/wer_17_1.0
%WER 22.03 [ 1812 / 8227, 582 ins, 58 del, 1172 sub ] exp/xhosa/tri5/decode_xhos
a_langp_tri5/wer_17_1.0
```

Figure A.3: Xhosa decode results with using lang and plp feature



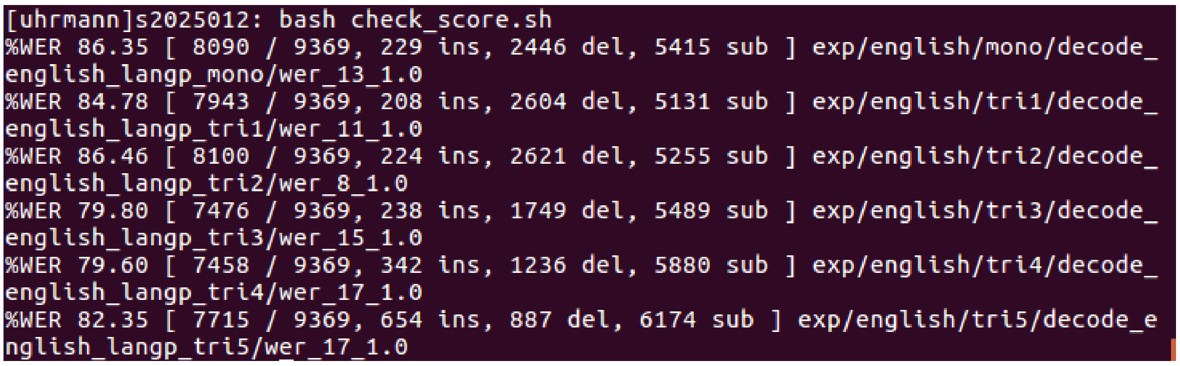
```
%WER 23.59 [ 1941 / 8227, 337 ins, 246 del, 1358 sub ] exp/xhosa/mono/decode_xho
sa_tri5_ali_mono/wer_16_1.0
%WER 21.02 [ 1729 / 8227, 399 ins, 102 del, 1228 sub ] exp/xhosa/tri1/decode_xho
sa_tri5_ali_tri1/wer_17_1.0
%WER 20.71 [ 1704 / 8227, 391 ins, 106 del, 1207 sub ] exp/xhosa/tri2/decode_xho
sa_tri5_ali_tri2/wer_17_1.0
%WER 20.46 [ 1683 / 8227, 422 ins, 93 del, 1168 sub ] exp/xhosa/tri3/decode_xhos
a_tri5_ali_tri3/wer_17_1.0
%WER 21.75 [ 1789 / 8227, 519 ins, 66 del, 1204 sub ] exp/xhosa/tri4/decode_xhos
a_tri5_ali_tri4/wer_17_1.0
%WER 22.40 [ 1843 / 8227, 582 ins, 57 del, 1204 sub ] exp/xhosa/tri5/decode_xhos
a_tri5_ali_tri5/wer_17_1.0
```

Figure A.4: Xhosa decode results with using tri5_ali and plp feature



```
[uhrmann]s2025012: bash check_score.sh
steps/score_kaldi.sh --cmd run.pl data/test exp/zulu/tri5/graph exp/zulu/tri5/de
code_zulu_test_tri5
steps/score_kaldi.sh: scoring with word insertion penalty=0.0,0.5,1.0
%WER 23.39 [ 1977 / 8452, 841 ins, 56 del, 1080 sub ] exp/zulu/tri5/decode_zulu_
test_tri5/wer_17_1.0
```

Figure A.5: Zulu decode results with using tri5_ali and plp feature

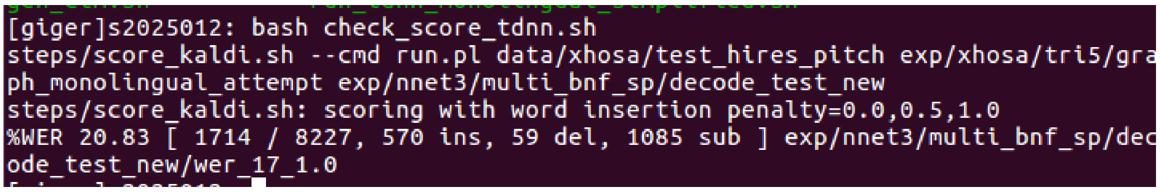


```
[uhrmann]s2025012: bash check_score.sh
%WER 86.35 [ 8090 / 9369, 229 ins, 2446 del, 5415 sub ] exp/english/mono/decode_
english_langp_mono/wer_13_1.0
%WER 84.78 [ 7943 / 9369, 208 ins, 2604 del, 5131 sub ] exp/english/tri1/decode_
english_langp_tri1/wer_11_1.0
%WER 86.46 [ 8100 / 9369, 224 ins, 2621 del, 5255 sub ] exp/english/tri2/decode_
english_langp_tri2/wer_8_1.0
%WER 79.80 [ 7476 / 9369, 238 ins, 1749 del, 5489 sub ] exp/english/tri3/decode_
english_langp_tri3/wer_15_1.0
%WER 79.60 [ 7458 / 9369, 342 ins, 1236 del, 5880 sub ] exp/english/tri4/decode_
english_langp_tri4/wer_17_1.0
%WER 82.35 [ 7715 / 9369, 654 ins, 887 del, 6174 sub ] exp/english/tri5/decode_e
nglish_langp_tri5/wer_17_1.0
```

Figure A.6: English decode results with using tri5_ali and plp feature

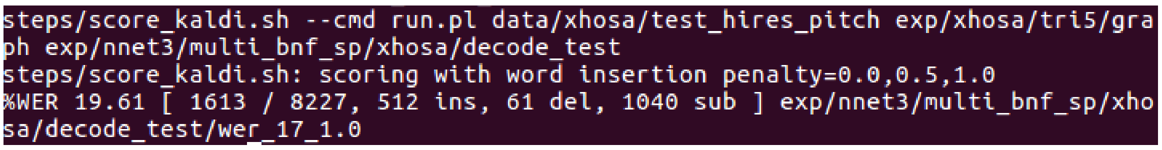
Appendix B

DNN section experiment results



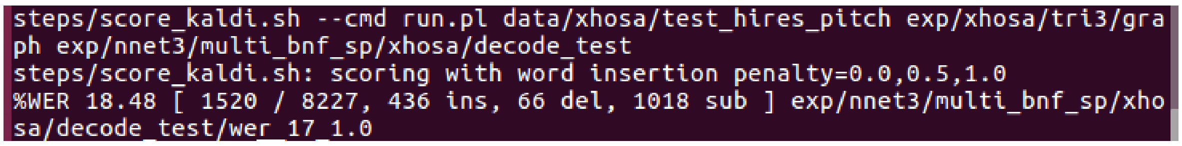
```
[giger]s2025012: bash check_score_tdnn.sh
steps/score_kaldi.sh --cmd run.pl data/xhosa/test_hires_pitch exp/xhosa/tri5/gra
ph_monolingual_attempt exp/nnet3/multi_bnf_sp/decode_test_new
steps/score_kaldi.sh: scoring with word insertion penalty=0.0,0.5,1.0
%WER 20.83 [ 1714 / 8227, 570 ins, 59 del, 1085 sub ] exp/nnet3/multi_bnf_sp/dec
ode_test_new/wer_17_1.0
```

Figure B.1: Monolingual result of Xhosa based on tri5



```
steps/score_kaldi.sh --cmd run.pl data/xhosa/test_hires_pitch exp/xhosa/tri5/gra
ph exp/nnet3/multi_bnf_sp/xhosa/decode_test
steps/score_kaldi.sh: scoring with word insertion penalty=0.0,0.5,1.0
%WER 19.61 [ 1613 / 8227, 512 ins, 61 del, 1040 sub ] exp/nnet3/multi_bnf_sp/xho
sa/decode_test/wer_17_1.0
```

Figure B.2: Xhosa-Zulu multilingual result of Xhosa based on tri5



```
steps/score_kaldi.sh --cmd run.pl data/xhosa/test_hires_pitch exp/xhosa/tri3/gra
ph exp/nnet3/multi_bnf_sp/xhosa/decode_test
steps/score_kaldi.sh: scoring with word insertion penalty=0.0,0.5,1.0
%WER 18.48 [ 1520 / 8227, 436 ins, 66 del, 1018 sub ] exp/nnet3/multi_bnf_sp/xho
sa/decode_test/wer_17_1.0
```

Figure B.3: Xhosa-Zulu multilingual result of Xhosa based on tri3