Warsaw University of Technology

FACULTY OF
ELECTRONICS AND INFORMATION TECHNOLOGY

Doctoral dissertation

in the field of study Computer Science

and specialisation Speech Recognition

A study on speech recognition and correction for non-native English speakers

Kacper Radzikowski

student record book number 6459

thesis supervisor

Professor Robert Nowak, PhD, Eng., Professor Osamu Yoshie, PhD, Eng.

2021

Contents

Chapter 1 Introduction

Nowadays the educational services industry is subjected to dynamic and multiple changes mainly related to the development of the Internet and the progress of various convenient communication tools.

E-learning is a new way of learning using electronic media, most often the Internet. In the last few years e-learning emerged as a new way to deliver online education. To meet the growing need for online education many companies provide platforms for online training and learning. Advantages and popularity of e-learning require the development of new tools supporting this process.

Compared to traditional classroom learning, e-learning has so many advantages from both teacher/trainer/tutor and learner's points of view.

E-learning simplifies learning from people of different cultures, countries and geographical regions. Such situations naturally imply a necessity to use a common language, which is not a native tongue for at least one conversation party. Communication using a common language requires listening, speaking, reading and writing skills for that particular language. Such a necessity implies that it might be beneficial for all the parties, to use numerous available methods for supporting the communication and mutual understanding. For example automatic speech recognition (ASR) tools can be a tool for creating automatic closed captioning for the e-learning lesson, in real time. The main purpose of the ASR process is to provide the accurate conversion mechanism between speech and text domains. It is fairly easy to implement a software or a module of e-learning platform, that could employ any ready-made cloud-based ASR system, which could facilitate in such a scenario.

However, there is a considerable difference in performance of cloud ASR systems, depending on whether the sound samples which are used for recognizing, represents the speech of a native or non-native speaker.

1.1 Motivation

Speech recognition

Automatic speech recognition (ASR) has been the subject of extensive research since 1950s. Enabling the communication between a human and a machine has been one of difficult problems to tackle and one of the most intensively studied topic. The ASR techniques operate in between the domains of speech, audio signal and text. In essence, this methodology provides a capability to recognize the speech signal and what is being uttered and pronounced by a speaker. The speech data is pronounced by a person and becomes an analog signal of the air pressure traversing the space and changing over time as the person continues to speak. The normal frequency of the speech is usually between 300 and 4000 Hz. The microphones convert such a fluctuation into an electric signals, voltages or currents, in which form we usually deal with speech signals in speech processing. Then a digital-to-analog converter changes the analog signal into digital one. The audio signal data is then converted to processed form which depends on the chosen ASR methodology. For example, Mel Frequency Cepstral Coefficients (MFCC) are a very common choice for initial processing of the sound signal. The processed data is used as the direct input to ASR algorithm, e.g. deep neural networks. Finally the algorithm yields result containing the text representation of the sample that was used as the input data. This textual form on the result is defined as a finite alphabet of letters depending on the particular language model utilized.

Traditional approaches for training speech recognition classifiers usually tend to employ supervised learning techniques ([1, 2, 3, 4, 5, 6]). While perfectly fitting for recognizing speech of most popular languages worldwide, supervised learning methodologies will not produce classifiers of decent quality for non-native speakers. The main reason is the lack of labelled datasets of non-native speech, which would be large enough to be used as a training set in a supervised learning algorithm.

Nowadays, automatic speech recognition (ASR) systems achieve higher and higher accuracy rates. The speech recognition techniques and methodologies that have been developed recently can work with up to 90-95% accuracy, depending on the dataset and benchmark test used ([7]).

However, such accuracy levels can be reached only when the system is used for recognizing the speech of native speakers (e.g. English language for North American people). Such performance can be reached only when the system is used for recognizing the speech of native speakers (i.e. the native speakers of the language represented by the dataset used to train such an ASR). In the case of non-native speakers of the language of the ASR system, even the most advanced speech recognition systems can achieve accuracy not higher than 50-60%. The score decreases significantly when the same ASR system is being used with a non-native speaker of the language to be recognized. In the case of non-native speakers, even the most advanced speech recognition systems can only achieve an accuracy of up to 50-60%. The level of accuracy decreases significantly when the same ASR system is used by a non-native speaker of the language which was used to train the ASR model.

The reason for the decreased accuracy of the ASR in cases when the target user is a non-native speaker of the language is that the non-native speech differs very significantly from the native one. Specific pronunciation and accent features related to the

mother tongue of such speakers influence the pronunciation. Non-native speakers have a different mother tongue than the one that is being recognized. Usually, the language used most often by a person is his or her mother tongue. The pronunciation of this language, with its patterns and characteristics, affects the pronunciation of a foreign language, causing the failure of speech recognition systems. This makes their pronunciation of a foreign language biased to some extent which causes the speech recognition systems accuracy to decrease in such cases ([8, 9, 10, 11]). The presence of patterns related to the speaker's mother tongue, which can influence the pronunciation of the second language, represents the general reason behind the drop of accuracy.

The current development pace of the global economy, education and easy flow of the workforce create the need to address the issues related to recognizing the speech of non-native speakers. Global integration creates the need to properly recognize non-native speakers, who represent the vast majority of users nowadays.

The easiest way for speech recognition systems to achieve higher accuracy with non-native speakers would be to train a classifier for speech recognition for a specific language and nationality/ethnic group of non-native speakers of that language ([2], [3]).

However, this idea is not feasible in most real world cases. The reason for this is the size of available speech datasets. In traditional methods of training speech recognition classifiers, supervised learning techniques are usually applied. Those require labelled datasets of a large size. While perfectly fitted for recognizing the speech of tens of the most popular languages worldwide, supervised learning techniques do not provide classifiers of decent quality for non-native speech. The main reason for this problem concerns the size of speech datasets for a particular language. Even if they exist, the number of samples is usually not large enough to build an acoustic model which

could reflect the real-world distribution of speech signal characteristics in one particular language. Additionally, the vocabulary in such databases usually comprises not more than a few thousand words, while a typical dictionary contains tens of thousands of words. Moreover, training a speech recognition classifier for one language and one nationality/ethnic group of non-native speakers would require a new database, which would involve a large workforce and budget. For these reasons, traditional methods of training classifiers for speech recognition are usually not applicable for non-native speech. At the same time, the volume of labelled datasets of non-native speech samples is extremely limited both in size and in the number of existing languages. This problem makes it difficult to train or build sufficiently accurate ASR systems targeted at non-native speakers, which, consequently, calls for a different approach that would make use of vast amounts of large unlabelled datasets.

In comparison to labelled datasets, unlabelled datasets are easily available and larger for many ethnic groups speaking a second language. This vast amount of unlabelled data could theoretically be used to develop a method for training classifiers in recognition of non-native speech.

Non-native speech differs from a native speech in many ways. The phrase "non-native speech" utilized in the context of the ASR, encompasses a wide range of speech forms and proficiencies. Even when regional diversity is considered, this range for a language like English is substantially wider than the range of native speech. However, there are a few characteristics that appear to be very beneficial when describing non-native speech. Style, accent, choice of words, syntax quality are all speech characteristics that could define and distinguish native speech from non-native speech.

Accent

The term "accent" has been extensively researched for decades. Many people have abandoned both words in favour of the more neutral and ambiguous variant due to misunderstanding (and a failure to resolve an absolute distinction) between dialect or accent, as well as rising awareness of negative connections with marked accents and dialects. Firstly it is difficult to develop a scholastic definition for accent because it's not absolute in the common sense; a listener detects an accent if the way the speech sounds is different from his own. Scholarly articles repeatedly state that "unaccented" English does not exist, native speakers' definition of the term accent will always be relative to their speech, and the perception providing the recovery efforts. There isn't a precise set of characteristics defining an accent. It could be understood as "unspecified group of prosodic and segmental elements spread throughout geographic or national space". A listener can usually tell if an accent is present or not. Although accent borders vary for each speaker, I believe some speakers whoose speech any native speaker would recognize as having an accent. Suppose the meaning of accent is "the way people pronounce what they say" and accents have clear regional associations. In that case we could define a foreign accent as "pronounciation associated with a geographic region having a language other than English as the primary language spoken".

Style

For native and non-native speakers, the amount of planning and attention required to generate an utterance may differ significantly. The attention used for utterance generation may also impact actual production, reducing the number of cognitive cycles necessary for producing a sentence and articulating it properly (Pawley and Syder,

1983). Variables that characterize the speech task, formality level, and performance were used to describe the level of consciousness in the past. Speech recognition literature frequently uses careful speech and informal speech to represent speech styles (Eskenazi, 1997). While Labov (1972) advocated for "a one dimension of style, dictated by the level of consciousness devoted to speech," other contemporary characterizations include convention level and connection between speaker and listener. In general, the term style refers to systematic linguistic choices made in response to a group of circumstances. I will use the varying mode to convey the level of preparation required to produce an utterance, and I'll limit the definition of style to formality and difficulty level (audience-directed lexical and structural choices). The variable register will be used to express lexical and structural decisions that are task and situation specific. As I have characterized it, mode varies over a continuum and is intimately linked to non-native proficiency. It also has a direct impact on performance. A native speaker and a non-native speaker with limited skill may talk in the same manner and register (asking a stranger for directions, for example). Still their modes indicate very different degrees of attentiveness. The non-native speaker's ability to enunciate complex phone sequences may be harmed by the higher cognitive load absorbed by attention, resulting in a harsher accent than he would usually display for single words. I believe that the mode differs from the other factors described above. It is not directly visible in the speech produced; instead, it has an impact on how the speech is generated that differs between native and non-native speakers.

Syntax quality

Even though the second language learners are often exposed to the grammar from the beginning of the study process, the inadequate syntax is one of the characteristics of

even highly skilled speech. One theory says that the L1 grammar learned as children instantiate the biologically given Universal Grammar; nevertheless, there is no clear proof on whether L2 learners have access to this resource. Adult learners struggle with concepts related to a different grammar in L1 or L2, such as co-referene through a reflexive. Attention and learning stage have also been found to interfere with the generation of even syntactic notions shared by L1 and L2, like e.g. the acquisition of definiteness for Polish English learners.

Native speakers sometimes do not use prescriptively accurate syntax. Native speakers, on the other hand, are well-versed in the application of fundamental concepts such as definiteness marking. A syntactically wrong phrase may not always come from the faulty application of syntactic rules. In English, native German speakers commonly mix between past and past perfect. Consider throwing a party on a Saturday night. On a Monday morning, being asked "did you go to the party?" is not incorrect; nevertheless, the correct grammar would suggest "have you been to the party?" perplexes, leading one to question if the party was still going on. Non-nativeness is indicated by this form of syntactic misinstantiation, which is subtle but often startling.

Choice of words

The words a speaker uses to communicate a notion can tell whether or not he is a native speaker. Even if a sentence is semantically and syntactically correct, it is non-native. Consider the following pair of sentences.

1. I'm going to have a jelly and peanut butter sandwich.

2. I'm going to have a peanut butter and jelly sandwich.

and

1. Let's disassemble the puzzle.

2. Let's take apart the puzzle.

Every instance is theoretically accurate, although the first is less likely than the second to be pronounced by a native speaker. There are several geographical variations in the way native speakers pick terms (for example, British "lift" vs. General American (GA) "elevator"). In contrast to regionalisms, a lack of knowledge of common lexical patterns indicates peculiarities in non-native speech. Because the language model stores the distribution of words in native speech, this variable might challenge speech recognition.

Pace and smoothness

In conversational speaking, native speakers frequently backtrack, stammer, halt in the middle of a phrase, and talk in fragments. Even across languages, these impacts are comparable (Eklund and Shriberg, 1998). Disfluencies in speech, are not just seen in conversational "modes"; they may also be observed in reading speech when readers fumble over the text.

Native speech has a lot of variation in terms of pace. Some indigenous people speak rapidly, while others talk slowly. Some people talk in spurts, while others speak at a steady pace. However, it appears that fluency measures can be used to differentiate between native and non-native speech. Cuhiarini et al. (2000), for example, indicate that pace has a shaky relationship with perceived proficiency. Some non-native speech reading mistakes are distinct and measurable. While certain disfluencies appear to follow universal patterns, others, such as the native-language interjections in the exchange reproduced above, strongly suggest that the speaker is not a native English

speaker.

It appears that native speakers may identify non-native speakers based on accent, syntax, and fluency. Children may recognize and mimic particular features of speech that distinguish it as belonging to a non-native group. When a listener is first exposed to non-native speech, he may struggle to grasp it at first, but he can usually adjust fast if he is a cooperative listener. Humans are very well-equipped to interpret speech and tolerate minor variations. Unfortunately, the machine does not possess any of these abilities. The statistical models of patterns observed in training corpora are used to help computers interpret speech. When the speaker's accent, syntax, and lexical choice are not well-represented in a training corpus, the models must be altered in some way if good recognition is to be accomplished. We can consider several approaches to combating such adaptation. The acoustic model specifies the predicted mapping of auditory events to phonetic units. This is an incredibly fine-grained representation in a completely continuous context-dependent system like the one described in later chapters. Acoustic events are represented on a sub-phonetic level, allowing for much more variants to be identified than in a standard phonetic analysis; the recognizer employed in this dissertation models 118 different realizations of /t/ in GA. The appropriate location to describe phonetic variations in realization for a particular speaker's accent would be in the acoustic model.

Modeling phonemic disparities and phonological adaptation in production would be possible using the lexicon, which defines the phonemic makeup of words. Phonemic substitutions, epenthesis, elision, and phonetic realizational variations may all be simply expressed by changing the lexeme specifications. The issue is that the changed lexion may not interact with the acoustic model in the way it should. Lexial modeling, on the other hand, is a simple method that has been employed with little effectiveness

for native and non-native speech for non-LVCSR tasks (Humphries and Woodland, 1997; Huang et al., 2000). (Fung and Liu, 1999).

The language model encodes the recognizer's understanding of how words are sequenced. The recognizer has no grasp of the meaning of a postulated speech without a natural language comprehension component. It must rely on a statistical model to evaluate the likelihood of a sequence of words being pronounced. The limits on likely word sequences could be loosened by changing the language model, allowing for more tolerance of divergence from natural speech patterns. Alternatively, a statistical model of non-native speech could be trained, specifically describing patterns often occurring in non-natives' speech.

At the end, the system might be tweaked to provide more flexibility in non-native speech processing. Like listeners, who can inquire the person speaking to repeat the utterance, defer processing for the sake of context creation, and silently prompt syntactic, lexical and phonetic mappings for negative and positive examples, so could a system that tries to parse the speech of a non-native speaker using the help of natural language understanding and dialogue components.

1.2 Purpose of the dissertation

This dissertation represents the research on automatic speech recognition adapted for non-native speech. The research done within the scope of this thesis is meant to provide a solution to the problem of ASR for non-native speakers, explained in details in Section 1.1.

In the conducted research, I was looking for a way to increase the accuracy of ASR process in cases where the target user does not represent the native speaker of the

language for which the specific ASR system had been created. At the same time, I was looking to make sure that the appropriate solution remains scalable and usable in real-world cases, mainly in the face of numerous combinations of languages and, respectively, their non-native speakers representing different nationalities. Namely, considering the possible nationalities and language backgrounds of non-native speakers of only one language (e.g. English), we can already see the vast number of cases for which it is necessary to develop a methodology of increasing the accuracy of ASR systems.

The scale of the problem is already significant even if we consider only one language and its non-native speakers. Therefore, the focus of my work was to come up with a methodology that could be easily applied not only to one language but potentially to any number of languages and their respective non-native speakers. An ideal methodology would also have to be scalable in a way that it would not require costly manual labor such as labeling the audio samples by hand.

1.3 Research target as the machine learning problem

The problem which I tried to solve within the scope of this research, consists of several levels of a generic ASR pipeline, to which an improvement can be applied. Namely, I tried to provide supplementary algorithms for increasing the final accuracy of the ASR system by:

- developing a scalable methodology of creating an ASR system,
- creating the algorithm for adapting the input speech samples, so that the overall accuracy of the system can be improved,

- creating a supplementary algorithm representing a language model of the non-native speaker's mother tongue.

Each of the aforementioned problems presents a challenge as one of many problems related to the machine learning domain.

Every problem was deeply researched within the scope of this thesis. For each problem, multiple algorithms were designed and evaluated with experiments using real-world data. Each problem and set of examined solutions were published and presented in journal papers.

1.4 Layout of the dissertation

This document is organized into seven chapters. **Chapter 2: Related works** provides a summary of extensive research conducted on the problem of speech recognition for non-native speakers. **Chapter 3: Dual supervised learning** describes research conducted on possible ways of creation and training of acoustic models adapted for any particular group of non-native speakers, without the necessity of obtaining large vocabulary datasets of the non-native speech. It also describes the research conducted on methods for text-level support and correction of results for language models, adapted for non-native speech recognition. It is an auxiliary method, acting as an extension of method described. **Chapter 4: Audio style transfer for on-the-fly speech correction** represents research conducted on methodologies for real-time modification of accent for non-native speech samples. The idea presented in the chapter provides the possibility of using already existing ASR systems for recognition of non-native speech not constrained by nationality and mother tongue. **Chapter 5: Conclusion** provides a brief description of all the research done within the range of this dissertation as well

as the main contributions. It also discusses directions for future research on the topic. **Appendix A** describes multiple datasets used in experiments carried out within the range of this research.

I have published presented research in scientific journals and proceedings of scientific conferences. **Dual supervised learning** (Chapter 3) is partially depicted in [12]; **Audio style transfer for on-the-fly speech correction** (Chapter 4) is presented in [13]. Moreover I am a co-author of [14, 15, 16, 17, 18], where some of my models, algorithms and tools are presented.

Chapter 2 Related works

2.1 Automatic speech recognition

Davis, Biddulph, and Balashek developed a digit recognition method in 1952 based on formant frequencies observed during each digit's vowel regions. The setup was designed to accommodate only one speaker.

Olson and Belar attempted to distinguish a single speaker's ten different syllables as embodied in ten monosyllabic words ([19]). RCA Laboratories completed the work in 1956.

Fry and Denes of University College in England, built a phoneme recognizer [19] that could react to nine consonants and four vowels in 1959. The employed techniques included pattern matcher for the recognition and spectrum analyzer. With the help of statistics, they managed to increase the phoneme recognition accuracy for words composed of several phonemes. Experiment was the initial utilization of statistical analysis for the ASR area.

In the 1960s, Martin created a list of elementary time normalization methods [20] for detecting voice beginning , end points. It provided a significant contribution to reducing the variability of the recognition scores. Simultanouelsy, Vintsyuk suggested the utilization of dynamic programming methods, e.g. dynamic time warping, for time alignment of a pair of speech samples including algorithms for joined word recognition.

Hearsay I managed to utilize semantic information for limiting the amount of choices for the recognizer to be analyzed, in 1973. The Harpy system at CMU was able to recognize speech with reasonable accuracy utilizing a vocabulary of 1011 words. DARPA provided funding for these initiatives (Defense Advanced Research Projects

Agency).

The template-based methodology changed to a statistical approaches in 1980. Hidden Markov Model (HMM) was one of the major innovations, however it wasn't widely used until 1980s. Cepstral coefficients were introduced by Furui as spectral features for speech recognition. For large vocabulary speech recognition systems, IBM invented the n-gram models, defining the likelihood for occurrence of sequenced n characters or tokens. The construction of a language model, describing the probability that a sequence of language symbols will emerge in a speech sample, was the main focus.

The DARPA program was resumed throughout the 1990s. The focus was on various speech comprehension application domains, such as broadcast news transcriptions and conversational speech. Multiple applications, including an automatic voice-based document indexing and retrieval systems, built using BN transcription technique combined with information extraction and retrieval technology [21]. To lessen the influence of the background noise, microphones, and single voices, many alternative techniques such as structural maximum a posteriori (SMAP) method, model decomposition, maximum likelihood linear regression (MLLR), parallel model composition (PMC) were created.

Rabiner and Sambur suggested "A Statistical Decision Approach to the Recognition of Connected Digits" [22]. Each three-digit string utterance was first examined to determine start point and end point as well as a voiced and silenced parts of the utterance. Based on the voiced-unvoiced-silence information, the digit string was then divided into individual digits. Linear predictive coefficients are used to examine the voice region in each segmented digit (LPC). To determine average digit length, reflection coefficients and linearly distorted or the LPC coefficients are transformed to parcor. A distance metric based on minimal residual error is used to recognize each digit within the string. The effect of coarticulation and multiple repeats is also taken

into account. This approach can be utilized in instances when the speaker is both independent and dependent. In the speaker-dependent mode, the ASR system was evaluated using six speakers. The level of precision was 99 percent. The system was tested with 10 new speakers in speaker-independent mode, and the claimed accuracy reached 95%.

"Simplified, robust, training technique for speaker trained, isolated word recognition systems" ([23]) method by Rabiner and Wilpon, was proposed to reduce the amount of training that is required in machine learning based approaches. It provides a methodology that combines the benefits of averaging and clustering. Casual training is less reliable and robust than the recommended strategy. For the first time, the user's spoken word is evaluated and saved. When he or she says it a second time, the DTW distance between the second utterance pattern and the previous pattern is calculated. In case the threshold is greater than the distance, the system creates the reference pattern and training related to the word is finished; if not, another pass is performed to create the word reference again. The technique is repeated for all the words or a maximum number of repetitions has been reached. An experiment with nine talkers was conducted to examine the effectiveness of the training technique (five males, four females). For a 39-word vocabulary comprising of alphanumeric, numerals, and three designated words, a Word reference template was constructed. The experimentation revealed that the first four repetitions of a word pronounced by a specific speaker yield a single reference pattern for 95.2 percent of all words.

"Large Vocabulary Speaker Independent Speech Recognition System Using HMM" [24] by Lee described SPHINX, an HMM-based speaker independent large vocabulary recognizer, in their article. Function-word-dependent and context-independent phone models are both used by the system. Each word in SPHINX is represented by a phonetic

network, and the set of grammar-accepted sentences is represented by a word network. Three sets of parameters, like LPC cepstrum factors, differenced LPC cepstrum factors, power and differenced power, are computed in order to add knowledge to HMM. The speech sample frequency is 16 kHz, and there are 12 LPC cepstral factors created, which are later for Mel-scale via a bilinear transformation and the vector converted into three blocks, improving ASR accuracy and reducing noise. The HMM recognizer (SPHINX) that is based on a phone. To simulate phones in 42 selected function words, totally 153 HMM were constructed utilizing a list of 105 HMM. A forwardbackward approach is used to train the 153 HMMs, which runs on a 4160 sentence database. A timesynchronous Viterbi beam search approach is utilized to recognize speech. A threshold is set, and those states that are poorer than optimal state more than the threshold are pruned at a given period. For a word pair language model, no language model and a bigram language model the system can recognize speech. The system was tested for 997 words, with accuracy of 53.4 %, 87.9% and 93 % for no language model, word pair and bigram model, respectively.

Suzuki proposed an ASR system composed of acoustic models by introducing variations in voice features [25]. The method utilizes a tree-based clustering technique to build a voice-characteristic dependent acoustic model. The phonetic context is determined using triphone models and language phonetic knowledge. The voice of each speaker voice is categorized based on results of a test in order to develop voice-characteristic-dependent acoustic models. As triphones for context-dependent setup can be quite big, they are clustered. Thereby, the speaker's vocal characteristics are clustered using a tree-based method. The creation of voice-characteristic-dependent acoustic models is enabled by the simultaneous grouping of voice characteristics and phonetic context. Every node that is a leaf with identical phonetic context except var-

ious voice features is combined as a mixture distribution for speech recognition. The procedure is replicated from top until leaves, selecting the positive or negative node in terms of phonetic characteristics and together positive and negative nodes in terms of speech qualities. Finally, we have a group of leaves that differ solely with their speech qualities. The model was trained with 20000 samples from 130 different speakers and evaluated with 100 samples from 23 different speakers. Samples are downsampled with 16 kHz and also parameterized with 12 Mel-cepstral factors for the evaluation. 43 Japanese phonemes as well as 146 phonological context questions with 20 voice characteristic questions were all modelled using three states HMMs. Before integrating a voice feature dependent model and after it, embedded training was used. In the case of males, the suggested method outperforms the traditional 4 mixture model, while for females, the algorithm outperforms the traditional 8 mixture model.

"Speaker Independent Continuous Speech and Isolated Digit Recognition using VQ and HMM" ([26]) developed by Venkatramani and Revathi utilizes perceptual characteristics of sound signal. For speech recognition, it employs a combination of vector quantization and Hidden Markov Models. Perceptual characteristics are recovered with calculation of the power spectrum of sound signal window first, before grouping to 21 essential bands in bark scale. Loudness equalization and cube root compression are used to imitate the power law of hearing. The LP coefficients are turned into cepstral coefficients after IFFT and LP analysis. Calculating characteristics based on testing and training data as well as creating vector quantization clusters of 0-9 digit and speech samples, compose the steps involved in voice recognition using VQ. The codebooks are created using the K-means clustering technique and training data. To improve the training set result vectors probability, Hidden Markov Models with initial probability distribution, probability of state transition and probability distribution of observation

character, are built. The algorithms were instantiated using 8 states and 256 observational sequences for discrete HMM. The models are trained using the indexes from code books. HMM models are fed sequences of observation from input vectors of features for all testing samples, and values for probability density are calculated. Secondly, the probability scores are being compared, after which the utterance with the highest likelihood is chosen. The system's average accuracy utilizing vector quantization and Hidden Markov Models reaches 93% for isolated digit samples, and 100% for continuous speech.

Punjabi Automatic Speech Recognition System was developed by Dua, using HTK composed of Hidden Markov Model [27]. Its graphical user interface was created in a Linux environment utilizing the JAVA platform. Acoustic analysis, training data composing, creation of the acoustic model and the GUI based decoder are four phases of the system design. The recording and classification of speech signals is the first phase. A unidirectional microphone was used to record 115 different Punjabi words, which were used to train the system. The data is sampled at a frequency of 16 kHz. The data was recorded by 8 speakers, with each speaker repeating each syllable three times. The feature preparation stage, where the original sample is converted to a collection of acoustic vectors, is the second step. The MFCC (Mel frequency cepstral coefficient) approach is used to extract the features. This signal is divided into a sequence of frames, each lasting 20 to 40 milliseconds. A windowing function is applied to each frame, and then a set of acoustical factors is retrieved based on each frame window. Comparisons are done throughout the acoustic model creation phase in order to distinguish unknown utterances. The HMM is first set up by creating a prototype for each word. Some topology is employed to generate the prototype, consisting of four observation functions and two states. The HRest tool is then used to estimate optimal

values for HMM parameters. The test sample is transformed into a list of acoustic feature maps in order to distinguish speech. This information, along with the HMM definition, the dictionary for Punjabi, the network of the task, and the resulting HMM vector, is passed into the HVite (HTK tool), comparing it to the recognizer's Markov models and displays the result text. System's accuracy is evaluated with a variety of settings, with a total of six distinct speakers saying 35-50 phrases each. The average performance falls anywhere between 94 and 96 percent.

A system called Continuous Hindi Speech Recognition was proposed by Kumar, with the usage of Gaussian Mixture HMM [28]. The research compares the system's performance to a variety of Gaussian mixtures. The goal is to determine the best number of Gaussian mixtures for optimum accuracy. The system makes use of a database of 51 words captured at a sample rate of 16 kHz. The MFCC method is used to extract features. In this experiment, 39 MFCC are employed. 5 states from left to right without stops skips are utilized for proof-of-concept algorithm in Hidden Markov Model training of continuous Hindi speech recognition system, and 40 prototype HMM models for all Hindi monophones are generated. To improve recognition accuracy, the monophone model is extended to a triphone model. To evaluate the system's performance, many sorts of experiments were carried out. Experiments with various vocabulary sizes revealed that the system performs better with a short vocabulary. Five different numbers of Gaussian mixtures were used in the experiments. With four mixes GMM, a triphone-based continuous speech recognition system claimed good accuracy. Second experiment revealed a tri-phone-based context-dependent system outperforms a mono-phone-based system that is context independent. With a 51-word vocabulary, the authors were able to attain 97.04 percent accuracy.

The deep neural network based improvement of ASR system, uses exemplar based

approach was developed by Baby [29]. As a preprocessing stage, the system used connected dictionaries. The noised speech is initially represented as a weighted sum of symbols in the dictionary with examplary samples drawn from a given domain. The resultant weights are put in to a connected dictionary with example samples into a short-time Fourier transform in order to provide direct estimations of speech and noise (STFT). Three alternative input spaces comprising of magnitude STFT, Mel and MS, were used to evaluate the system. Mel-Mel and Mel-DFT, DFT-DFT and MS-DFT settings were the three types of settings employed. DFT related space is utilized as the input example for DFT-DFT configuration. Randomly chosen part of acoustic data composed of T frames and its STFT of size F*T (in full resolution magnitude) is used to generate the input dictionary using DFT exemplars. Mel-Mel and Mel-DFT use a Mel dictionary with Mel exemplars to perform NMF-based decomposition. The magnitude STFT of size F*T is pre-multiplied with the STFT-to-Mel matrix to get Mel example. The MS-DFT configuration uses MS exemplars to generate a compositional model using NMF. T frames of acoustic data are used to generate MS-exemplars, which are then filtered using a filter bank of B channels. To represent non-negative nerve firings, the B band-limited signals are half-wave rectified and low-pass filter was employed at a 3 dB cut-off frequency. With both clean and multi-condition training, the system was trained and tested using the AURORA-4 database. The performance of various settings is evaluated and compared using average word mistake rates. With retrained and clean DNN, the system averaged all WERs of 11.9 % and 26.8%, respectively.

The English ASR system was improved by Nguyen ([30]) utilizing hybrid approach of Deep Neural Network and bottleneck features compiled by denoising encoders. For Hybrid HMM/GMM, the Deep Neural Network design comprises of a high amount of densely linked layers, before a classifier at the end. The architecture with extraction

of bottleneck feature resembles that of a hybrid HMM/GMM, except for bottleneck layer of a smaller size. Nguyen proposed utilizing TED talks sampled dataset to train the acoustic model, composed of audio spread across 920 talks and 22 hours. Noises were eliminated utilizing segmentation, leaving about 175 hours of speech for training. For language modeling, 12.5% of the Giga corpora was filtered with Moore-Lewis technique. The neural network predicts context-dependent HMM states during supervised training. The auto-encoders are pre-trained with a learning rate of 0.01 percent using the gradient descent approach. Masking noise has distorted the input vectors. The remaining layers are then given a bottleneck of 39 units. The system was evaluated by the authors using the 2012 development set and the 2013 test set. Baseline system's WER is 30% on dev2012 and 36.1 percent on test2013. With error rates of 18.7% and 22.7 percent, the hybrid DNN/HMM combination beats the baseline configuration.

An ASR system composed of Mel-Frequency Cepstral Coefficients and Dynamic Time Warping was implemented in MATLAB simulator by Mohan [31]. Feature Extraction and Feature Matching are the two steps of the system. Speech samples are transformed from Analog to Digital using Pre-Emphasis and filter before being extracted using MFCC. For preemphasis, a FIR filter is utilized, which increases the magnitude of higher frequencies in comparison to lower frequencies. The voice sample is framed between 20 and 30 milliseconds. A hamming window is then applied to each frame. The signal is then transformed into frequency domain using a Fast Fourier Transform for each frame. The Triangular MEL filter bank is then multiplied on each resultant frame. The MFCC values are the end outcome. Following feature extraction, the DTW algorithm is used to match features by computing the shortest distance between spoken word features and reference templates. The reference pattern having the lowest score gets chosen for the final result from among the computed scores.

Kumar Ravinder has purposed a speech recognition system for isolated word recognizer for Punjabi language [32]. The intended system is a speaker-based system that operates in real time. The Hidden Markov Model (HMM) and Dynamic Time Warp (DTW) methods were used to create a system for a limited vocabulary of isolated spoken words in an Indian regional language (Punjabi). Following the creation of systems, research is expanded to include a comparison of such systems. The goal of this research is to develop a template-based recognizer that uses linear predictive coding (LPC) for feature extraction, as well as dynamic programming computation and vector quantization using Hidden Markov Model-based recognizers. End point identification in purposed systems is accomplished by locating energy in the spoken signal wave. To extract background silence before and after the input voice, end point detection is used. The algorithms are trained five times for Punjabi numerals. The system performs 92.3 percent with DTW and 87.5 percent with HMM for Punjabi language numerals. The results of the systems demonstrate that the DTW method is better for Punjabi numerals and isolated spoken syllables.

The diagram below demonstrates the information related to datasets utilized in the mentioned research papers.

2.2 Automatic speech recognition with adaptation to non-native speech

As real-world systems begin to be implemented, multilinguality in speech recognition systems has gotten a lot of attention. The actual language utilized for development was less essential than the learning and modeling approaches that were being perfected when the research goal was to build a plausible speech model. While there were some

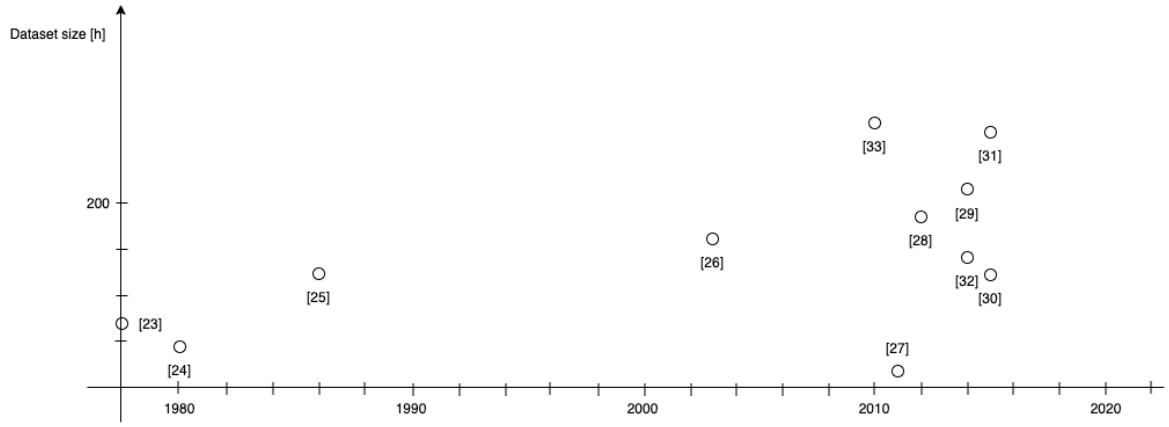

Figure 1: The sizes of datasets utilized across the ASR research over the years.

language-specific challenges to be overcome, such as vocabulary specification in German, French liaison and Chinese tone, most researchers focused on creating models of their languages and occasionally another widely used language e.g. English. However, when individuals began using the systems, significant overhead needed in establishing a recognizer in a new language, as well as the computational requirements required for executing many recognition systems that could recognize a variety of languages became more appealing. A multilingual system often employs a large list of phones encompassing all supported languages, with the training samples shared across languages for the phones that have comparable features and easing the process of a new language addition as the overall system's phoneme and language library develops.

While multilingual systems appear to be quite similar to non-native systems at first glance, there are a few key differences. Users represent both non-native and native speakers of the language to be recognized in a multilingual system. The skilled speakers with advances semantics and syntax, and phonology related to their language, despite the diversity that is always present in native speech. The common knowledge related to modeling of patterns for native speakers applies to all languages in a multilin-

gual system because the languages are spoken by people who are native to the area. L1 interference between speaker groups is not a problem. Shultz proposed an algorithm to introduce phonemic contexts into the allophonic decision tree. As the vectors of phonemes existing for every new language that is new, may be listed based on current linguistic information or the extension of the corpora with its phonemic representation, not existing in languages defined before inside the system, could be defined. By reducing branches impacted with newly introduced polyphones and recreating the sections using the previously unseen sound data and retraining the related distributions, the authors adjust the current decision tree to the new phonemic surroundings. Shultz and Waibel claim that this method reduces WER significantly while using a fraction of the acoustic data required to completely retrain the new polyphones.

Imperl suggested a polyphone clusterization approach that works with multiple languages. To exchange training data, he suggested merging together polyphones that have a triphone distance lower than a certain level, considerably reducing the amount of data represented by his system while suffering only a minor loss in terms of WER. Kohler experimented with comparing three methods of representing a phoneme inventory of both multilingual and independent of context systems, proving the density clustering initiated from the IPA transcription of phones in multiple languages to be superior to clustering based only on IPA symbol or clustering based solely on data. Kohler compares the representational differences between these methods, finding the most superior algorithm works at the sub-phone level, whereas employing IPA standard alone is not sufficient to allow for specialized modeling.

A variety of organizations have collected and acknowledged accented speakers. The Australian National Database of Spoken Language contains both Byrne corpora and datasets of non-native speakers. Amongst these, those born in Australia however claim-

ing their first language to be different than English and people who moved to Australia after becoming adults. The group of non-native speakers comprised mainly of the South Vietnam citizens and Arabic groups from Lebanon, while other migrant groups were represented as well. SRI's technique for collecting speech data of Latin American Spanish learners, is described in depth by Bratt in 1998. The non-native collection was part of a bigger pro jet that captured several different types of Latin American Spanish. Sentences were largely adapted from Spanish newspaper articles, and length and phonetic overage were taken into account. A subset of the 43,460 non-native speech utterances has been phonetically labelled so that articulation errors could be investigated. Transcribers were given the option of using the merged set of English and Spanish phones in their phonetic transcriptions, as well as diacritics in order to highlight how a Spanish phone could sound non-native provided that the errors were more complex in contrast to simply substituting the phoneme in English.

When creating a speech database, one of the most essential things to consider is how effectively disfluencies should be captured. We know that samples of frequent phrases and constructions are required for language model training and must be solicited during data collecting. While interruptions in the speech regularity are a substantial reason behind mistakes for Switchboard, and hesitation tokens could be utilized for more accurate anticipation of other words, improved disfluency modeling did not increase ASR accuracy accuracy considerably, as it has been proved by Stolke in 1996).

Although disfluency patterns seem to have similarities between Swedish and English (Shriberg, 1998), it may not necessarily imply the existence of the analogical relationship for other language pairs, or that non-native speakers are characterized by disfluency patterns of their L1, L2, or a mix of the two. When working with previously unsampled populations, many of the assumptions established when gathering

voice data are questioned. When capturing data from children and speakers of languages with low literacy standards, In 1997, Eskenazi mentioned speaker competency in speaking and reading as being among the characteristics to consider. Similar characteristics must be addressed in case of collecting data of non-native speakers who are lettered in the L1 language, which presents a unique difficulty for data collecting protocol design and implementation. One does not want to irritate the speaker since this will undermine the data's integrity and leave the speaker with unpleasant sentiments.

Ethics have evolved in areas where recording speech in order to analyze speech related patterns has been customary practice, which we should be urged to respect, even when it may appear that sole reason behind the interviews is due to speaker's poor speech quality. In 1984, Labov explained a variety of problems in speech data gathering in the content of description of the field methodology for the research on linguistic variation. He underlined the importance of not making the speaker feel objectified or misunderstood after the data collection experience.

In case that both are arranged situations prepared to obtain natural speech before transcribing and analyzing, the sociolinguistic research activities often rely on the interview as a method that is often used to collect samples. Both have differing notions of "natural speech" and whether it can be obtained in a controlled environment. The main distinction is the volume and diversity in data required; for ASR training, we need several hours of samples from speech coming from a variety of speakers, but much sociolinguistic research focuses on only narrow list of speakers. With the expansion of the dataset collection to include additional speech groups, it would be beneficial to refer to insights and advice sociolinguistics scientists, who frequently target users of dialects of languages like English.

Briggs (1986) makes a number of observations about the interview that appear to

be pertinent to data collecting for the LVCSR. He highlights the significance of the speaker comprehending the purpose of related event, e.g. an interview, usage of a speech based translation application. The researcher may be used to recording for a study project, while the speaker is not. While native English speakers differ in their comfort reading and speaking, there are significantly less cultural factors that might contribute to misunderstanding in case of the researchers being native speakers of English.

The problem with eliciting natural speech has received a lot of attention in the field of linguistics, particularly in sociolinguistics, where entire studies can be based on the data for only several speakers, which makes it critical that the speech collected accurately represents the natural speech patterns of the speaker being studied. In 1976 Wolfson characterized an idea for characteristic discourse being legitimately comparable to that of suitable speech; as not comparable to unselfconscious discourse." Her proposition raises a few circumstances, when normally we speak carefully, which cautious discourse in such settings ought to not be considered unnatural.

Chapter 3 Dual supervised learning

3.1 Introduction

My research hypothesis states that it is possible to create a method that uses unlabelled datasets of two speech-related domains: speech samples without corresponding transcripts and text corpora without corresponding speech samples, to train speech recognition classifiers in a way which is as efficient and accurate as training methods provided by traditional solutions. The unlabelled data used in my method is far cheaper and easier to obtain and it usually comes in larger amounts than labelled data required by the traditional methods that have been widely used until now. The methodology used in my experiments is based on the dual supervised learning (DSL) technique ([33]). It exploits the fact that speech recognition and speech synthesis are complementary to each other.

DSL is a concept introduced by Xia ([33]). It is founded on the idea that many supervised learning tasks may be divided into two categories (e.g. English-to-French and French-to-English translation, speech recognition and synthesis, image classification and image generation, etc.). Because their models have a probabilistic association, the dual tasks are intrinsically linked.

To take advantage of the duality, a novel learning scheme including two tasks - a primary task and a secondary task - can be devised. The primary job uses a sample from space X as input and maps it to space Y, whereas the secondary work uses a sample from space Y as input and maps it to space X. Using the language of probability, the primal task learns a conditional distribution $P(y|x;\theta_{xy})$ parameterized by θ_{xy}, and the dual task learns a conditional distribution $P(x|y;\theta_{yx})$ parameterized by θ_{yx}, where

$x \in X$ and $y \in Y$. In the new scheme, the dual tasks are jointly learned and their structural relationship is exploited to improve the learning effectiveness.

DSL for machine translation has been successfully handled ([33]). Using the dual features of the issue, the researchers demonstrated that a similar method may be used to train a fully functioning and accurate translation system. This technique was altered and used to the domains of text and sound in this research.

The concept is built on reinforcement learning algorithms, which don't require as much data as supervised learning requires. It just requires two unlabelled datasets. A set of speech recordings by non-native speakers of the L language who are of the N nationality is one dataset. The text corpora of the L language are the second.

To train two different models, it will make use of the simple access to unlabelled datasets. A language model (ML) is one of them. It is purely based on a text corpus. There are two functions that must be present: 1. The ability to produce a new phrase in that language in textual form; 2. Possibility of estimating a probability score in that language for a given phrase (i.e. how natural a given sentence is, according to the language model).

The second model is an acoustic one (M_S). It is created with only unlabelled speech recording datasets. I would like it to have a similar functionality as the first model, but for the speech domain. Namely, it is necessary for it to be able to synthesize a new recording from the represented sound distribution as well as estimate the probability score for a given sound sequence, saying how accurately the sound sequence can be recognized as speech according to the acoustic model. During the distinct tasks of language modeling and acoustic modeling, the two models were trained individually, in isolation from any other models. At this time, the DSL approach had not been employed. Only unlabelled datasets were used in the training procedures of the two

models. A text corpus was utilized in the language model. A set of speech recordings was used for the acoustic model. Following training, each of those models could create a random sample from the taught probability distribution and evaluate a sample's likelihood score in relation to the learned probability distribution. The sample becomes a textual phrase in the language model, and a soundwave, a recording of a voice, in the acoustic model.

The setup of this method contains two more models. The first one is a speech recognition model (M_{STT}), which can recognize phonemes for a given sound sequence. The other one, complementary to the first, is a speech generation model (M_{TTS}), with the functionality of generating a speech signal for a given textual sentence. In the process of training using the DSL approach (described later), these two latter models (M_{STT} and M_{TTS}) will be the only trainable ones. They will be initialized by means of either a warm-start or a semi warm-start mode, and will have their weights updated according to a gradient descent-based algorithm.

The two former models (M_L and M_S) were trained before starting the DSL-based training process. They were trained in isolation from any other models, using unlabelled datasets. Therefore, they did not take part in the DSL-based training process.

The method uses all four aforementioned models, closed in a feedback loop.

Each model's function is critical since it is in charge of either synthesizing fresh data samples, assessing the outcomes of the preceding model, or transferring data between the textual and auditory domains. After that, it was discovered that both language and acoustic models may create a new data sample in the form of a textual phrase or a recording. Furthermore, they may use the learnt probability distribution of data in their domain to estimate the accuracy (by assigning a likelihood score) for a particular sample (either a text corpus or recording datasets). As a result, these models can

provide feedback to the model that converts data across domains, allowing the model to acquire weights that will result in improved (in terms of the feedback-giving model) conversion outcomes during the following training iteration.

The methodology for linguistic modelling that supports the ASR process was also developed. This linguistic model provides an additional text-based conversion mechanism between the results of a ASR model which, while might have managed to correctly recognize words, the original textual content of the sentence might have been produced with some impact related to the speaker's linguistic background. The linguistic model that perform the said conversion is based on the encoder-decoder setup, composed of recurrent neural network layers.

3.2 Method overview

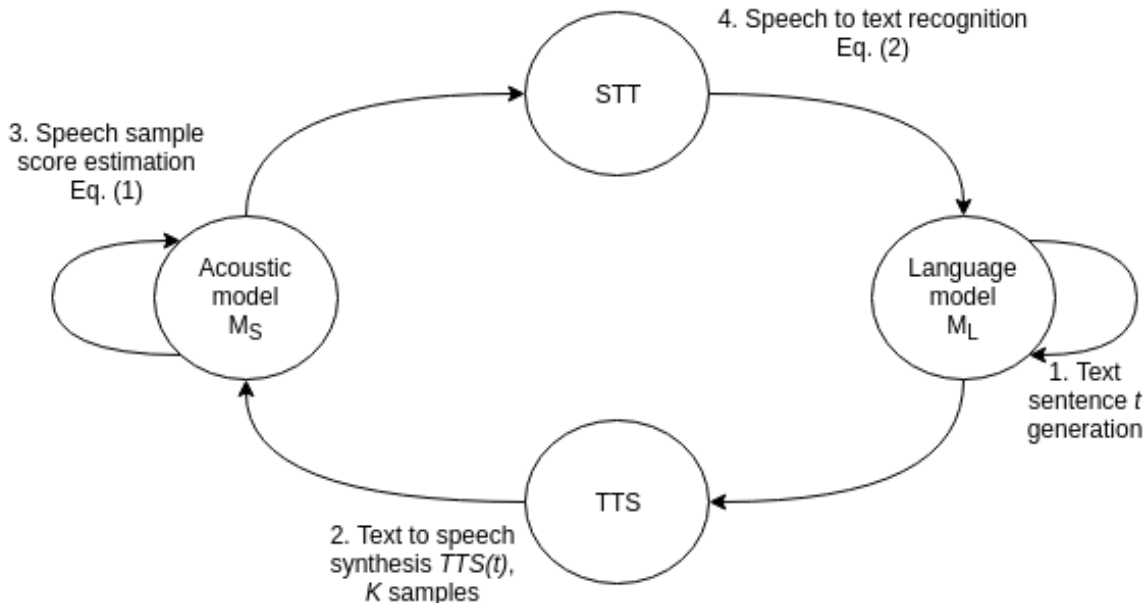

Figure 2: Feedback loop L design

In the process of training, two kinds of loop were used.

The first type (called loop L) is depicted in Figure 2. The loop begins from a language model M_L generating a t sentence in text form.

Then, speech generation model (M_{TTS}) generates sound samples which can potentially represent how the t sentence may sound when pronounced, according to M_{TTS}. M_{TTS} generates K different soundwaves $TTS(t)_k$ from t sentence, using a beam search algorithm.

The third step is a probability estimation for each of the K generated samples. This is achieved by utilizing the acoustic model M_S. The score for each sample equals

$$a_k^{im} = M_S(TTS(t)_k) \tag{1}$$

where:

a_k^{im} – immediate reward score for k sample of soundwave *TTS(t)* for loop L

$M_S(TTS(t)_k)$ – likelihood score for k sample

This says how "probable" it is that the synthesized recording could be an actual speech sample in a particular language.

Lastly, the speech recognition model (M_{STT}) transfers a previously synthesized sample $TTS(t)_k$ into textual form. At this step, a probability score for each of the K synthesized samples was also calculated and it says how correctly the M_{STT} model recognizes the k sample as the original sentence t. The score equals

$$a_k^{lt} = logP(t|TTS(t)_k; M_{STT}) \tag{2}$$

where:

a_k^{lt} – long term reward score for k speech sample of *TTS(t)* for loop L

$P(t|TTS(t)_k; M_{STT})$ – probability score for receiving sentence t from k speech sample *TTS(t)*, when recognizing using M_{STT}

The second kind of loop is similar to the first, but starts at another point. It is shown in Figure 3, and is called loop S.

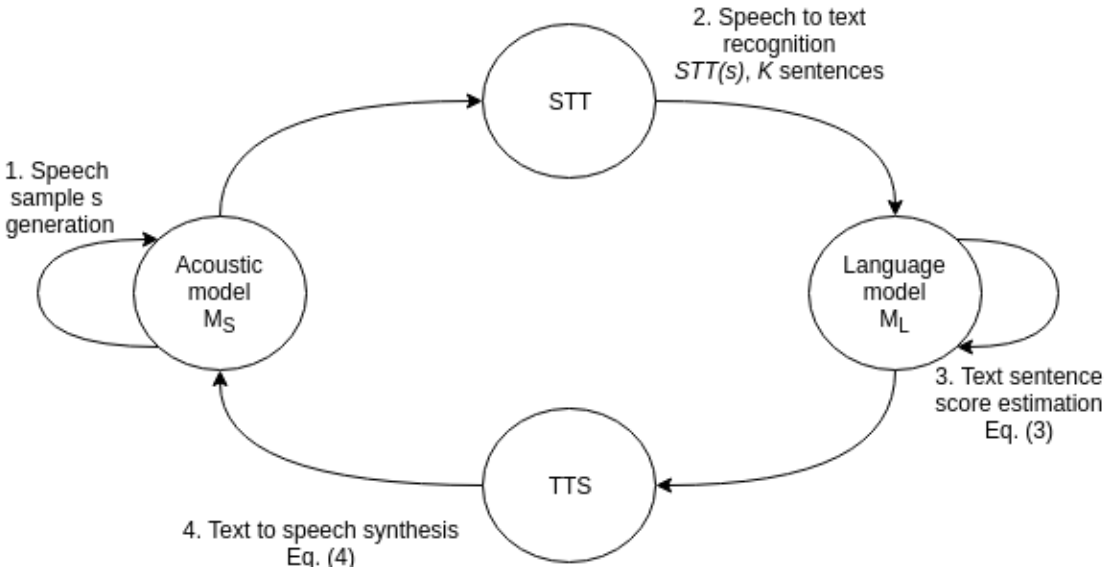

Figure 3: Second kind of loop (loop S)

This loop begins from the acoustic model M_S generating a speech sample s.

Then, M_{STT} recognizes a generated sample as textual sentences which are potentially transcripts for s sample, according to M_{STT}. M_{STT} produces K most probable sentences $STT(s)_k$ from s sample, also using a beam search algorithm.

The third step is a probability score estimation for each of the K recognized sentences. This is achieved by applying the language model M_L. The score for each sentence equals

$$b_k^{im} = M_L(STT(s)_k) \tag{3}$$

where:

b_k^{im} – immediate reward score for k sentence of *STT(s)* for loop S

$M_L(STT(s)_k)$ – likelihood score for k sentence

This says how "probable" it is that the recognized sentence could be an accurate sentence in a particular language.

Lastly, the M_{TTS} model synthesizes the previously recognized sentence $STT(s)_k$ into speech form. At this step, the probability for each K - recognized sentence was also calculated. The probability gives information on how correctly the M_{TTS} model generates a speech sample for k sentence with s being the original sample. The score equals

$$b_k^{lt} = logP(s|STT(s)_k; M_{TTS}) \tag{4}$$

where:

b_k^{lt} – long term reward score for k sentence of *STT(s)* for loop S

$P(s|STT(s)_k; M_{TTS})$ – probability score for receiving speech sample s from k sentence *STT(s)*, when synthesized using M_{TTS}

One iteration in the learning process contains the single performance of both aforementioned loops. After the iteration is completed, a pair of scores (a_k^{im}, a_k^{lt}) for each of the K generated speech samples and a pair of scores (b_k^{im}, b_k^{lt}) for each of the K recognized sentences are produced.

The scores are then used in a policy gradient algorithm as immediate rewards (a_k^{im} and b_k^{im}) and long term rewards (a_k^{lt} and b_k^{lt}). The total reward for the k sentence (or

sample), is set as

$$a_k = \alpha a_k^{im} + (1-\alpha) a_k^{lt}$$
$$\text{or} \tag{5}$$
$$b_k = \alpha b_k^{im} + (1-\alpha) b_k^{lt}$$

where:

$\alpha -$ a factor specifying the weight of the immediate reward in the DSL approach

Having done that, the problem can be formulated as optimizing the a_k and b_k functions. As described before, this function will be optimized by modifying the weights of two trainable models M_{STT} and M_{TTS}. Gradient-based methods of optimization was used. Gradients of the estimator of the total reward's expected value with respect to those models were calculated. In Eq. (6) and Eq. (7) the calculation for loop L (loop starting from the language model) is depicted. The calculations for loop S are analogical.

$$\bigtriangledown_{M_{TTS}} E[a_k] =$$
$$E[a_k \bigtriangledown_{M_{TTS}} \log P(TTS(t)_k | t; M_{TTS})] \tag{6}$$

$$\bigtriangledown_{M_{STT}} E[a_k] =$$
$$E[(1-\alpha) \bigtriangledown_{M_{STT}} \log P(t | TTS(t)_k; M_{STT})]$$

$$\nabla_{M_{TTS}} \hat{E}[a] = \frac{1}{K} \sum_{k=1}^{K} [a_k \nabla_{M_{TTS}} \log P(TTS(t)_k | t; M_{TTS})]$$

$$\nabla_{M_{STT}} \hat{E}[a] = \frac{1}{K} \sum_{k=1}^{K} [(1-\alpha) \nabla_{M_{STT}} \log P(t | TTS(t)_k; M_{STT})] \tag{7}$$

where:

$E[a_k]$ – expected reward for a k sample

$\nabla_{M_{TTS}} E[a_k]$ – gradient of the expected reward per k sample, with respect to M_{TTS} model

$\nabla_{M_{STT}} E[a_k]$ – gradient of the expected reward per k sample, with respect to M_{STT} model

$\nabla_{M_{TTS}} \hat{E}[a]$ – gradient of the expected reward, with respect to M_{TTS} model

$\nabla_{M_{STT}} \hat{E}[a]$ – gradient of the expected reward, with respect to M_{STT} model

After calculating the gradients, models M_{STT} and M_{TTS} are updated according to the following formulas:

$$M_{TTS} = M_{TTS} + \eta_{TTS} \nabla_{M_{TTS}} \hat{E}[a]$$
$$M_{STT} = M_{STT} + \eta_{STT} \nabla_{M_{STT}} \hat{E}[a] \tag{8}$$

$$M_{TTS} = M_{TTS} + \eta_{TTS} \nabla_{M_{TTS}} \hat{E}[b]$$
$$M_{STT} = M_{STT} + \eta_{STT} \nabla_{M_{STT}} \hat{E}[b] \tag{9}$$

where:

η_{TTS} – learning rate for M_{TTS} model

η_{STT} – learning rate for M_{STT} model

After one iteration is complete, another oneis started, containing both types of loops, and starting from M_L and M_S generating different samples from learned distribution. The proposed DSL process is depicted in Figure 4.

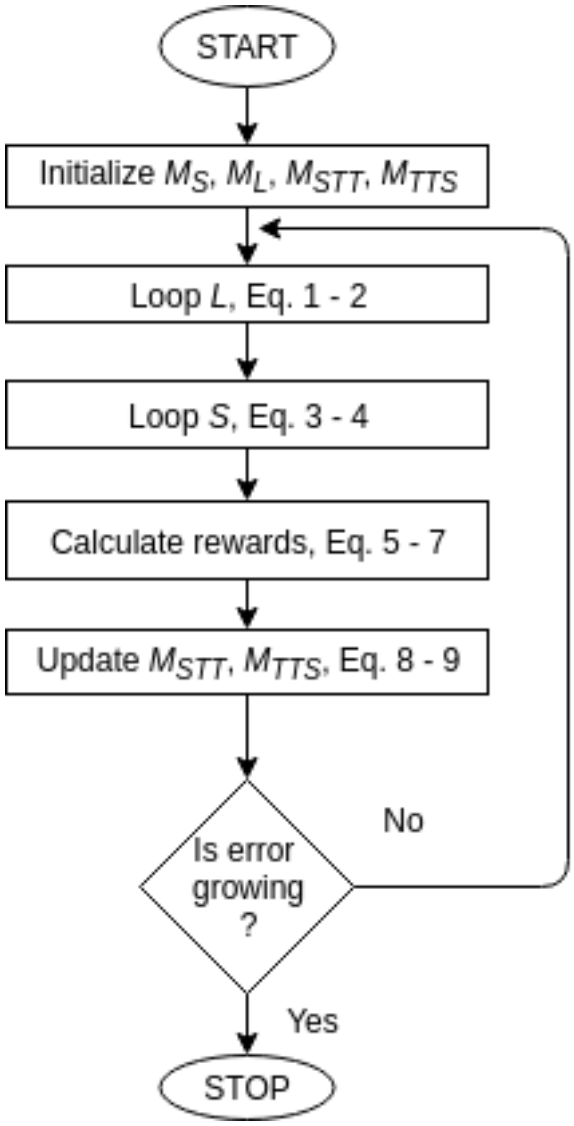

Figure 4: DSL process for non-native speech recognition

In this feedback loop setup, both M_{TTS} and M_{STT} models are trained. For the purpose of non-native speech recognition M_{STT} and its accuracy are the top priority. After the training process, the speech recognition model M_{STT}, adjusted to pronunciation features of particular non-native speakers, will be created. Also, M_{TTS} as a speech synthesizer becomes a by-product of the training process. It produces speech biased to the pronunciation patterns of non-native speakers of the language that was in the training dataset.

3.3 Experiments

3.3.1 Algorithms chosen and tested for each model

Several algorithms for each model were chosen and tested how well the DSL methodology acts in different setups.

Language models M_L in the approach:

- vanilla recurrent neural network (RNN),
- RNN with a long short-term memory (LSTM) cell,
- 3-gram model.

The RNN and LSTM language models were created on a character level. A single one-hot encoded row of data which was fed to the network during training was related to one particular character. Then again, during inference time, the network was also fed data samples related to one character. On the other hand, the 3-gram model operates on trigrams of consecutive characters. One aspect of the research was testing whether DSL methodology could be applied and actually useful in different kinds of

setups, with different kinds of architectures for each model. That was why a 3-gram based language model was used in one of the experiments ([34], [35]).

For acoustic model M_S, the following models were chosen:

- vanilla RNN,
- RNN with an LSTM cell.

The speech recognition models M_{STT} are as follows:

- vanilla RNN,
- RNN with an LSTM cell.

Only Deepmind's Wavenet as speech synthesis model M_{TTS} was examined, because speech synthesis was not a primary issue to address in this research.

Table 1 depicts an overview of the architecture of the models.

Table 1: Model architecture for each setup

setup	M_L	M_S	M_{STT}	M_{TTS}
1	RNN 3x512	RNN 3x512	RNN 2x1024	Wavenet
2	LSTM 3x512	LSTM 3x512	LSTM 2x1024	Wavenet
3	3-gram	LSTM 3x512	LSTM 2x1024	Wavenet

Three different setups of the above models were tested. The architecture of the models in these setups was chosen using local search algorithms in isolated tasks of language modeling, acoustic modeling and speech recognition.

The first setup (setup 1) contained a 3-layer vanilla RNN with 512 hidden units per layer, for the language model. The same network was used for the acoustic model. As per M_{STT} model, a 2-layer RNN was chosen, with 1024 hidden units per layer.

Descriptions for setup 2 and setup 3 are analogical to setup 1 and are shown in Table 1.

The reason for choosing the RNN-based neural networks (vanilla RNN and RNN with an LSTM cell) is their performance results on the type of datasets being used in this research. The datasets represented by textual and acoustic domains contain sequences of interdependent data samples. The letters (or words) in any textual sentence that belongs to any text corpus are not to be understood as completely independent of each other. There are sequences of letters where the former ones have a significant impact on which letter may appear as a latter one. Analogical sequential dependency exists in the acoustic domain. An RNN is a straightforward adaptation of the standard feed-forward neural network to allow it to model sequential data. At each timestep, the RNN receives an input, updates its hidden state, and makes a prediction. The RNN's high dimensional hidden state and nonlinear evolution enable the hidden state of the RNN to integrate information over many timesteps and use it to make accurate predictions. Even if the non-linearity used by each unit is quite simple, iterating it over time leads to very rich dynamics. The standard RNN is given a sequence of input vectors, then it computes a sequence of hidden states and a sequence of outputs. A RNN with an LSTM cell addresses the exploding and vanishing gradient problem, therefore making it possible to track long-time dependencies in the sequential data ([36], [37], [38], [39]).

The aforementioned Wavenet model was designed in a similar manner to Deepmind's original Wavenet ([40]). The general idea of the model is to predict the audio sample based on the series of previous audio samples. In order to realize the actual functionality of *TTS*, following the authors' method, I added the possibility to condition the model's prediction locally, on the textual sentence corresponding to a speech

sample. In the experiments I decided to use the Wavenet that consists of three stacks of dilated layers (10 layers per stack, dilation rate up to 512) and two fully connected layers. Other parameters included a filter width of 2, 32 residual channels, 32 dilation channels and 256 quantization channels.

3.3.2 Types of experiment performed

The purpose of the conducted experiments was to confirm the hypothesis described in 3.1 as well as to estimate the accuracy of this method on different setups. In order to assess the quality of this methodology, I designed several experiments.

In the first experiment, I decided to check and evaluate the influence of a warm start on the overall accuracy of M_{STT} model. Warm start refers to a training mode where M_{STT} and M_{TTS} models are initially trained with a small amount of labelled data, before I start to train them in a dual supervised manner.

In the second experiment, I checked a semi warm-start approach, training only M_{STT} model with a small amount of labelled data before switching to DSL.

These two experiments were conducted for each of the three model setups.

The last experiment I conducted became a baseline method in this research. This baseline experiment does not make any use of the method I present in this research but instead uses the traditional supervised learning approach, where there is only one, fully labelled dataset.

In this case I trained a 2-layer RNN with 1024 hidden units in a LSTM cell as M_{STT} model, in a traditional way. In this approach, I trained an end-to-end speech recognition setup that consisted of one network performing conversion from the acoustic domain to the textual one. Because I decided to use the end-to-end model, I used a Connectionist Temporal Classification (CTC) loss function. This loss function does

not require a frame-level alignment (matching each input frame to the output token). Therefore, it allows the use of the labelled speech datasets, without the need to align the text with the soundwave frames ([41], [42], [43], [44], [45], [5]).

There was only one model (M_{STT}) in the whole setup, and it was trained in the experiment. I performed the training in a normal, supervised manner, using only a labelled dataset, so that I can show that the results of this traditional approach and the DSL-based one (from previous experiments) are actually comparable.

3.3.3 Datasets used in the experiment

I conducted the first two above-mentioned experiments on two cases of Japanese and Polish people pronouncing English sentences.

For training language models M_L I used the Corpus of Contemporary American English (COCA) described in Appendix A.4.

For training acoustic models M_S, I used pieces of recordings scraped from Youtube website resources (mostly either Japanese people teaching Japanese to an English audience, or Japanese expatriates living abroad and creating videos in English). The same source was used in the case of Polish people pronouncing English sentences.

During the warm-start and semi warm-start approach, for training M_{STT} and M_{TTS} models I used 10% (around 7000 recordings) of the *English Speech Database Read by Japanese Students (UME-ERJ)* (Appendix A.1) for Japanese people, and 10% (a similar quantity) of recordings scraped from Youtube for Polish speakers, which I labelled myself. The rest of the data was used for verification.

In the last, baseline experiment, I used only the *UME-ERJ* dataset since the amount of time necessary to label the whole scraped dataset for Polish people pronouncing English was too long. In this case I used 80% (around 56000 recordings), 10% (around

7000) and 10% of data as training, validation and testing sets respectively.

A random shuffle strategy was used for selecting each subset of training, testing and validation sets.

3.3.4 Evaluation of DSL method accuracy

As measure of error, I chose character error rate, or length normalized character-level edit distance. Accuracy is obviously $1 - error$.

Since there are not many popular benchmarks for ASR of either Japanese or Polish pronunciation of English sentences, I decided to evaluate the DSL approach for the speech recognition problem by comparing the accuracy result of M_{STT} model created using the methodology described in my paper (DSL) to the accuracy of M_{STT} created using the traditional approach, based on supervised learning (the last of the conducted experiments). In this way I show that the result yielded by the DSL methodology is comparable to the one achieved by the traditional method. Having said that, I state that the result achieved in the last experiment becomes a baseline result, against which I compare the results from the first two experiments.

3.3.5 Results

The results of experiments are presented in the table below. The scores show the best accuracy of M_{STT} model that I managed to obtain during the training process. In order to make the results more reliable, each of the scores shown in the table is an averaged score of six runs of any particular setup. As stated before, during each run, the datasets used for training, testing and validation were chosen using the random shuffle strategy.

Below I present how the error rates changed during the training time for the warm-

Table 2: Results of conducted experiments for setup 1

	English by Japanese	English by Polish
Warm-start	84.12%	83.23%
Semi warm-start	82.92%	81.04%

Table 3: Results of conducted experiments for setup 2

	English by Japanese	English by Polish
Warm-start	89.43%	88.14%
Semi warm-start	88.21%	86.92%

Table 4: Results of conducted experiments for setup 3

	English by Japanese	English by Polish
Warm-start	86.43%	84.51%
Semi warm-start	85.21%	83.92%

Table 5: Results of conducted experiments for the traditional method (baseline)

	English by Japanese
Traditional method	87.24%

start approach (Figure 5) with setup 2 and the traditional approach (Figure 6) for English pronounced by Japanese people.

The warm-start approach chart clearly reflects the moment when I switch from (initial) pre-training to DSL (around the 130th epoch). The convergence rate for the M_{STT} model declines from that point. That means more time is required to achieve comparable results. However, the final accuracy achieved by the warm-start DSL approach is higher.

Even though the DSL method yields better results, they are achieved at a cost of training time. On average, a single run of an experiment using the traditional method took us four days to complete using a single GTX 1080 Ti graphics card. The average time needed for a single run of the DSL-based approach to finish was five weeks. However, the use of multiple cards allowed us to run the experiments in parallel, and,

consequently, to save time. Below, I depict the average necessary time, together with the number of epochs it took to achieve the best result.

Table 6: Average time necessary for training each setup

	setup 1	setup 2	setup 3	baseline setup
Time	5 weeks	5 weeks	5 weeks	4 days
Epochs	3000	2800	3200	380

While the time necessary for the DSL-based method to achieve the desired results is clearly much longer, it is still acceptable for the purpose of running the experiments and evaluating the methodology.

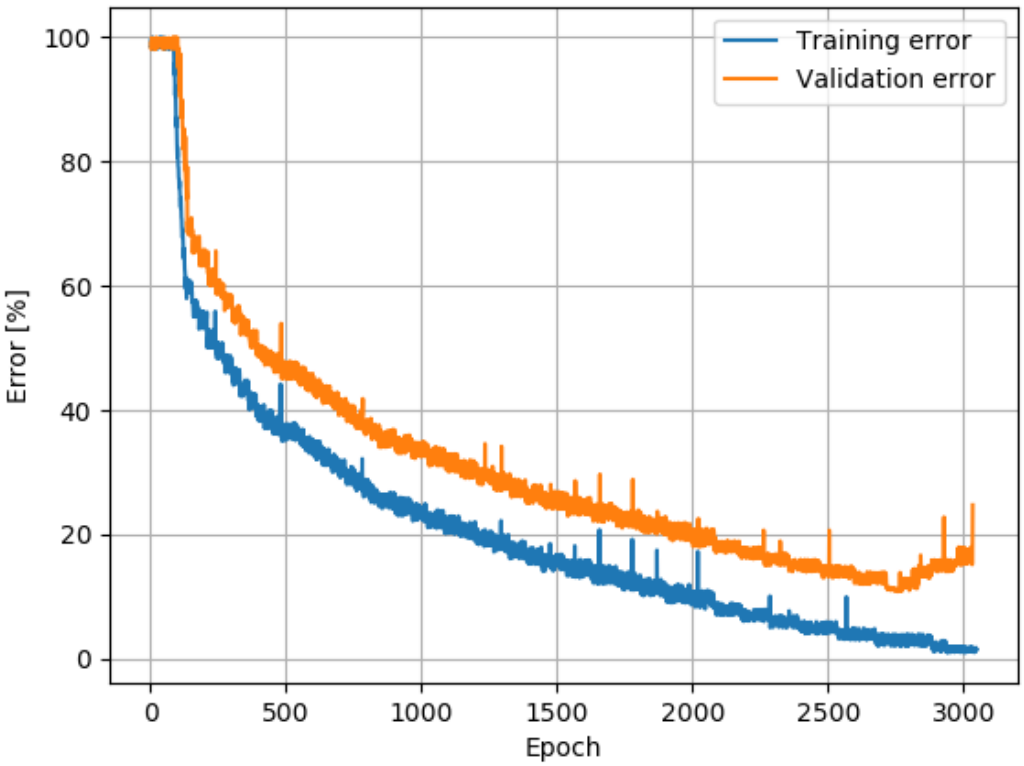

Figure 5: Training process of warm-start approach

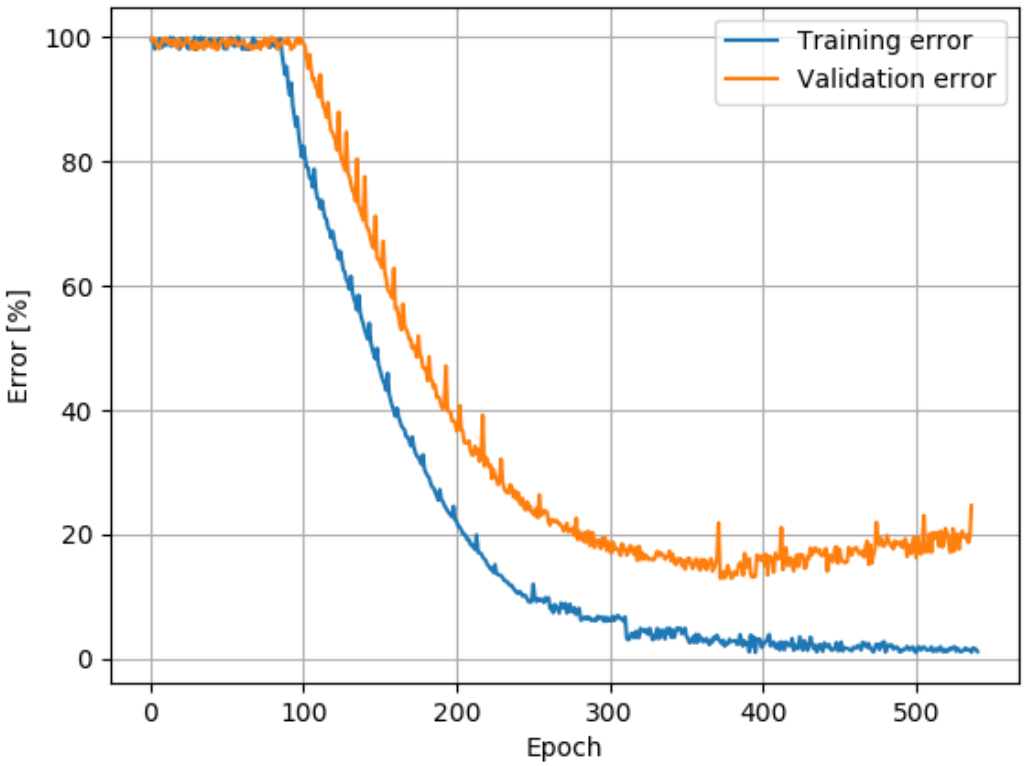

Figure 6: Training process of traditional approach

3.4 Proposed methodology for lexical model

Methodologies described in Sections 3.2 represent approaches to creating the non-native speakers adapted ASR systems so as the overall accuracy of the STT process can increase. Although I explained and proved the significance of the two methods by multiple experimental setup, there is still a lot of room for improvement in terms of non-native speech recognition.

Non-native speakers often introduce various patterns from their mother tongue into the second language they speak. These patterns are often visible in forms of accent, pronunciation and intonation, as well as grammar or choice of vocabulary. The method

described in Sections 3.2 operates only on the speech samples, without trying to utilize the information included in the speech, but on the textual level.

In this section I describe an additional methodology for post-ASR text result correction. It is a processing step which is supplementary to the algorithm for creating the ASR systems adapted for non-native speech as described in Section 3.2.

In this subsection I describe a methodology for text modification based on the potential differences in grammar, order and choice of words, spelling, semantics etc., between a sentence uttered by a non-native speaker and the way it would have been constructed by a native speaker. I created a language model able to visualize the correspondence between texts representing sentences uttered by a non-native speaker from one particular nationality (thus possibly containing errors or mother tongue-related patterns), and these sentences uttered by a native speaker. Effectively, a modelling technique was created, capable of representing conversion method of grammar and language-related patterns in one's mother tongue to corresponding sentences formed in a correct way of proper English language. An encoder-decoder network was created for the purpose of rearranging the order and choice of words in a sentence uttered by a non-native speaker and converted to text format by a ASR system.

The problem of such conversion can be understood as a sequence-to-sequence problem. Within the research performed with the purpose of finding the best language modelling algorithm, several architectures of encoder-decoder network with different base cells ($LSTM$ and GRU) as well as different number of layers in encoder and decoder models were evaluated.

After an isolated evaluation of the encoder-decoder algorithm, I evaluated the its impact on the ASR pipeline for non-native speakers of English. In this particular experiment I utilized the speech data of Japanese people pronouncing English sentences.

3.4.1 Sequence-to-sequence problem statement

The term "sequence modeling" refers to challenges in which the input or output are both sequences of symbols (e.g. letters, words). To illustrate an example let's consider a problem of machine translation, a chatbot that automatically replies to queries, or a movie captioning problem. Both input and output are sequences of tokens, here words or letters.

Sequence-to-sequence problems represent a perfect application of RNNs. Had the traditional neural networks been employed, it would have been necessary to encode the input sequence as a fixed-length sequence utilizing algorithms like e.g. BOW, Word2Vec. However, the order of words is not retained here, so when the input vector is fed into the model, it has no notion what order the symbols are in, thereby missing a critical information related to the input sequence. RNN is the algorithm that can address the aforementioned issues. Namely, for a given input $X = (x_0, x_1, ..., x_t)$ with a changeable features number, the process performed at each time-step is the RNN cell taking a token x_t as input and creates the output h_t while at the same time passing information vector to the next time-step. Depending on the task at hand, these outputs can be employed.

To tackle this kind of problems encoder-decoder models were originally employed.

3.4.2 The algorithm design of the encoder-decoder network

An encoder-decoder model can be thought of as two blocks, the encoder and the decoder, joined by a vector we'll call the *context vector* at a high level.

- **Encoder**: Each token in the input sequence is processed by the encoder. It aims to pack all of the information about the input sequence into a fixed-length vector

called the 'context vector.' The encoder provides this vector to the decoder after passing through all of the tokens.

- **Context vector**: The vector is designed to encompass the entire semantics of the input sequence and assist the decding block in making generating accurate sequences. This is the last internal stage of the encoder block.

- **Decoder**: This block scans the vector of context and attempts, token by token, to anticipate the target sequence.

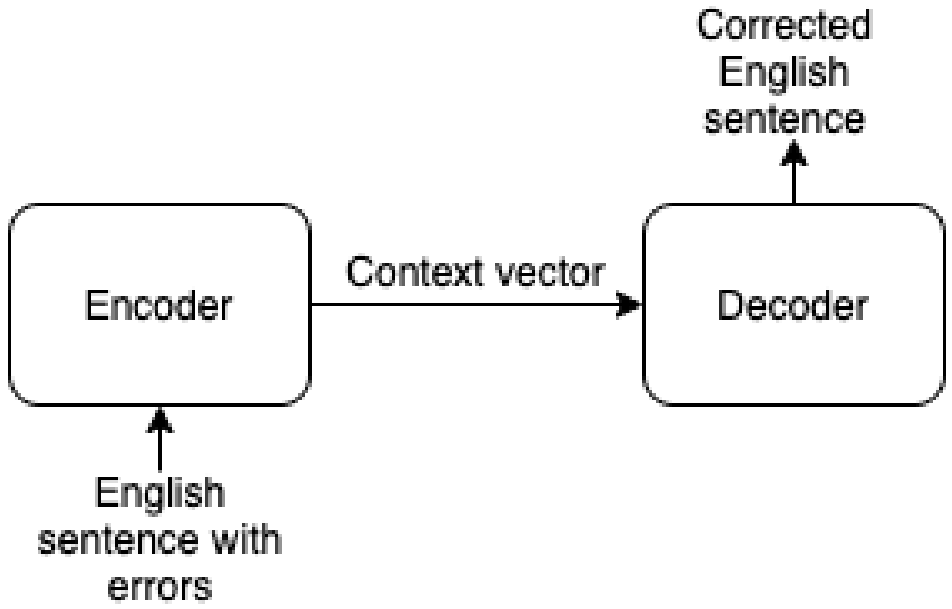

Figure 7: The basic diagram of encoder-decoder model

The internal structure of each block is reflected in Figure 8.

Two basic cells (in this research *LSTM* and *GRU* types of cells were used) with a connection in between, can encapsulate the encoder-decoder model. The most important part is the methodology of processing data between the blocks.

An LSTM cell serves as the encoder. The input sequence is fed into it over time, and the encoder tries to encompass and save the information and semantics as the final

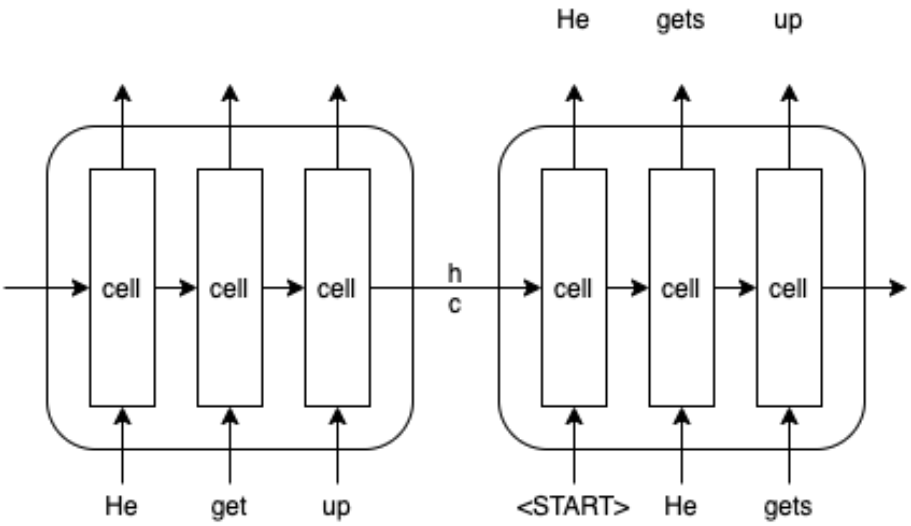

Figure 8: The internal structure of encoder-decoder blocks

form of internal states *ht* (hidden state) and *ct* (common state) (cell state). They are in turn later sent to the decoder, which attempts to generate the predicted sequence based on the passed information. This describes the mechanism of the *context vector* I mentioned about earlier.

The encoder part's outputs are entirely discarded at each time step.

The encoder delivers its internal states as input of the decoder after reading the entire input sequence, and this is where the output sequence prediction occurs. The decoder is an LSTM cell as well. The important thing to notice here is that the decoder's beginning states (*ht*, *ct*) are set to the encoder's end states (*ht*, *ct*). These serve as a context vector, assisting the decoder in generating the desired target sequence.

3.5 Experiments with linguistic modelling

3.5.1 Language model creation

Firstly, in order to create the model described in Section 3.4, the *NICT Japanese Learner English (JLE) Corpus* dataset described in details in A.3 was used.

The training of the model using the *NICT Japanese Learner English (JLE) Corpus* dataset described in details in Section A.3, was performed.

3.5.2 Datasets used

The dataset used in this part of the research is the *NICT Japanese Learner English (JLE) Corpus*, described in detail in Appendix A.3.

It represents a set of samples each of which consists of a sentence produced by a non-native speaker, thus potentially containing some mistake, and the label which is the same sentence but with corrections and fixes introduced by a native speaker expert. In this way, the dataset can visualize the correspondence between erroneous sentences produced by non-native speakers and their respective corrected versions, that could have been produced by native speaker of English.

Dataset preparation The dataset contains 15637 samples fully labelled. For the purpose of experiments, I divided randomly into three parts: training, validation and test, in 8:1:1 ratio respectively.

3.5.3 Experiments and metrics

As performance metric of the model, CER was chosen. Training of the model was then executed and kept monitoring the validation accuracy of the network. The results

shown in Table 7 depict the CER values averaged 10-fold cross validation.

Table 7: Results of language model training

Encoder	Decoder	Validation set result [CER]
LSTM x 1	LSTM x 1	87.3%
LSTM x 2	LSTM x 2	89.8%
LSTM x 3	LSTM x 3	90.1%
GRU x 1	GRU x 1	88.4%
GRU x 2	GRU x 2	89.9%
GRU x 3	GRU x 3	91.2%

3.6 Summary

Convergence point Training two networks in such a way that both models learn from one another can bring the risk of the models converging to a point that is not desired. For instance, in the speech recognition and speech synthesis domain, M_{STT} and M_{TTS} models were used. There is a possibility that M_{TTS} may learn pronunciation of a different w word (or sentence), while the language model M_L comes up with a completely different t word. Yet, the immediate reward associated with $M_S(TTS(t))$ may be actually significant since the pronunciation itself is correct according to the acoustic model. If this happens, there is a risk of the M_{STT} model learning to associate the pronunciation of w word with a textual form of t word. The learning process will try to maximize the long time reward associated with $logP(t|TTS(t); M_{STT})$, and in such an event the M_{STT} model understands that t word becomes a label for an incorrect $TTS(t)$ speech sample (which was mistakenly generated by M_{TTS} earlier). This may lead to a situation where both models learn the incorrect association between speech features and text sentences. Particularly, M_{TTS} can learn the incorrect distribution of $P(TTS(t)|t)$ (i.e. it can learn distribution which would normally represent w sentence). A similar situation may occur for M_{STT} model.

Warm start and its influence Pre-training, or the warm-start approach in a chosen methodology, is helpful for preventing models from learning incorrect associations between speech features and text sentences. It is very useful for speeding up the learning process and increases the chance of achieving a desired convergence point as it provides a good starting position for the optimization algorithm. Due to the application of pre-trained M_{STT} and M_{TTS} models, the DSL process is started from the point where distributions of $P(TTS(t)|t)$ and $P(STT(s)|s)$ are partially learned from the labelled dataset. Assuming the correctness of the dataset itself, the distributions are correct, but do not represent the full feature space yet.

As the algorithm shifts from pre-training using labelled datasets into DSL, M_{STT} and M_{TTS} models could expand previously learned distribution using unlabelled data while the learning process continues.

This allows us to both make use of a vast amount of unlabelled datasets and make sure the models are converging towards a desired direction.

Warm start with only one of two pre-trained models According to the results of the experiments it appears that the warm start with both models initially trained is not a prerequisite for the models to be correctly trained. One pre-trained model is enough for the whole setup to achieve a desired convergence point.

In this research the problem of non-native speech recognition was shown as well as the issues that appear if traditional approaches for building ASR systems for such cases were used.

I also described in detail the idea behind DSL methodology and explained why this method is suitable for solving this problem.

Then experiments were executed, employing different algorithms in different setups,

in order to show that DSL methodology can produce ASR systems with an accuracy comparable to currently used ASR products, while at the same time making use of far cheaper and larger unlabelled datasets.

Warm-start and semi warm-start approaches were tested, and the results of experiments show that they work well. However, until the solution to the non-native speech recognition problem in a fully unsupervised manner (without warm start) has been developed, there is still room for improvement.

Lexical modelling In this chapter a supplementary method of language modelling adapted for non-native speakers was also designed. The method is designed to provide an additional correction of ASR system results which is used by a non-native speaker of a particular language. In this particular case I focused on people of Japanese nationality pronouncing utterances in English.

The designed language model has been plugged into this pipeline right after the dual supervised learning algorithm for creating an ASR system. The impact of my language model on the textual result from the ASR system, was examined. It was shown that such additional language model contributes significantly to the accuracy of the whole designed pipeline.

While the method proved to be effective in case of one specific language and once specific non-native nationality, it may not always be feasible for real-time use cases. The reason behind this is the necessity to establish a clearly labelled dataset consisting of corresponding sentences produced by a non-native speaker and corrected sentences, produced by a native speaker, in a form described in Section A.3. Depending on the combination of language and respective non-native nationality, it might be difficult to compose a sufficiently large and varied dataset with aligned sentences. The ex-

isting aligned datasets might be too small to represent necessary lexical information, necessary for creating a well-performing conversion model. Because of that it might be reasonable to use a different method for training a encoder-decoder described in Section 3.4.

In Section 3 a method suitable for training ASR models for cases when the labelled datasets are extremely scarce, was described. The method has been developed with the purpose of training ASR models specialized for non-native speech recognition. The approach used was an extension of dual supervised learning methodology, adapted for speech and voice field. A similar way of approaching a problem could be applied in case of training encoder-decoder model using scarce textual corpora of non-native and native texts. Namely, I could state the problem of creating encoder-decoder algorithm between native and non-native textual corpora as a dual problem consisting of two smaller: a conversion from non-native text to native one and the other way. By doing so it is possible to leverage the much more vast resources of not labelled datasets for the purpose of creating a sufficiently accurate conversion mechanism.

Chapter 4 Audio style transfer for on-the-fly speech correction

4.1 Introduction

In this chapter I present an approach for handling the problem related to a specific, non-native accent. I created a method that modifies the accent of a non-native speaker so that it resembles the accent of a native speaker to a higher extent. The purpose of this method is to increase the accuracy of ASR systems which had already been developed and trained using a native speech dataset, without the necessity to train new ASR models adapted for a specific non-native accent.

Our idea is to modify the accent of speech using the representation of a sound wave in a graphical domain, i.e. a spectrogram.

The general flow of our approach is depicted in Fig. 9.

I plan to tackle the problem of non-native accent using the style transfer methodology ([46, 47]) adapted for the case of speech. The application of style transfer in audio domain is not new. In [48] authors investigated how to transfer the style of a reference audio signal to a target audio content. They proposed a flexible framework for the task, which uses a sound texture model to extract statistics characterizing the reference audio style, followed by an optimization-based audio texture synthesis to modify the target content. In contrast to mainstream optimization-based visual transfer method, the process proposed by the authors is initialized by the target content instead of random noise and the optimized loss is only about texture, not structure.

At the beginning I transform a sound wave file into a spectrogram (the process

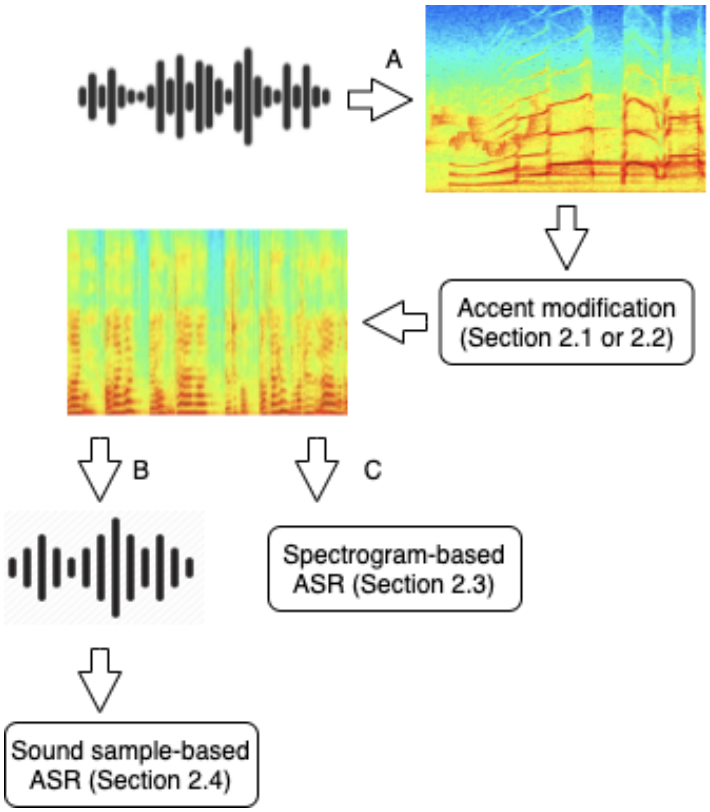

Figure 9: Overall diagram of the solution. A - conversion from sound sample to a spectrogram, B - conversion from spectrogram to a sound sample, C - spectrogram-based ASR

indicated as **A** on the diagram). Secondly, accent modification is performed. Within the second step I decided to check two ways of modifying accent with spectrograms and they are described in detail in Sections 4.2 and 4.3.

Finally, the sound wave, in modified form, is fed into the ASR system in order to recognize the speech into text. In our research I experimented with two kinds of speech recognition process. As shown in the figure, one way is to revert the modified spectrogram back to the sound wave (the process indicated as **B** on the diagram) in a form of WAV file and then feed it to a previously agreed ASR system. The second way indicated as **C**) is to feed the spectrogram directly to another ASR system (adapted for recognizing the speech from spectrograms) created within this research (Section 4.4).

The accent modification algorithms are trained in a way that they learn how to modify the spectrogram representing non-native speech to one resembling the same utterance by a native speaker. The correctness of the modified speech is determined by the accuracy of the speech recognition system trained on a dataset containing native speaker samples, allowing us to evaluate the quality of accent modification. The criterion which decides whether or not the style-modifying algorithm can perform well is a reduction in the metrics related to the error yielded during the inference using the ASR networks, which were trained on native speaker samples.

4.2 Accent modification using autoencoder

In this approach I came up with an autoencoder based on a convolutional neural network (CNN). Our idea is to employ such a network for the purpose of changing the pronunciation style.

The autoencoder was written using the Keras library ([49]), and its detailed architecture is described in Table 8.

During the training phase the autoencoder is fed spectrograms of samples of non-native speakers, whereas the autoencoder's output is compared against the spectrograms of exactly the same utterances pronounced by native speakers of the particular language. Then back-propagation causes the autoencoder to learn the conversion of the same words and sentences from the speech containing a non-native accent to the one with the modified accent.

During the inference phase the input spectrogram created in the first step of our pipeline is fed into the autoencoder as input data. The output of the autoencoder is a spectrogram which is slightly converted according to the CNN layers' weights learned

after the training.

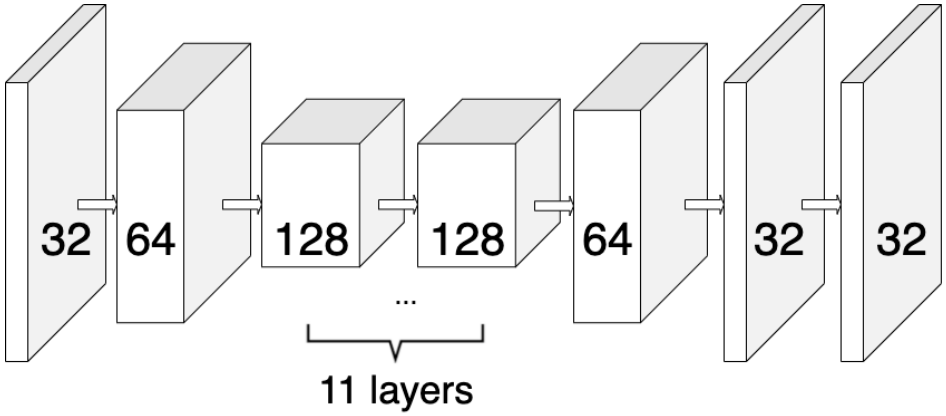

Figure 10: Autoencoder schema. Simple diagram of the CNN-based autoencoder. The numbers represent the filters count per each layer

Abbreviations used in Table 8:

$Conv2D$	– 2-dimensional convolutional layer,
$Conv2DTr$	– 2-dimensional convolutional transpose layer,
F	– number of filters,
K	– kernel size,
B	– batch dimension,
X	– dimension related to spectrogram's frequency,
T	– dimension related to spectrogram's time steps

4.3 Accent modification using style transfer-based approach

Another approach I decided to experiment with employs a style transfer methodology adapted for the domain of speech and sound. Specifically I decided to create a method that resembled the style transfer feedforward algorithm from the graphical domain.

To briefly explain the problem of the graphical style transfer: I try to modify an

Table 8: Detailed architecture of the autoencoder

Layer	Output shape	Parameters
Conv2D (F=32, K=3)	B, X, T, 32	417344
Conv2D (F=64, K=3)	B, X, T, 64	18496
Conv2D (F=128, K=3)	B, X, T, 128	73856
Conv2D (F=128, K=3)	B, X, T, 128	147584
Conv2D (F=128, K=3)	B, X, T, 128	147584
Conv2D (F=128, K=3)	B, X, T, 128	147584
Conv2D (F=128, K=3)	B, X, T, 128	147584
Conv2D (F=128, K=3)	B, X, T, 128	147584
Conv2D (F=128, K=3)	B, X, T, 128	147584
Conv2D (F=128, K=3)	B, X, T, 128	147584
Conv2D (F=128, K=3)	B, X, T, 128	147584
Conv2D (F=128, K=3)	B, X, T, 128	147584
Conv2D (F=128, K=3)	B, X, T, 128	147584
Conv2DTr (F=64, K=3)	B, X, T, 64	73792
Conv2DTr (F=32, K=3)	B, X, T, 32	18464
Conv2D (F=32, K=3)	B, X, T, 32	82976
	Total params:	2,160,768
	Trainable params:	2,160,768
	Non-trainable params:	0

image in a way that its style resembles the style of another, a so-called style image. At the same time the content of the transformed image ideally should not be modified.

The general flow of the accent modification using style transfer is depicted in Fig. 11.

In order to utilize such a setup I first train a network (here, called a loss network) separately, beforehand, which will be used as a speech recognizer in the style transfer approach. Its role is to separate speech spectrograms into multiple layers using a convolutional network. It will be used for extracting content (related to the utterance) and style (related to the accent and pronunciation) from the images (spectrograms). The loss network is depicted on the diagram as *LN*. In order to properly extract style and content from the input spectrograms the loss network must be trained using data from such a domain. The data utilized in the training process is described in Section 4.6.1

As the loss network model for automatic speech recognition tasks, properties of convolutional and recurrent layers were combined, where the former layers become, in fact, employed as feature extractors. Convolutional neural networks have been proven

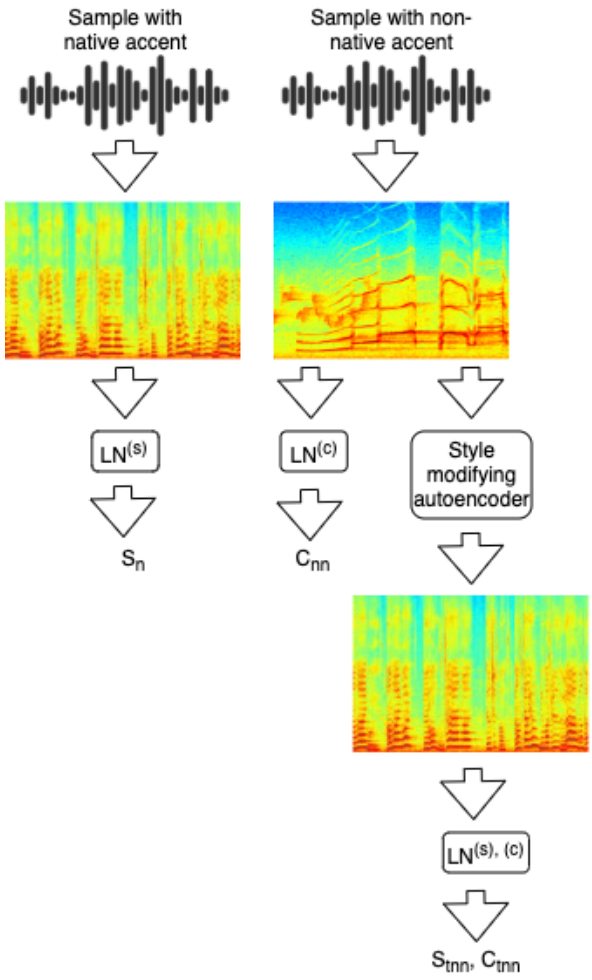

Figure 11: Audio style transfer schema during training process. The basic diagram of style transfer-based accent modification

to give outstanding results when applied to images, here spectrograms. They are able to detect and learn local features which are later passed on to recurrent layers. The architecture of the neural network is depicted in Table 9. It accepts an image as the input and outputs a sequence of letters.

Abbreviations used in Table 9:

$Conv1D$	– 1-dimensional convolutional layer,
$Maxpool$	– max pooling layer,
$Bidirectional$	– wrapper with RNN,
$BatchNormalization$	– batch normalizing layer,
$TimeDistributed$	– layer for every temporal slice of the input,
$Dropout$	– dropout layer,
$SoftmaxActivation$	– softmax activation function layer,
P	– pool size,
U	– number of hidden units in the RNN

The main step of this approach is training the autoencoder for style modification, which performs the essence of the idea. Its architecture is the same as the one of the autoencoder used in the previous approach for accent modification and is described in detail in Table 8.

During one training step the spectrogram of a sample with a native accent is fed into the loss network, which extracts style matrix Sn from certain layers. It is depicted in the diagram as $\mathbf{LN^{(s)}}$. Next, the spectrogram of a sample containing a non-native accent is pushed through the same loss network which results in extraction of content matrix Cnn. The process is depicted in the diagram as $\mathbf{LN^{(c)}}$. The sample is also fed into the style modifying autoencoder which outputs a modified spectrogram that is fed into the loss network to extract matrices representing style and content of the transformed sample ($Stnn$, $Ctnn$ respectively). It is symbolized in the figure as $\mathbf{LN^{(s),\ (c)}}$.

After having received Sn, Cnn, $Stnn$, $Ctnn$, the content and style losses are for-

Table 9: Detailed architecture of the CNN-RNN model used as the loss network in style transfer approach

Layer	Output shape	Parameters
InputLayer	B, T, 161	0
Conv1D (F=220, K=3)	B, T, 220	389840
Conv1D (F=220, K=3)	B, T, 220	389840
Maxpool (P=2)	B, T, 220	880
Conv1D (F=150, K=3)	B, T, 150	265800
Conv1D (F=150, K=3)	B, T, 150	265800
Maxpool (P=2)	B, T, 150	600
Conv1D (F=100, K=3)	B, T, 100	177200
Conv1D (F=100, K=3)	B, T, 100	177200
Maxpool (P=2)	B, T, 100	400
Conv1D (F=80, K=3)	B, T, 80	141760
Conv1D (F=80, K=3)	B, T, 80	141760
Maxpool (P=2)	B, T, 80	320
Conv1D (F=80, K=3)	B, T, 80	141760
Conv1D (F=80, K=3)	B, T, 80	141760
Maxpool (P=2)	B, T, 80	320
Conv1D (F=80, K=3)	B, T, 80	141760
Conv1D (F=80, K=3)	B, T, 80	141760
Bidirectional (U=200)	B, T, 400	505200
BatchNormalization	B, T, 400	1600
TimeDistributed	B, T, 29	11629
Dropout	B, T, 29	0
TimeDistributed	B, T, 29	870
SoftmaxActivation	B, T, 29	0
	Total params:	3,038,059
	Trainable params:	3,038,059
	Non-trainable params:	0

mulated. Content loss is calculated as:

$$L_c = \sum_{l} \sum_{i,j} (\alpha Cnn_{i,j}^{l} - \alpha Ctnn_{i,j}^{l})^2 \tag{10}$$

where l is the set of convolutional layers representing the content of the sound wave.

Style loss is calculated as:

$$L_s = \sum_{l} \sum_{i,j} (\beta Gn^l_{i,j} - \beta Gtnn^l_{i,j})^2 \tag{11}$$

where:

Gn^l – the Gram matrix of lth layer of Sn received from the Loss network

$Gtnn^l$ – the Gram matrix of the lth layer of $Stnn$

Gram matrix is the result of the multiplication of the matrix by its transpose.

Therefore the final loss function is represented as:

$$L = L_s + L_c \tag{12}$$

After having formulated our loss function the algorithm backpropagates the error to train the style modifying autoencoder network for the task of accent modification. At this step the weights of the loss network are already frozen and do not take part in the training process.

Such sequences are executed repeatedly with samples drawn from native speech datasets and non-native ones, respectively. It is worth mentioning that in cases of style transfer, as opposed to the autoencoder approach, it is not necessary for both spectrograms (with native and non-native accents) to represent the same content. As mentioned in the experimental part of this article, several runs of training the autoencoder were performed in order to find the best subsets of convolutions to represent the style and content layers.

During the inference phase only the trained autoencoder that modifies the accent of a new sample, is used.

Table 10: Detailed architecture of the CNN-RNN model used as the ASR module

Layer	Output shape	Parameters
InputLayer	B, T, 161	0
Conv1D (F=250, K=3)	B, T, 250	443000
Conv1D (F=250, K=3)	B, T, 250	443000
Maxpool (P=2)	B, T, 250	940
Conv1D (F=150, K=3)	B, T, 150	265800
Conv1D (F=150, K=3)	B, T, 150	265800
Maxpool (P=2)	B, T, 150	600
Conv1D (F=100, K=3)	B, T, 100	177200
Conv1D (F=100, K=3)	B, T, 100	177200
Maxpool (P=2)	B, T, 100	400
Conv1D (F=80, K=3)	B, T, 80	141760
Conv1D (F=80, K=3)	B, T, 80	141760
Bidirectional (U=200)	B, T, 400	505200
BatchNormalization	B, T, 400	1600
TimeDistributed	B, T, 29	11629
Dropout	B, T, 29	0
TimeDistributed	B, T, 29	870
SoftmaxActivation	B, T, 29	0
	Total params:	2,576,759
	Trainable params:	2,576,759
	Non-trainable params:	0

4.4 Speech recognition using spectrograms

At the end of our pipeline the speech recognition process is performed. One of the two approaches evaluated is using an ASR system trained on spectrograms. A model for the speech recognition using spectrograms converted from WAV files, was created. The architecture of the network playing the role of the ASR system is depicted in Table 10. I used a combination of convolutional and recurrent neural networks (CNN-RNN) in order to train a new speech recognition system. Similar to the loss network mentioned earlier, this network also accepts images, and outputs a sequence of letters.

The network was trained using a popular and publicly available dataset *LibriSpeech*.

The details, together with the metrics and the results of the training process, are shown in Section 4.6.

4.5 Speech recognition using sound sample-based ASR

4.5.1 Cloud-based ASR

Another way of speech recognition that I decided to check, is an online ASR service. In my research I decided upon Google Cloud Speech-to-Text and used the results of recognized text to calculate accuracy metrics.

4.5.2 TDNN architecture-based ASR

Another network-Time Delay Neural Network (TDNN, [50]) was used as an evaluation tool in our methodology.

For the acoustic model, which translates the acoustic signal into a phonetic representation, the time-delay neural network is frequently employed in speech recognition software. The frames of acoustic features are the network's inputs. The TDNN produces a probability distribution over each of the phones defined for the target language as an output. To put it another way, the purpose is to read the audio frame by frame and classify each frame into the most likely phone.

Each input frame is a column vector representing a single time step in the signal, with the rows representing the feature values, in one layer of the TDNN. The network employs a smaller matrix of weights (the kernel or filter), which glides over the signal and transforms it into an output using the convolution operation.

4.6 Experiments

4.6.1 Datasets used

One of the two datasets utilized within this research is *English Speech Database Read by Japanese Students (UME-ERJ)* explained in details in Appendix A.1

The dataset was employed for training the both the autoencoder in Section 4.2 and the style transfer network in Section 4.3, as it contains sentences and words pronounced by both native and non-native speakers. The training dataset contains around 18,662 pairs of spectrograms representing the exact same utterances from native and non-native speakers The remaining test and validation subsets did not overlap with the training subset.

Another dataset used in the research is the *LibriSpeech* dataset described in Appendix A.2. It was used to train both the spectrogram-based ASR module (after converting samples to spectrograms) used as the last part of our pipeline (Section 4.4) and the TDNN-based network (4.5.2). Another application of the dataset is training the loss network for the style transfer approach in one of the accent modification variants (Section 4.3).

The summary of utilized datasets is show in the Table. 11.

Table 11: Summary and splits of the utilized datasets

Trained network	Utilized dataset
Autoencoder	UMEERJ (18,662 samples subset)
Style transfer network	UMEERJ (split (0.8/0.1/0.1))
Loss network	LibriSpeech (train-clean-360)
CNN-RNN-based ASR	LibriSpeech (train-clean-360)
TDNN-based ASR	LibriSpeech (train-clean-360)

4.6.2 Experiments and metrics

The autoencoder introduced in 4.2 as well as the loss network (4.3) and the spectrogram-based ASR system (4.4) were trained using the Connectionist temporal classification (CTC, [51]) function.

In our research separate experiments for several processes in our pipeline were designed. Namely, the experiments were performed and the results evaluated for:

1. relative improvement in the speech recognition accuracy in case of autoencoder-based accent modification, including both approaches for ASR in the final stage

2. relative improvement in the speech recognition accuracy in cases of audio style transfer-based accent modification, including both approaches for ASR. In this approach several runs of training the style modifying autoencoder were performed to check the best combination of subsets of style and content layers in the loss network.

4.6.3 Metrics

Two different evaluation processes were employed depending on the experiment type.

As a quality metric for the speech recognition processes (loss network, ASR module and the cloud-based service), three different metric types were chosen. First is the standard Word Error Rate (WER) and the second one is Character Error Rate (CER), which is expressed as:

$$CER = \frac{i + s + d}{n} \tag{13}$$

where:

i – number of insertions,

s – number of substitutions,

d – number of deletions,

n – total number of characters

Another metric type introduced is phoneme similarity ([52]). It is expressed as Mean Similarity Score (MSS) in the results section of our work.

As for the evaluation of the accent modification itself, I decided to present a relative decrease in CER yielded by the ASR module from the last part of our pipeline (Google Cloud Speech-to-Text and the spectrogram-based ASR trained using *LibriSpeech*).

4.6.4 Results

Each respective result below represents an average over ten runs of each experiment with a particular setup.

The results without accent modification The ASR module (Section 4.4) was trained using spectrograms converted from *LibriSpeech train-clean-360* subset. It was evaluated using the *test-clean* dataset and achieved **15.7%** CER and **19.7%** WER. This model, evaluated with a 10% test subset of the spectrograms of *UME-ERJ* dataset, achieved only **46.3%** CER and **56.8%** WER.

The averaged result of speech recognition using spectrograms yielded by our Loss network is **11.2%** CER and **14.9%** WER using the *LibriSpeech test-clean* dataset.

The 10% test subset of the WAV samples of the *UME-ERJ* dataset was also used to evaluate the performance of the Google Cloud Speech-to-Text API and the result obtained was **39.8%** of CER.

The *LibriSpeech test-clean* dataset was also used for evaluating the TDNN-based

ASR network trained, and the result achieved in our test was **10%** CER and **12.5%** WER.

Impact of the autoencoder-based accent modification After activating the autoencoder-based accent modification in our pipeline, the same test subset of the *UME-ERJ* dataset gave a result of **36.1%** CER (evaluation by the spectrogram-based ASR model trained only on the *LibriSpeech* training set). Therefore, it yielded a **22%** $\left(\frac{46.3\% - 36.1\%}{46.3\%}\right)$ relative improvement in terms of CER.

In the case of the Google Cloud API the data was first fed to the autoencoder and then converted the modified spectrograms back to the sound wave format. At the end it was sent to the cloud service and recorded the recognized text. The 10% subset of samples from *UME-ERJ* were used and after the process obtained a result of **27.3%** CER, which translates to **31.4%** of the relative improvement.

Impact of the accent modification based on the style transfer approach For each combination of subsets tested for style and content layers, I checked the CER value on the 10% subset of spectrograms from the *UME-ERJ* dataset by feeding it into the trained autoencoder that modifies the style (Section 4.3) and feeding the respective result into the ASR (Section 4.4). The best setup gave a result of **31.7%** CER. Therefore, it yielded a **32%** relative improvement in terms of CER.

In the case of the cloud service evaluation the analogical process was followed as in Section 4.6.4. The best combination of style and content layers achieved the result of **23.9%** CER which means a relative improvement of **40%**.

All experimental results with the best CER are presented in Tables 12 and 13. The results for experiments conducted on the style transfer approach with different style

and content layers are presented in Table 15 and 16. The tables from 12 to 16 represent the results obtained when testing the setup with both 10% subset of the *UME-ERJ* dataset and specifically prepared 100-sentences test subset of *UME-ERJ*.

In the tables the following symbols were used:

WM – results of the experiment without the accent modification process,

A – the experiment with the autoencoder-based modification process,

ST – the experiment with the style transfer-based modification process,

RI_{CER} – relative improvement for the CER metrics

Table 12: Results of the style modification process in cases of evaluation done using trained CNN-RNN ASR model

	WM	A	ST
CER	46.3%	36.1%	31.7%
RI_{CER}	-	22%	32%
WER	56.8%	43.2%	34.9%
MSS	-0.94	1.43	1.97
CER (100-sentences)	42.1%	33.2%	28.3%
RI_{CER} (100-sentences)	-	21%	24.7%
WER (100-sentences)	51.8%	39.1%	32.3%
MSS (100-sentences)	-0.85	1.56	2.09

Table 13: Results of the style modification process in cases of evaluation done using Google cloud service

	WM	A	ST
CER	39.8%	27.3%	23.9%
RI_{CER}	-	31.4%	40%
CER (100-sentences)	34.3%	23.7%	23.5%
RI_{CER} (100-sentences)	-	30.9%	31.5%

Table 14: Results of the style modification process in cases of evaluation done using TDNN-based ASR network

	WN	A	ST
CER	38.1%	27.1%	26.1%
RI_{CER}	-	28.8%	31.4%
CER (100-sentences)	34.1%	23.1%	22.2%
RI_{CER} (100-sentences)	-	32.3%	34.8%

Table 15: Relative improvement depending on the content and style layers in cases of ASR model-based evaluation using 10% *UME-ERJ* subset

Style layers	Content layers	RI(CER)
1-10	6-12	32%
1-8	8-12	29.6%
1-10	10-12	30.1%
1-5	5-12	26.7%
1-4	4-12	15.6%

Table 16: Relative improvement depending on the content and style layers in cases of Google cloud-based evaluation using 10% *UME-ERJ* subset

Style layers	Content layers	RI(CER)
1-10	6-12	40%
1-8	8-12	36.1%
1-10	10-12	38.3%
1-5	5-12	29.4%
1-4	4-12	21.1%

4.7 Discussion

This article contains study of the audio style transfer methods used for improving accuracy of ASR which had been trained using native speech datasets and is used by non-native speakers.

I found that the style transfer methodology adapted to the speech domain yields better results than an autoencoder trained in a supervised way. I think that the reason behind it lies in the fact that training step was performed repeatedly by sampling ran-

dom samples to include, respectively, non-native and native accents, and transforming them into the spectrograms. The samples in each such pair do not have to represent the same speech content. This observation may suggest another interesting idea that if this approach were to be extended it might be possible to create a more universal autoencoder that might convert the accent of non-native speakers from multiple nationalities into one (e.g. North American English) accent. This is going to be one of the steps for the future of our research.

Another observation is an extension of the fact that the accent modification process is not conditioned on any variables related to the speaker or speech environment. That causes the situation where during the experiment testing phase the gender of the speaker in the sample after style transformation is different than in the one from before, while preserving the actual content of the pronounced word or sentence. However I did not treat such cases as failed, as our primary goal was to increase the accuracy of the ASR system, which was eventually achieved. Nevertheless, it will be a next step for our team to address.

As another future step in our research more experiments will be designed, i.e. the evaluation of the style transfer approach described in Section 4.3 using more datasets. I am also planning to evaluate the process of a style transfer-based approach and autoencoder-based approach with longer sentences and samples.

I am planning to develop our idea for non-native speech recognition further and to constantly improve the quality of the designed methodology. Furthermore, additional experiments will be conducted, i.e. using multiple nationalities of non-native English speakers, as well as using different datasets including samples of languages other than English.

4.8 Summary

In this research the problem of non-native speech recognition and the reason why training ASR systems adapted for such speech may be problematic were explained.

The idea behind style transfer methodology and our adaptation to the speech and sound domain were described in detail. The method was presented as a way to transform non-native pronunciation so that it resembles native speech to a higher extent, thus enabling the ASR system to perform better when being used by a non-native speaker.

Initial experiments were performed using the *UME-ERJ* dataset and several different pipelines for pronunciation modification tested. I evaluated each approach on a custom ASR trained to recognize speech from spectrograms, as well as on the publicly available Google Cloud Speech-to-Text. Our initial findings show that it is possible to augment the non-native speech samples in a way that they will be recognized with a higher accuracy by an ASR system.

I also pointed out several issues that appeared while training and evaluating our algorithms. This proves that there is definitely a lot of room for improvement in order to adapt the method to multiple speaker-dependent conditions and other non-native nationalities.

Chapter 5 Conclusion

Nowadays, more often than ever the humankind communicates online using a wide variety of conferencing tools and voice chats.

At the same time, with the current pace of globalization, the communication between people of different nationalities, languages and cultures represents the vast majority of such online-based interactions. Usually in the majority of such cases English is being used as a common language. Naturally the above mentioned facts mean that in such conversations at least one, or very often even all of the participants have to use English as a common language of choice during the conference, meaning that almost all of them represent non-native speakers of English.

Simultaneously it is rather obvious that the knowledge level of English language may differ greatly depending on the speaker. This in turn may lead to potential issues in communication between such participants, i.e. problems related to listening and understanding one another as well as speaking and pronouncing words correctly or introducing the correct accent and intonation.

The problem of communication between such speakers represents the topic of my research conducted within the scope of this dissertation. I presented the speech recognition technology as a potential facilitator in communications between non-native speakers of a particular language. I explained the reasons for the standard ASR tools performing rather poorly in cases where the end user is a non-native speaker of the language that had been used to train the ASR model. Namely, in Section 1.1 I explained in more detailed way the relation between speaker's nationality, linguistic background and such, and his knowledge and abilities related to second language performance.

To overcome the mentioned problems I researched and designed 3 algorithms and techniques which allow for a higher accuracy and lower error rates of ASR techniques when used by a non-native speaker. The first method is described in Section 3. It is a methodology for creating the high performing ASR models from the ground up, adapted for non-native speech. It is based on the idea of dual supervised learning, a setup that allows for the possibility of using unlabelled non-native speech datasets that usually come in much larger volumes than labelled data. It is based on a designed setup consisting of an STT and TTS models, as well as language and speech models connected in a loop, where models learn based on the feedback received from one another. The very important point to underline in case of this algorithm is the fact that it is an actor-critic based approach with a feedback loop consisting of 2 models playing the role of the critic and two models set to learn from the respective feedback of the critics models.

ASR models used in cloud-based solutions are usually trained with datasets composed of native speech samples. Therefore the samples representing a non-native speech are usually not converted to text with a sufficient accuracy. In order to tackle this problem I designed and implemented a methodology for real-time conversion of speech samples. The idea is described in details in Section 4. The algorithm is based on the notion of neural style transfer applied to the domain of speech and sound. The purpose of this idea is to modify the accent, pronunciation and other features of the non-native speech sample, so that it resembles a native speech sample to a higher extent. The algorithm modifies only the accent other features related to the speech characteristics. The content of the speech, namely the pronounced content is not modified. The result of such process is a speech sample with the same content, but with accent and pronunciation modified so that the cloud-based ASR models can achieve higher speech

recognition accuracy. The important difference and therefore the contribution is the fact that the data in form of the spectrograms is depicted as an image with time dimension as the X axis and the frequency dimension at the Y axis. While a traditional approaches to analyze static images with convolutional neural networks utilize the fact of the translational invariance of the process, the same cannot be said for the case of analaysing the spectrograms with networks based on the convolutional feature extractors. Moreover in the case of the static images the feature extractor can share its filters' weights across the whole image, effectively being able to correctly infer or classify the picture irrespective of the object's location in the picture. For the case of spectrograms however, a slight shift in the direction of frequencies might result in very different effect on the final outcome. The convolutional neural networks employed in our case are utilizing the weights shared only within one specific frequency band thus having multiple sets of such weights.

The third idea described in details in Section 3.4 represents a correction on the text level between sentences produced by a non-native speaker and the same sentences uttered by a native speaker of a particular language. The purpose of this approach is to correct the results of the ASR model applied to non-native speech in order to ensure that the final result resembles a sentence by a native speaker. The algorithm provides a encoder-decoder approach for this purpose. It is applied right after the ASR process, to its textual results, in order to provide a supplementary text-based correction of the non-native speech converted to text. This algorithm can be applied right after any of the two previous speech domain algorithms.

I evaluated each of the designed methodologies experimentally, and proved that it is possible to significantly increase the accuracy of non-native speech recognition.

While the presented solutions significantly contribute to solving the aforementioned

problem, I believe there is still a lot of room for improvement in terms on the ASR for non-native speech. Namely, the algorithms designed in the scope of my research represent a set of methodologies developed for the purpose of increasing the accuracy of speech recognition process targeted for non-native speakers. The goal of the research is to reduce the errors of the ASR. The methods do not take into account other paralinguistic features related to the speech and the speaker. For example the algorithm designed for performing the accent modification focuses only on modifying the speech samples so that the overall recognition result increases. It does not preserve features like speakers' gender or age. During the experimental evaluation of this methodology we came across cases where the algorithm modified the speech sample to that extent, that the result contained a speech as if it was pronounced by a speaker of opposite gender. While this is not considered a mistake in the research approach, because the final goal is to increase the ASR accuracy, naturally it is clear that our methodology cannot be used yet for modifying the speech of a non-native speaker with the purpose of re-playing it to a listener.

Another issue which will be addresses in the future is the processing time of inference using our algorithms. Currently the latency introduced by performing the accent neutralization process represents a main bottleneck preventing the algorithm to be used in the on-the-fly fashion as a live speech converter for interactions between speakers of different nationalities and linguistic backgrounds.

These issues that still remain to be solved compose a list of points for future work on this subject.

References

[1] O. Abdel-Hamid, A. R. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, vol. 22, no. 10, pp. 1533–1545, Oct. 2014. [Online]. Available: http://dx.doi.org/10.1109/TASLP.2014.2339736

[2] N. Dave, "Feature extraction methods lpc, plp and mfcc in speech recognition," *International Journal for Advance Research in Engineering and Technology*, vol. 1, pp. 1–5, 7 2013.

[3] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. Audio, Speech and Lang. Proc.*, vol. 19, no. 4, pp. 788–798, May 2011.

[4] M. Li, K. Han, and S. S. Narayanan, "Automatic speaker age and gender recognition using acoustic and prosodic level information fusion," *Computer Speech Language*, 2013.

[5] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," *Proceedings ICASSP. IEEE*, 2013.

[6] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. Engel, L. Fan, C. Fougner, T. Han, A. Hannun, B. Jun, P. LeGresley, L. Lin, S. Narang, A. Ng, S. Ozair, R. Prenger, J. Raiman, S. Satheesh, D. Seetapun, S. Sengupta, Y. Wang, Z. Wang, C. Wang, B. Xiao, D. Yogatama, J. Zhan, and Z. Zhu,

"Deep speech 2: End-to-end speech recognition in english and mandarin," 2015. [Online]. Available: arXiv:1512.02595

[7] W. Xiong, L. Wu, J. Droppo, X. Huang, and A. Stolcke, "The microsoft 2017 conversational speech recognition system," in *Proceedings IEEE ICASSP*, 2018, pp. 5934–5938.

[8] T. Drugman and T. Dutoit, "Glottal closure and opening instant detection from speech signals," *Proceedings Interspeech*, 2009.

[9] K. Livescu and J. Glass, "Lexical modeling of non-native speech for automatic speech recognition," *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2000.

[10] T. Tan and L. Besacier, "Acoustic model interpolation for non-native speech recognition," *Proceedings ICASSP*, 2007.

[11] L. M. Tomokiyo, "Recognizing non-native speech: Characterizing and adapting to non-native usage in lvcsr," Ph.D. dissertation, Carnegie Mellon University, 2001.

[12] K. Radzikowski, L. Wang, O. Yoshie, and R. Nowak, "Dual supervised learning for non-native speech recognition," *EURASIP Journal on Audio, Speech and Music Processing*, vol. 2019:3, pp. 1–10, 2019, doi:10.1186/s13636-018-0146-4, https://rdcu.be/bgUxy.

[13] K. Radzikowski, L. Wang, O. Yoshie, and R. M. Nowak, "Accent modification for speech recognition of non-native speakers using neural style transfer," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2021, pp. 1–10, 2021, doi:10.1186/s13636-021-00199-3.

[14] J. Ficek, K. Radzikowski, J. K. Nowak, O. Yoshie, J. Walkowiak, and R. Nowak, "Analysis of gastrointestinal acoustic activity using deep neural networks," *Sensors*, vol. 21, no. 22, 2021.

[15] J. K. Nowak, R. Nowak, K. Radzikowski, I. Grulkowski, and J. Walkowiak, "Automated bowel sound analysis: An overview," *Sensors*, vol. 21, no. 16, 2021.

[16] K. Radzikowski, L. Wang, and O. Yoshie, "Non-native english speaker's speech correction, based on domain focused document," in *Proceedings of the Conference of Institute of Electrical Engineers of Japan, Electronics and Information Systems Division*, 2016.

[17] K. Radzikowski, W. Le, and O. Yoshie, "Non-native english speakers' speech correction, based on domain focused document," in *Proceedings of the 18th International Conference on Information Integration and Web-based Applications and Services*, ser. iiWAS. New York, NY, USA: ACM, 2016, pp. 276–281.

[18] K. Radzikowski, L. Wang, and O. Yoshie, "Non-native speech recognition using characteristic speech features, with respect to nationality," *Proceedings of the conference of institute of electrical engineers of japan, electronics and information systems division*, 2017.

[19] S. Furui, "50 years of progress in speech and speaker recognition research," in *The Journal of the Acoustical Society of America*, 2005.

[20] T. Gulzar, A. Singh, D. Kumar, and N. Farooq, "A systematic analysis of automatic speech recognition: An overview," *International Journal of Current Engineering and Technology*, vol. 4, 06 2014.

[21] S. Furui, "Cepstral analysis technique for automatic speaker verification," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 29, pp. 254 – 272, 05 1981.

[22] M. Sambur and L. Rabiner, "A statical decision approach to the recognition of connected digits," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 24, no. 6, pp. 550–558, December 1976.

[23] L. R. Rabiner and J. G. Wilpon, "A simplified, robust training procedure for speaker trained, isolated word recognition systems," *The Journal of the Acoustical Society of America*, vol. 68, no. 5, pp. 1271–1276, 1980. [Online]. Available: https://doi.org/10.1121/1.385120

[24] K. F. Lee and H. W. Hon, "Large-vocabulary speaker-independent continuous speech recognition using hmm," in *ICASSP-88., International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, April 1988, pp. 123–126.

[25] H. Suzuki, H. Zen, Y. Nankaku, C. Miyajima, K. Tokuda, and T. Kitamura, "Speech recognition using voice-characteristic-dependent acoustic models," in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, vol. 1, April 2003, pp. I–I.

[26] A. Revathi and Y. Venkataramani, "Speaker independent continuous speech and isolated digit recognition using vq and hmm," in *2011 International Conference on Communications and Signal Processing*, 2011, pp. 198–202.

[27] M. Dua, R. Aggarwal, V. Kadyan, and S. Dua, "Punjabi automatic speech recognition using htk," *International Journal of Computer Science Issues*, vol. 9, 07 2012.

[28] A. Kuamr, M. Dua, and T. Choudhary, "Continuous hindi speech recognition using gaussian mixture hmm," in *2014 IEEE Students' Conference on Electrical, Electronics and Computer Science*, March 2014, pp. 1–5.

[29] D. Baby, J. F. Gemmeke, T. Virtanen, and H. Van hamme, "Exemplar-based speech enhancement for deep neural network based automatic speech recognition," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 4485–4489.

[30] Q. B. Nguyen, T. T. Vu, and C. M. Luong, "Improving acoustic model for english asr system using deep neural network," in *The 2015 IEEE RIVF International Conference on Computing Communication Technologies - Research, Innovation, and Vision for Future (RIVF)*, Jan. 2015, pp. 25–29.

[31] B. J. Mohan and N. R. Babu, "Speech recognition using mfcc and dtw," *2014 International Conference on Advances in Electrical Engineering (ICAEE)*, pp. 1–4, 2014.

[32] K. Ravinder, "Comparison of hmm and dtw for isolated word recognition system of punjabi language," in *Proceedings of the 15th Iberoamerican Congress Conference on Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, ser. CIARP'10. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 244–252. [Online]. Available: http://dl.acm.org/citation.cfm?id=1948207.1948251

[33] Y. Xia, T. Qin, W. Chen, J. Bian, N. Yu, and T.-Y. Liu, "Dual supervised learning," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70, International Convention Centre, Sydney, Australia, 06–11 Aug 2017, pp. 3789–3798.

[34] F. Bimbot, R. Pieraccini, E. Levin, and B. Atal, "Variable-length sequence modeling: Multigrams," *Signal Processing Letters, IEEE*, vol. 2, pp. 111 – 113, 07 1995.

[35] S. Deligne and F. Bimbot, "Language modeling by variable length sequences: theoretical formulation and evaluation of multigrams," in *1995 International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, 1995, pp. 169–172.

[36] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[37] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010)*, vol. 2010, no. 9, 2010, pp. 1045–1048.

[38] I. Sutskever, J. Martens, and G. Hinton, "Generating text with recurrent neural networks," in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ser. ICML'11. USA: Omnipress, 2011, pp. 1017–1024. [Online]. Available: http://dl.acm.org/citation.cfm?id=3104482.3104610

[39] A. Graves, "Generating sequences with recurrent neural networks," 2013. [Online]. Available: arXiv:1308.0850

[40] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," 2016. [Online]. Available: arXiv:1609.03499

[41] X. Liu, "Deep convolutional and lstm neural networks for acoustic modelling in automatic speech recognition," http://cs231n.stanford.edu/reports/2017/pdfs/804.pdf.

[42] W. Song, "End-to-end deep neural network for automatic speech recognition," https://cs224d.stanford.edu/reports/SongWilliam.pdf, 2015.

[43] Y. Miao, M. A. Gowayyed, and F. Metze, "Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding," *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 167–174, 2015.

[44] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ser. ICML'14. JMLR.org, 2014, pp. II–1764–II–1772. [Online]. Available: http://dl.acm.org/citation.cfm?id=3044805.3045089

[45] X. Tian, J. Zhang, Z. Ma, Y. He, J. Wei, P. Wu, W. Situ, S. Li, and Y. Zhang, "Deep lstm for large vocabulary continuous speech recognition," 2017. [Online]. Available: arXiv:1703.07090

[46] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," 2015. [Online]. Available: arXiv:1508.06576

[47] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," 2016. [Online]. Available: arXiv:1603.08155

[48] E. Grinstein, N. Q. K. Duong, A. Ozerov, and P. Pérez, "Audio style transfer," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 586–590.

[49] P. Charles, "keras," 2019. [Online]. Available: https://github.com/charlespwd/project-title

[50] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *INTERSPEECH*, 2015.

[51] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 369–376. [Online]. Available: https://doi.org/10.1145/1143844.1143891

[52] B. Hixon, E. Schneider, and S. L. Epstein, "Phonemic similarity metrics to compare pronunciation methods," in *INTERSPEECH*, 01 2011, pp. 825–828.

[53] K. Kita, T. Kawabaa, and T. Hanazawa, "Hmm continuous speech recognition using stochastic language models," in *International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, 05 1990, pp. 581 – 584.

[54] S. Lee, Y. Lee, and N. Cho, "Multi-stage speech enhancement for automatic speech recognition," in *2016 IEEE International Conference on Consumer Electronics (ICCE)*, Jan 2016, pp. 383–384.

[55] P. Verma and J. O. Smith, "Neural style transfer for audio spectograms," 2018. [Online]. Available: arXiv:1801.01589

[56] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015. [Online]. Available: arXiv:1409.1556

[57] A. Metallinou and J. Cheng, "Using deep neural networks to improve proficiency assessment for children english language learners," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[58] K. Tokuda and H. Zen, "Directly modeling speech waveforms by neural networks for statistical parametric speech synthesis," *Proceedings ICASSP*, 2015.

[59] G. Shi, M. Shanechi, and P. Aarabi, "On the importance of phase in human speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions*, 2006.

[60] T. T. Ping, "Automatic speech recognition for non-native speakers," Ph.D. dissertation, Université Joseph-Fourier - Grenoble, 2008.

[61] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[62] J. T. Geiger, Z. Zhang, F. Weninger, B. Schuller, and G. Rigoll, "Robust speech recognition using long short-term memory recurrent neural networks for hybrid acoustic modelling," *Proceedings Interspeech*, 2014.

[63] "Ume english speech database read by japanese students (ume-erj)," 2007. [Online]. Available: http://research.nii.ac.jp/src/en/UME-ERJ.html

[64] "Librispeech: An asr corpus based on public domain audio books," 2015.

[65] "Ets corpus of non-native written english ldc2014t06," 2014. [Online]. Available: https://catalog.ldc.upenn.edu/LDC2014T06

[66] "Nict japanese learner english (jle) corpus," 2004. [Online]. Available: https://alaginrc.nict.go.jp/nict_jle/index_E.html

[67] "Corpus of contemporary american english." [Online]. Available: https://www.english-corpora.org/coca/

Appendix A Public datasets

My research requires high quality datasets. I prepared several such datasets using data publicly available and converted to form required for the purpose of training the models. I decided to depict these datasets in separate appendix providing their symbols, sizes and URLs.

A.1 UME-ERJ

UME-ERJ is a set of around 75,000 samples called *English Speech Database Read by Japanese Students (UME-ERJ)* [63] containing Japanese, as well as Americans, pronouncing English sentences.

1. Sentences for learning phonemic pronunciation:
    - 460 phonetically-balanced sentences,
    - 32 sentences including phoneme sequences difficult for Japanese to pronounce correctly,
    - 100 sentences designed for test set,
    - 302 minimal-pair words,
    - 300 phonemically balanced words.
2. Sentences for learning prosody of speech:
    - 94 sentences with various intonation patterns,
    - 120 sentences with various accent and rhythm patterns,
    - 109 words with various accent patterns.

This dataset is available at *http://research.nii.ac.jp/src/en/UME-ERJ.html*. The dataset was employed for training the both the autoencoder in Section 4.2 and the style transfer network in Section 4.3, as it contains sentences and words pronounced by both native and non-native speakers. The training dataset contains around 18,662 pairs of spectrograms representing the exact same utterances from native and non-native speakers. This amount of the recordings represents around 26h of speech. The remaining test and validation subsets did not overlap with the training subset.

A.2 Librispeech

Another dataset used in the research is the *Libri Speech* dataset ([64]). *Libri Speech* is a corpus of approximately 1000 hours of 16kHz read English speech, prepared by Vassil Panayotov with the assistance of Daniel Povey. The data is derived from read audiobooks from the LibriVox project, and has been carefully segmented and aligned.

It was used to train both the spectrogram-based ASR module (after converting samples to spectrograms) used as the last part of our pipeline (Section 4.4) and the TDNN-based network (4.5.2). Another application of the dataset is training the loss network for the style transfer approach in one of the accent modification variants (Section 4.3). Also, we used the dataset to train the TDNN-based ASR system, as another network evaluating the performance of our pipeline.

A.3 NICT Japanese Learner English (JLE) Corpus

The Japanese National Institute of Information and Communications Technology (NICT) created the *NICT Japanese Learner English (JLE) Corpus* ([66]), which contains 2 million words contributed by 1200 Japanese students. These students were interviewed

for the Standard Speaking Test (SST), a Japanese English language competency test rated on a scale of 1 to 9. The 15-minute interview consists of casual conversation, visual descriptions, role-playing, and tale telling. These interviews were transcribed, and grammatical mistakes in 45 categories were personally marked and fixed for 167 of them.

For example,

I lived in <at crr="">the</at> New Jersey

where *at* is the article error label, "the" is the word that should be deleted, and *crr* is the word that should be inserted (in this instance, the null string).

Dataset preparation On the interviewee turns in these transcripts, sentence segmentation was done, and sentences with just one word were discarded. This method produced 15637 sentences with a total word count of nearly 153000. Articles, noun number, and prepositions are the three most common mistake groups, followed by a variety of verb-related errors. Before we began our experiments, it was necessary to parse the entire dataset and prepare it in the form of sentences by the non-native speaker , aligned with the same sentence corrected by the interviewer (native speaker). In this way, a fully labelled dataset of 15637 samples was created.

In Table 17 we demonstrated one example of such labelled sample:

Table 17: Two examples from parsed JLE dataset

	Text by a non-native speaker	Corrected text
1	And my sister is twenty years old. She's **free arbeiter**.	And my sister is twenty years old. She's **a part-time worker**.
2	My husband is **so-called salary man**. He **get** up early in the morning and **leave the** home early around six o'clock and comes home late at night.	My husband is **a so-called business man**. He **gets** up early in the morning and **leaves** home early around six o'clock and comes home late at night.

The dataset was used for creating the text-level conversion method described in Section 3.4.

A.4 Corpus of Contemporary American English (COCA)

The Corpus of Contemporary American English ([67]) is the first major, genre-balanced corpus of any language that has been built from the ground up as a'monitor corpus,' allowing researchers to track and examine current changes in the language. The corpus of 400 million words is evenly split between spoken word, fiction, popular magazines, newspapers, and scholarly publications. Most significantly, the genre balance remains nearly same year after year, allowing it to properly represent developments in the 'real world.' The corpus is almost evenly divided between spoken, fiction, popular magazines, newspapers, and academic journals—20 percent in each genre (see Davies 2009b for a more complete overview of the textual corpus, as well as Davies (2005) and Davies (2009a) for information on earlier versions of the corpus architecture). There are about 160,000 texts in the corpus as of August 2009, and they originate from a range of sources:

- Spoken: Transcripts of unscripted dialogue from over 150 various TV and radio shows (examples: All Things Considered (NPR), Newshour (PBS), Good Morning America (ABC), Today Show (NBC), 60 Minutes (CBS), Hannity and Colmes (Fox), Jerry Springer, Oprah, and others).

- Fiction: (79 million words) Short stories and plays from literary journals, children's magazines, popular magazines, first chapters of first edition novels from 1990 to the present, and movie scripts are all available.

- Popular magazines: Nearly 100 different magazines, with a decent mix of specific

areas (overall and by year) (news, health, home and gardening, women, financial, religion, sports, and so on). Time, Men's Health, Good Housekeeping, Cosmopolitan, Fortune, Christian Century, Sports Illustrated, and others are only a few instances.

- Newspapers: USA Today, New York Times, Atlanta Journal Constitution, San Francisco Chronicle, and others are among the ten newspapers from around the United States. The newspaper also has a solid balance of diverse sections, such as local news, opinion, sports, finance, and so on.

- Academic journals: There are about 100 peer-reviewed journals. These were chosen to span the whole range of the Library of Congress classification system, both overall and by number of words per year (e.g., a particular percentage from B (philosophy, psychology, religion), D (global history), K (education), T (technology), and so on).

This dataset was used to train the language model described in Section 3.2.

Appendix B List of my publications

- (J1) J. Ficek, **K. Radzikowski**, J. K. Nowak, O. Yoshie, J. Walkowiak, and R. Nowak, "Analysis of gastrointestinal acoustic activity using deep neural networks," *Sensors*, vol. 21, no. 22, 2021, doi:10.3390/s21227602.

- (J2) J. Nowak, R. Nowak, **K. Radzikowski**, I. Grulkowski and J. Walkowiak, "Automated Bowel Sound Analysis: An Overview," *Sensors*, vol. 21, 2021, doi:10.3390/s21165294.

- (J3) **K. Radzikowski**, L. Wang, O. Yoshie, and R. M. Nowak, "Accent modification for speech recognition of non-native speakers using neural style transfer," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2021, pp. 1–10, 2021, doi:10.1186/s13636-021-00199-3.

- (J4) **K. Radzikowski**, L. Wang, O. Yoshie, and R. Nowak, "Dual supervised learning for non-native speech recognition," *EURASIP Journal on Audio, Speech and Music Processing*, vol. 2019:3, pp. 1–10, 2019, doi:10.1186/s13636-018-0146-4, https://rdcu.be/bgUxy

- (C1) **K. Radzikowski**, O. Yoshie and R. Nowak, "Support software for Automatic Speech Recognition systems targeted for non-native speech," *Proceedings of the 22nd International Conference on Information Integration and Web-based Applications & Services (iiWAS '20),* November 30-December 2, 2020, Chiang Mai, Thailand. ISBN 978-1-4503-8922-8/20/11, ACM International Conference Proceedings Series, ACM, December 2020.

- (C2) **K. Radzikowski**, M. Forc, L. Wang, O. Yoshie and R. Nowak, "Accent

neutralization for speech recognition of non-native speakers," *Proceedings of the 21st International Conference on Information Integration and Web-based Applications & Services (iiWAS '19),* Munich, Germany, December 2019, ISBN 978-1-4503-7179-7, ACM International Conference Proceedings Series, ACM, December 2019.

- (C3) **K. Radzikowski**, M. Forc, L. Wang, O. Yoshie and R. Nowak, "Non native speech recognition using audio style transfer," *Proceedings of SPIE,* vol. 11176, pp. 1–6, 2019, doi:10.1117/12.2536535.

- (C4) **K. Radzikowski**, K. Checiński, M. Forc, L. Lepak, M. Jablonski, W. Kusmirek, B. Twardowski, P. Wawrzynski and R. Nowak, "Widget detection on screenshots using computer vision and machine learning algorithms," *Proceedings SPIE,* vol. 11176, pp. 1–8, 2019, doi:10.1117/12.2536406

- (C5) **K. Radzikowski**, R. Nowak and O. Yoshie, "Neural style transfer for non-native speech recognition," *Proceedings of PP-RAI 2019,* Polskie Porozumienie na rzecz Rozwoju Sztucznej Inteligencji, ISBN: 978-83-943803-2-8

- (C6) **K. Radzikowski**, "Audio style transfer for non-native speech recognition," *Proceedings of Conference on Photonics Applications in Astronomy, Communications, Industry, and High-Energy Physics Experiments* 2018, Wilga, Poland

- (C7) **K. Radzikowski**, L. Wang and O. Yoshie, "Non-native speech recognition using characteristic speech features, with respect to nationality," *Proceedings of the conference of institute of electrical engineers of japan, electronics and information systems division* 2017, CD-ROM