Distribution Agreement

In presenting this thesis as a partial fulfillment of the requirements for a degree from Emory University, I hereby grant to Emory University and its agents the non-exclusive license to archive, make accessible, and display my thesis in whole or in part in all forms of media, now or hereafter now, including display on the World Wide Web. I understand that I may select some access restrictions as part of the online submission of this thesis. I retain all ownership rights to the copyright of the thesis. I also retain the right to use in future works (such as articles or books) all or part of this thesis.

Ran Xu April 13th, 2021

Analysis of Deep Learning-based Speech and Text Models for Early Detection of Alzheimer's Disease

By

Ran Xu

Jinho D. Choi
Advisor

Department of Computer Science

Jinho D. Choi
Advisor

Davide Fossati
Committee Member

Yuanzhe Xi
Committee Member

2021

Analysis of Deep Learning-based Speech and Text Models for Early Detection of Alzheimer's Disease

by

Ran Xu

Jinho D. Choi
Advisor

An abstract of
a thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors

Department of Computer Science

2021

Abstract

By Ran Xu

This paper presents a new dataset, B-SHARP, which can be used to detect Mild Cognitive Impairment (MCI), an early stage of Alzheimer's Disease. The dataset contains 721 speech recordings from 144 MCI patients and 185 health controls, on three topics about daily activity, room environment and picture description. Given the B-SHARP dataset, several hierarchical transformer models on the text side based on the transcription and multiple speech models with different encoding methods based on acoustic information are developed. And finally, the model performance are evaluated and a comparison is drawn between text models and speech models.

Analysis of Deep Learning-based Speech and Text Models for Early Detection of Alzheimer's Disease

by

Ran Xu

Jinho D. Choi
Advisor

A thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Science with Honors

Department of Computer Science

2021

Acknowledgements

First and foremost, I would like to thank my advisor, Professor Jinho D. Choi, who introduced me into the world of Natural Language of Processing. I started to work with Dr. Choi on my junior year and I gradually developed my interests in NLP with the influence of his great dedication. He offered a lot guidance and support towards both my project and my background knowledge study.

In addition, I also want to thank my committee member Professor Davide Fossati and Professor Yuanzhe Xi. They spent their spare time to read my paper and attend the defense, and their valuable advice gave me much help to improve my thesis.

Last but not least, I want to thank my parents who emotionally and financially support me throughout my college years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Alzheimer's disease (AD) is a progressive disease which destroys memory and other important mental functions including semantic and pragmatic levels of language processing [12]. So far, there are around 5.8 million Americans living with AD in 2020. By 2050, this number is expected to increase to approximately 14 million[1]. Therefore, it is critial to detect AD in the early stage as it will potentially rescue enormous patients from suffering AD. To this end, many traditional cognitive assessments such as positron emission tomography or cerebrospinal fluid analysis [16] have been proposed for AD detection. However, these approaches usually takes long time and are expensive. Such drawbacks may cause delay in treating AD, and put a heavy burden on public health, especially for seniors whose life expectancy is rapidly growing yet are more susceptible to AD [27]. Therefore, finding

[1]https://www.nia.nih.gov/health/alzheimers-disease-fact-sheet

a easy-to-use approach with commonly used available devices is the major challenge for early detection of AD.

Mild Cognitive Impairment (MCI) is considered as the first phase that patients start having biomarker evidence of brain changes that can eventually lead to AD [3]. During this phase, patients often develop problems linked with lexical-syntactic difficulties as their words are often hard to understand [23]. Moreover, although these patients can speak sentences that are morphologically, syntactically, and phonologically correct but the sentences are often vague and consists a lot of filler words [34]. However, such language and speech changes for MCI are often subtle and are nearly imperceptible to people other than friends and relatives. Till now, designing effective method to detect MCI is still an important while challenging research problem.

To this end, various machine learning methods have been proposed for the detection of language impairments indicating AD from speech, text [4, 44, 46, 27, 32]. Paper [46, 4, 44] use traditional machine learning techniques such as decision trees and support vector machines (SVM) to classify AD patients from the healthy group. However, these methods heavily rely on feature engineering to extract the relevant features, and extracting such features is often labor intensive, cost-prohibitive, and raises patient information leakage issues. There

are also some methods [27, 21] using deep learning to automatically extract useful information from the text or speech. Such model only consider the audio or text information separately without combining them together which still limits their detection performance. What's worse, most previous works only concentrated on the detection of dementia while the detection of MCI have been only based on relatively small datasets with less than 100 examples. This has become the biggest hurdle that prevents deep learning models from being deployed for MCI tasks.

This paper proposes several methods that consider and compare both text and speech information with deep neural networks. First, I present a new dataset that involves 721 speech recordings of three tasks from 329 subjects. Such a dataset provides a benchmark for me to develop approaches to combining the audio and text transcribed from speech information. Then, motivated by the observation in [6] that pre-trained language models (e.g. BERT [11]) outperform feature-based approaches on the AD detection task, I fine-tune such language models to transfer the general semantics and syntactic information inside them for the MCI task for text modeling and achieve an accuracy of 73%. For the speech information, I have tried on different neural architectures including Convolutional Neural Network (CNN), Recurrent

Neural Network (RNN) and Transformers with various tokenization strategies for tacking the pause information. The best accuracy is around 66% achieved by hierarchical LSTM model with token-level MFCC encoding. Detailed analysis are provided to recover the strengths and shortcomings of distinct types of speeches.

The main contributions of this paper are as follows:

- I present a new dataset called B-SHARP, which contains 721 speech recordings from 144 MCI patients and 185 health controls that can be used to develop NLP and acoustic models for the detection of Mild Cognitive Impairment (MCI).
- I employ pre-trained language models for the text data and leverage LSTM and Transformers for the speech part for the MCI detection task and conduct extensive experiments to demonstrate their performance over the B-SHARP dataset.
- I contrast the performance of these different approaches on the MCI detection task, and discuss advantages and disadvantages for existing differences.

The structure of this paper is described as follows: I discuss the existing

works in early detection of Alzheimer's disease in chapter 2. Then, I describe the proposed dataset in chapter 3. After discussing the text and speech model in chapter 4 and 5 respectively, I conclude my research findings and discuss the potential future work in chapter 6.

Chapter 2

Related Works

There are various existing works on detecting Mild Cognitive Impairment(MCI) using machine learning techniques. Both text-based and audio-based approaches have been used towards MCI classification. Asgari et al. [4] group spoken words using Linguistic Inquiry and Word Count (LIWC) from transcriptions of 14 MCI patients and 27 health controls, and apply SVM and random forest for the classification. Toth et al. [44] focus on the speech audio of 48 MCI patient and 38 health controls and manually extract multiple acoustic features such as lexical frequency of words, part-of-speech tags, and hesitation and speech rates, for classification. Fraser et al. [14] consider not only the text and speech, but also eye movements and head stabilization with 26 MCI patients and 29 health controls, and use logistic regression and SVM as classifier. All the previous works mentioned above are based on less than 100 subjects, and most of them manually extract linguistic features for

baseline classifier, while my work is based on 721 recordings from 329 subjects and involves many latest deep neural models.

The detection of dementia, is a similar but much more popular task. Becker et al. [8] present the famous dataset DementiaBank that involves 552 speech recording from 194 dementia patients and 99 health controls. Many research have been conducted based on this dataset. Fraser et al. [13] select the best 35 features from 370 features extracted and get an accuracy of 87.5%. Sabah et al. [2] demonstrates a robust method based only on acoustic features, which selects the best 20 features for classification and could reach an accuracy of 94.7%. My work differs from the previous ones in that I tackle the detection of MCI rather than dementia, which adds more challenges to this project since MCI is not as obvious as dementia.

Chapter 3

Dataset

3.1 Data Description

3.1.1 B-SHARP

My research is conducted based on data collected as part of the Brain, Stress, Hypertension, and Aging Research Program (B-SHARP) [27]. The dataset contains 185 normal controls and 144 MCI patients selected by specialists according to their neuropsychological and clinical assessments. Besides, all the subjects have been examined with multiple cognitive tests including the Montreal Cognitive Assessment (MoCA) [33] and the Boston Naming Test (BNT) [17]. Each year, each subject is asked to record a speech task protocol in Section 3.1.2 and 46.2%, 35.6% and 0.6% of the subjects have so far come back on their 2nd, 3rd and 4th year to take new voice recordings, respectively. Table 3.1 displays the statistics of B-SHARP dataset, from which one could

see that the subject ratio and recording ratio are similar and are around 50%, so the dataset could be considered as a balanced dataset.

For MoCA scores, patients with scores lower than 26 might have abnormal cognitive functions and it can be easily observed that the average MoCA score of MCI patients is way below 26 while the average MoCA score of control group is just above 26, which indicates it's probably more difficult to predict the control group correctly. For BNT score, patients with scores of 14 and above might have a neurological or psychiatric disorder. However, the table shows that the control group has average score above 14 and MCI patients have lower scores, which is not align with this rule. So from this prospective, this contradiction might add some challenges to the prediction.

	1st	2nd	3rd	4th	Sbj	Sbj Ratio	Rec	Rec Ratio	MoCA	BNT
C	26	83	75	1	185	56.23%	421	58.39%	26.2	14.2
M	32	69	42	1	144	43.77%	300	41.61%	21.5	13.4
T	58	152	117	2	329	100%	721	100%	24.2	13.9

Table 3.1: Statistics of B-SHARP dataset. C: the control group, M: the MCI group, T: total of both control and MCI groups, 1st/2nd/3rd/4th: # of subjects who came back for visits each year till the 1st/2nd/3rd/4th year, Sbj: # of subjects, Rec: # of recordings, MoCA/BNT: average scores and standard deviations from MoCA/BNT.

3.1.2 Speech Task Protocol

A speech task protocol is conducted each year for recording. Each subject is asked to speak 1-2 minutes on three topics respectively - Q1: daily activity, Q2: room environment, and Q3: picture description - guided by the same instructions as in A.1. An important thing to note is that since the subjects were guided to have similar activities before test, and were sitting in the same room and shown the same picture "The Circus Procession" in Fig A.1 during the test, their speech content won't vary a lot, which reduces potential variance.

But I also find some exceptions after I listen to all the recordings. Due to the COVID-19 pandemic, many subjects who revisited in 2020 could not conduct the speech task protocol in person and their speeches were instead recorded via phone call or virtual meeting. As a result, those subjects discussed their personal daily activities in Q1 and their own room environment in Q2, which are very different from most other subjects and thus could potentially influence the model prediction. Moreover, the speeches recorded remotely tend to have a worse quality than the normal recordings, which also makes future research more challenging.

3.2 Data Preprocessing

3.2.1 Transcription

Since the original dataset only contains the recordings, I need first transcribe them into text for future research on the transcription. I try three different online transcription tools – Temi[1], Amazon Transcribe[2], and Rev.ai[3] – on several recordings to compare their accuracy. I choose them because they all could generate a timestamp for each token while transcribing it into text at the same time, which is very useful for tokenization and analysis on speed and pauses. Among these three transcription tools, Temi is the one that Li et al. [27] used in his experiments and it will be easier for me to continue to work on it because he already fixed many transcription errors and segmented the transcription into three parts by tasks. But his revision on the transcription also introduces a problem that the timestamps of the tokens are messed up and thus I might lose a valuable information. For the second transcription tool, Amazon Transcribe, the problem is simply inaccurate transcription. I read several transcriptions generated by Amazon Transcribe and I find it hard to understand the context without the help of the original recordings. And

[1]Temi: https://www.temi.com/
[2]Amazon Transcribe: https://aws.amazon.com/transcribe/
[3]Rev.i: https://www.rev.ai/

finally, Rev.ai is chosen because it generates more accurate transcription and proper timestamps, but I also need to spend much time on formatting the data in this case.

3.2.2 Data Format

The transcription along with the timestamps is generated in a complicated format and is hard to use. Thus, I need to transform it into a clear format that will be easy to fit into models. A main challenge in this process is that each transcription and timestamp need to be manually segmented into three parts by reading the transcripts and listening to the recordings, since all the three tasks are recorded in a single file. The final data format consists of full transcriptions, a list of tokens and a list of corresponding timestamps of each task.

3.2.3 Data Split

In order to train both the text models and speech models, I split the dataset into three parts - training set, development set and evaluation set, where training set is used to help models learn the parameters, development set is to help me fine tune the models and select the best models, and evaluation set is to check if the models have good performance.

Table 3.2 displays the distribution of control group and MCI group in each set. The three sets are split based on the rules that (1) development set and evaluation set do not have shared subjects with training set in case the models rely solely on those specific subjects, (2) the ratio of control and MCI group for both subjects and recordings have little difference from the ratio in the entire dataset, and (3) the number of samples in training set is much larger than development and evaluation set to help models learn better and avoid overfitting.

Training Set				
	Subjects		Recordings	
	Count	Ratio	Count	Ratio
Control	135	57.20%	291	53.08%
MCI	101	42.80%	210	41.92%
Total	236	100.00%	501	100.00%
Development Set				
	Subjects		Recordings	
	Count	Ratio	Count	Ratio
Control	43	58.11%	60	60.00%
MCI	31	41.89%	40	40.00%
Total	74	100.00%	100	100.00%
Evaluation Set				
	Subjects		Recordings	
	Count	Ratio	Count	Ratio
Control	49	57.65%	70	58.33%
MCI	36	42.35%	50	41.67%
Total	85	100.00%	120	100.00%
All Data				
	Subjects		Recordings	
	Count	Ratio	Count	Ratio
Control	185	56.23%	421	58.39%
MCI	144	43.77%	300	41.61%
Total	329	100.00%	721	100.00%

Table 3.2: Data distribution for training/development/evaluation set and all data

Chapter 4

Text Model

4.1 Background

4.1.1 Word Embeddings

Many language systems rely on text representations as their first step to extract semantic information for text. Preliminary representation learning methods for text, such as one-hot encoding [7], N-grams [9] and TF-IDF [39] methods usually embed text into a sparse representation. Such methods, despite their simplicity, do not capture semantic correlations between features. What's worse, as the size of the vocabulary is usually large, then the representations are sparse and high-dimensional, which is computationally expensive. To tackle these challenges, distributed representation methods are proposed. Such methods represents text units with lower-dimensional vectors.

To obtain such distributed vector, there are various ways such as Word2Vec [31],

GloVe [37] and Fasttext [20]. These methods uses co-occurrence statistics to learn embeddings for each word, as they push the embedding of words that co-occur frequently more close while pull others apart. For example, Word2Vec propose two approaches to capture the word similarities: *CBoW* uses a window to predict the center word and *SkipGram* uses a center word to predict its surrounding words. However, these methods usually uses a fixed vector to represent words, which fail to consider the ambiguity issue, i.e. a word may have multiple meanings. Motivated by this, contextual word embedding methods are proposed [38, 11], where the word representations depend on the entire input sentence. These methods calculate the embedding based on the context information, thus being able to capture semantics information effectively.

4.1.2 Transformers and Pre-trained Language Models

Transformer is one of the most popular models for NLP tasks [45]. Different from the existing neural architectures such as Convolutional Neural Networks or Recurrent Neural Networks, the Transformer model contains stacked self-attention and point-wise, fully connected layers with residual connections without any explicit recurrent structure, which can be calculated

more efficiently.

Specifically, Vaswani et al. [45] propose a new attention function using the scaled dot-product as the attention score, expressed as

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{4.1}$$

where $Q, K, V \in \mathbb{R}^{\ell \times d}$ are the vector representations of all the words in the sequences of queries, keys and values accordingly. $\frac{1}{\sqrt{d_k}}$ is the scaling vector that pushes the softmax function into regions where it has extremely small gradients.

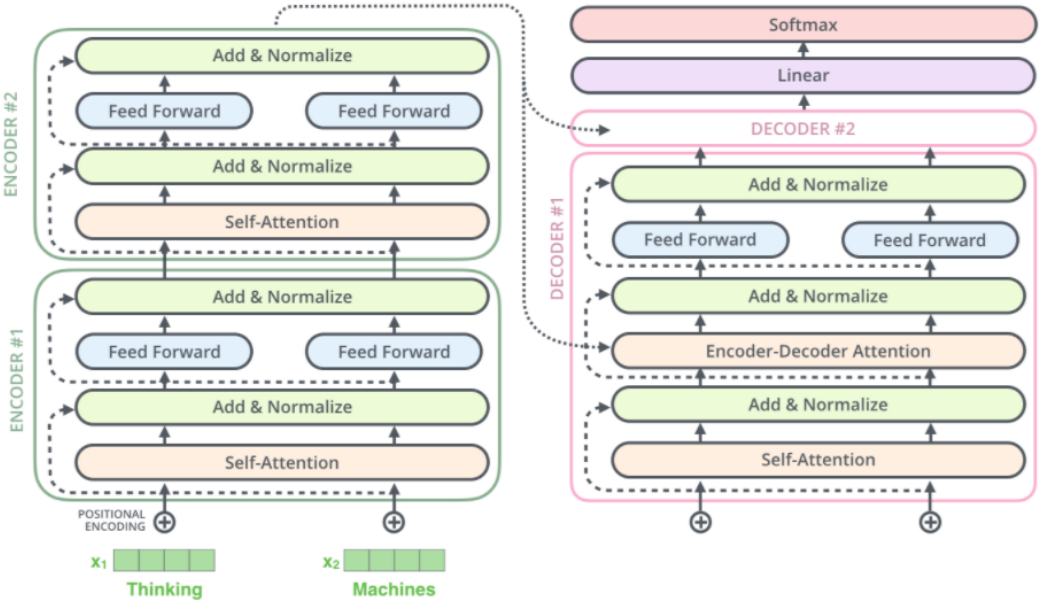

Figure 4.1: The architecture for Transformer model.

Based on the above attention function, [45] further propose multi-head attention to allow the model to jointly attend to information from different representation subspaces. In particular, for a multi-head attention module with m heads, we have

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_\text{m})W^O \tag{4.2}$$

where $head_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$ is the attention vector for the i-th attention head. In addition to the above multi-head attention modules, each layer in the encoder and decoder in Transformer contains a point-wise two-layer fully connected feed-forward network.

Motivated by the success of transformers, there are many studies attempted to leverage the power of transformers to incorporate context from both directions. Among them, pre-trained language models, such as BERT [11] and its variants (e.g., RoBERTa [29], ALBERT [26] and T5 [40]), have been shown to achieve state-of-the-art performance in many natural language understanding tasks, including text classification [43], named entity recognition [28] and question answering [22]. These models are essentially massive neural networks based on bi-directional transformer architectures, and are trained using open-domain data in a completely unsupervised manner. For example, the

popular BERT-base model contains 110 million parameters, and is trained using the Books Corpus (800 million words) and English Wikipedia (2500 million words). More importantly, many pre-trained language models have been publicly available online as one does not need to train them from scratch. The massive training data enables pre-trained language models to have strong expressive power for capturing general semantics and syntactic information effectively. When applying pre-trained language models to downstream tasks, such knowledge can be efficiently transferred to the downstream domains through efficient and scalable stochastic gradient-type algorithms.

4.2 Experiments

To get a sense of the models' learning ability on each task, I first build a transformer model for each individual question. I implement various kinds of transformer models on this dataset including BERT, BERT-large, ALBERT, ALBERT-large, RoBERTa, RoBERTa-large.

Table 4.1 shows the accuracy of the text models on Q1, Q2, Q3 individually, and it is easy to notice that BERT generally has the best and most stable performance, which also verifies that transformer models have the ability to extract features from the text, and also the transcriptions indicate the mental

Model	**Q1**	**Q2**	**Q3**
BERT	**0.69**	0.68	**0.68**
ALBERT	0.6	0.63	0.63
RoBERTa	0.64	**0.7**	0.67
BERT-large	0.63	0.64	0.63
ALBERT-large	0.61	0.63	0.62
RoBERTa-large	0.6	0.64	0.62

Table 4.1: The accuracy of the text models on task Q1, Q2, Q3 on evaluation set individually.

health condition of subjects.

Then the three tasks are integrated together to produce an ensemble model in two ways. One is in early fusion, which means the three models for the three tasks are trained separately and their predictions are used to vote for a final prediction, and this process is also known as majority voting. The other one is in late fusion, which means the three embedding from the three models are concatenated and updated together, and generate the final prediction as a whole. Figure 4.2 is the model that Li et al. [27] propose, which is also the late fusion ensemble model I use in my experiment.

Table 4.2 displays a comparison between my and Li's [27] approaches and results. Li does 5-fold cross validation on 650 samples, while I have 71 more samples this year that allows me to generate training, development and evaluation set. Li's model uses 9 embeddings from all the three questions

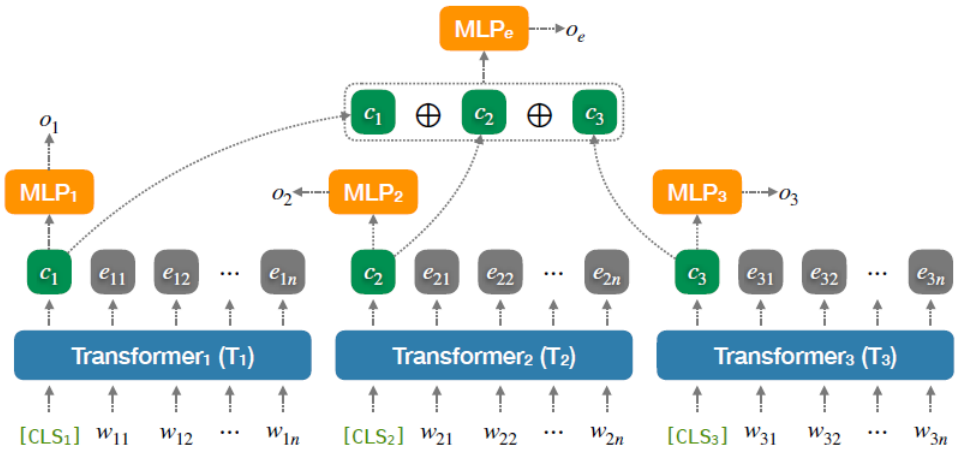

Figure 4.2: Ensemble Model by Li et al. [27]

trained by BERT/ALBERT/RoBERTa, while mine only uses 3 embeddings from the three questions trained by BERT since ALBERT and RoBERTa model do not work well in my case. And in the end, my late fusion model has an accuracy of 0.7 and early fusion reaches 0.73, which is very similar to his ensemble model accuracy 0.74. The accuracy of my ensemble model is higher than each individual model because ensemble model considers all the three tasks at the same time and could reduce the potential variance element that causes prediction error. Thus, the ensemble model could be more stable and have better performance than each individual model.

In conclusion, the ensemble hierarchical BERT model is able to powerfully extract features from text and give a promising accuracy of 73% to distinguish

MCI patients to normal controls, probably because transcriptions provide the context of the speeches which reflect whether the subjects have normal logical thinking or not. Besides, compared to Li's previous model, mine has similar accuracy but is more robust and more time and memory efficient because of the technique and model I use.

	Mine	**Li's**
Acc(late)	0.7	-
Acc(early)	0.73	0.74
Model	B_e	$B_e + A_e + R_e$
Technique	train/dev/test set	5-fold cross validation
# samples	721	650

Table 4.2: Comparison between my and Li' approaches and results. B_e: the model uses embeddings from all the three questions trained by BERT (3 embeddings together), $B_e + A_e + R_e$: the model uses embeddings from all the three questions trained by BERT/ALBERT/RoBERTa (9 embeddings together).

Chapter 5

Speech Model

5.1 Background

5.1.1 Mel Frequency Cepstral Coefficients (MFCC)

In audio processing, the Mel Frequency Cepstral Coefficients [30] is a representation of the short-term power spectrum of a sound, which is calculated based on a linear cosine transform of a log power spectrum on a nonlinear mel scale of frequency. With the audio as the input, the calcualtion of MFCC usully contains the following steps [41]:

- Take discrete Fourier transform and map the powers of the spectrum onto the mel scale with the triangular overlapping windows [10].
- Take the Log of amplitude spectrum for re-scaling.
- Take the discrete cosine transform of the log powers, as if it were a signal. The MFCCs are the amplitudes of the resulting spectrum.

MFCC have been commonly used as the backbone feature in speech recognition [1, 42], and the deep learning methods usually stack layers over MFCC features for downstram tasks.

5.1.2 Multi-layer Perceptron (MLP)

A multilayer perceptron (MLP) is a class of feedforward neural network. An MLP usually consists of at least three layers: an input layer, a hidden layer and an output layer. Each node is a neuron that uses a nonlinear activation function. MLP is the one of the most simple network architecture, as it serves as a universal function approximator.

5.1.3 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) [24] is a variation of feed-forward neural network, which is first proposed for image recognition. In CNN, the convolution and pooling layers are proposed to characterize the most salient information from the input through convolution and pooling operations. As a result, it transforms a image from a large-size representation to a smaller one and extract important local shapes. Later on, CNN has been adapted for sequential data such as text and audio [1, 18, 48]. Take audio as an example, each audio can be transformed to a spectral feature matrix via

MFCC as it is a sequence of voice. CNN is capable of modeling such local frequency structures by allowing the convolutional layer to receive input only from features representing a limited bandwidth of the whole speech spectrum. CNN model is computationally simple and efficient to train, and has been widely used as the feature extractor for audios. However, it cannot capture the sequential information to store the historical knowledge, which hampers its performance over the sequential data.

5.1.4 Recurrent Neural Network (RNN)

Recurrent Neural Network is a specific type of neural networks to tackle the sequential data. Different to the fixed contextual windows used as inputs in MLPs and CNNs, the RNN model feeds the activations from previous time steps as input to the network for the current input. As a result, such structure can better leverage the contextual information and have been widely used for text, video and audio analysis [15, 25, 42]. Long Short-Term Memory (LSTM) Network [19] is the one of the most popular RNN, which have been shown to perform better than RNNs on learning context-free and context-sensitive languages. The architecture of LSTMs contains special units called *memory blocks* in the recurrent hidden layer. The memory blocks contain memory

cells with self-connections storing (remembering) the temporal state of the network in addition to special multiplicative units called gates to control the flow of information. Each memory block contains an input gate which controls the flow of input activation into the memory cell and an output gate which controls the output flow of cell activation into the rest of the network. The mathematical formulation of LSTM model is shown as below.

$$
\begin{aligned}
f_t &= \sigma_g \left(W_f x_t + U_f h_{t-1} + b_f \right) \\
i_t &= \sigma_g \left(W_i x_t + U_i h_{t-1} + b_i \right) \\
o_t &= \sigma_g \left(W_o x_t + U_o h_{t-1} + b_o \right) \\
\tilde{c}_t &= \sigma_c \left(W_c x_t + U_c h_{t-1} + b_c \right) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \\
h_t &= o_t \circ \sigma_h \left(c_t \right),
\end{aligned}
\tag{5.1}
$$

where W denotes weight matrices (e.g. W_i is the matrix of weights from the input gate to the input), b denote bias vectors, σ is the logistic sigmoid function, and i, f, o and c are respectively the input gate, forget gate, output gate and cell activation vectors and $\odot$ is the element-wise product of the vectors and g and h are the cell input and cell output activation functions. Compared with CNN models, LSTMs usually take longer time to train but

can encode some contextual information for audio data. However, such model still suffer from the vanishing gradient problem [36] which limits its ability over long sequence.

5.1.5 Attention Mechanism

Attention is first proposed in [5] for the machine translation task. The attention mechanism strengthens not only the ability of RNN in capturing the long-range dependencies but also the interpretability as it enable the model to focus a subpart of the input that is most relevant to the task. The attention module is usually designed as a linear neural network that can be trained along with the whole neural network. The attention computation can be formulated as follows,

$$\begin{aligned} c &= \textstyle\sum_i \alpha_i s_i \\ \alpha_i &= \frac{\exp(f(h,s_i))}{\sum_i \exp(f(h,s_i))} \\ f\left(h,s\right) &= h^T \tanh\left(Ws\right) \end{aligned} \tag{5.2}$$

where s represents the features for each component, W is the learnable parameters for calculating the attention, h is the query vector which denotes current mobility status from the recurrent layer, f represents the score function, and c is the context output representing the global feature for the

whole input.

5.2 Experiments

5.2.1 Speech-level Embedding

To prepare for speech models, the audio recording of each subject is first split into three parts based on the timestamps of each task. For the baseline model, I encode each part of recording by MFCC [30], which has dimension of 13, and the sequence length depends on the length of audio. Thus, due to the different time duration of each speech, the feature length mostly varies from 8,000 to 15,000. And after truncating the longer ones and padding the shorter ones with zeros, I unify all the feature vectors for each task to have a sequence length of 12,000.

Given the input for the speech model, I conduct the experiment on multiple different neural models including multi-layer perceptron(MLP), MLP with attention mechanism(MLP+Attention), Convolutional Neural Networks(CNN), Recurrent Neural Networks(RNN), CNN and RNN together(CNN+RNN), and three-layers transformers, and compare these models' performance on speeches. Table 5.1 shows the test accuracy of each model on each task. It's obvious that RNN model has the best performance on all the of three tasks

because it can better tackle the sequential information in the audio. The three-layers transformer model also has higher accuracy than most others, but it takes much space and time for training and thus is not efficient enough.

Model	**Q1**	**Q2**	**Q3**
MLP	0.55	0.58	0.57
MLP+Attention	0.62	0.63	0.63
CNN	0.58	0.58	0.58
RNN	**0.65**	**0.64**	**0.65**
CNN+RNN	0.58	0.6	0.58
Transformers	0.64	**0.64**	0.63

Table 5.1: The accuracy of the speech models with speech-level embedding on task Q1, Q2, Q3 on evaluation set individually.

5.2.2 Token-level Embedding

As a common practice in most of previous works, speech-level encoding to MFCC does show its ability to extract features from audio for prediction, but I think this method cannot harness the token information and is also memory-inefficient - many audios have long time duration and directly feed them into neural models can easily cause memory overflow. Thus, I propose a model with token-level embedding, utilizing the timestamps for each token provided in transcriptions, to see if it could potentially solve this problem and give a better result.

In order to get token-level encoding, the recording is first split into tokens

by the timestamps, and then MFCC feature from each token is extracted. However, for the pauses between each two tokens, I apply four different ways to encode them, to find out if the pauses are able to bring performance gain: **(1)** No pause is encoded at all, which means only MFCC features for word tokens are used for training. **(2)** Each word and the two pauses before and after it is considered as a single token and MFCC feature is extracted for training. In this case, every two tokens would have an overlap for a pause, so that the model might potentially interpret the change of sequential audio. **(3)** The time duration of each pause token passes through a linear layer to get a pause token embedding, which is then concatenated to each word token embedding before it. **(4)** The time duration of each pause token also passes through the linear layer to get pause token embedding, but instead of concatenating to the word token embedding, it is considered as an individual token, and is placed right after each word token in the correct order.

After the token embeddings are obtained by these four methods individually, they are then put into a LSTM layer [19] to get speech embedding, which finally passes through a linear layer to get the prediction.

Figure 5.1 is a graph indication of the method (4) mentioned above. This hierarchical LSTM model is only for each individual task. For each encoding

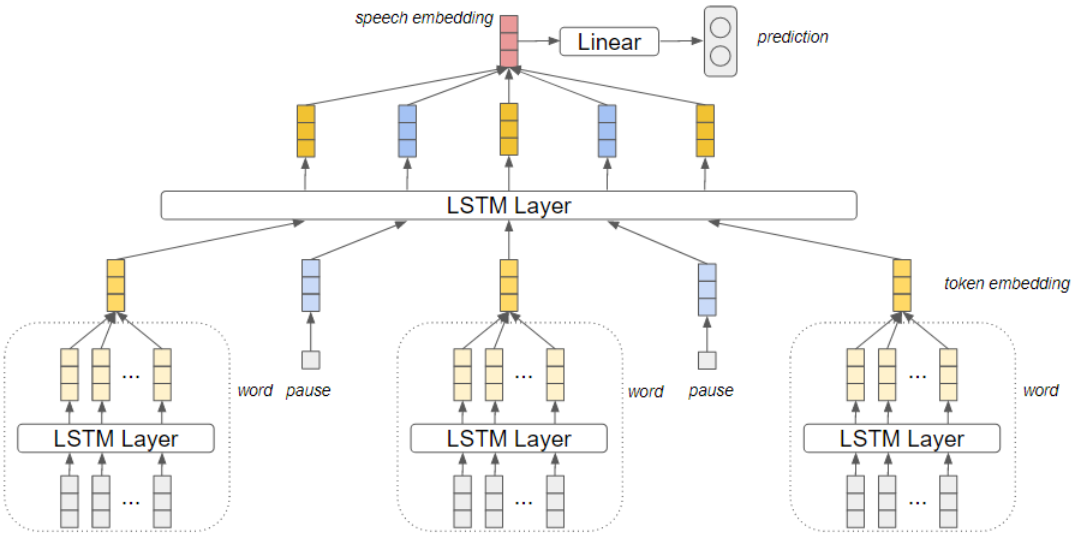

Figure 5.1: Hierarchical LSTM model encoding with method (4) the time duration of each pause token passes through the linear layer to get pause token embedding, which is considered as an individual token, and is placed right after each word token in the correct order.

method, I also build an ensemble model with early fusion similar to the ensemble model on the text part.

The table 5.2 shows the accuracy of this hierarchical LSTM model with the four different encoding methods mentioned above on the three task Q1, Q2, Q3 individually and the accuracy of ensemble model. From the table, one can notice that modeling pause as a specific token generally has the best performance. However, the accuracy of ensemble model is not better than individual models, probably because the individual models have contradictory signals and some of them might affect the prediction result in a negative way.

Besides, it can also be observed that the model without pause embedding has the worst accuracy, which indicates that the pause information could bring performance gain. Also, the low accuracy of the second encoding method also implies that direct modeling the pause MFCC information is not a good way.

Encoding Method	**Q1**	**Q2**	**Q3**	**Ensem**
Only word token embedding by MFCC	0.61	0.64	0.6	0.61
Word + two pauses token embedding by MFCC	0.61	0.65	0.64	0.61
Word token embedding by MFCC concatenated with pause token embedding by linear layer	**0.68**	0.64	0.63	0.64
Word token embedding by MFCC + pause token embedding by linear layer as an individual token	0.64	**0.67**	**0.67**	**0.66**

Table 5.2: The accuracy of the hierarchical LSTM model with four different encoding methods to get token-level embedding on evaluation set task Q1, Q2, Q3 individually and the accuracy of ensemble model.

5.2.3 Pretrained Language Model

Since the hierarchical LSTM model encoded by MFCC does not show a very promising result, I propose another model that fine tunes the pretrained language model for audio, just as the BERT model for the text part.

The pretrained language model[1] proposed by Winata et al. [47] is a transformer-based end-to-end code switching system for automatic speech recognition(ASR). Instead of encoding the input by MFCC features, this ASR model accepts a spectrogram of the audio as the input, and predicts characters from it. The models consists of 2 layers of encoders and 4 layers of decoders. But note that the decoders are only used for pretraining but not in the finetuning process. Convolutional layers are to learn a universal speech representation for input embedding, and multi-head attention is used to allow the model to jointly attend to information from different representation subspaces at a different position. A detailed comparison between this audio transformer model and the BERT base text model are shown in table 5.3 below.

Model	**Audio Transformer**	**BERT(base)**
Input	Spectrogram of audio	Text tokens
Embedding	CNNs (VGG)	Embedding matrix
# layers	M=2,N=4	M=12
# attention heads	A=4	A=12
Pretrain task	ASR	MLM + NSP

Table 5.3: Comparison between audio transformer and BERT text model. M: # encoders, N: # decoders, ASR: Automatic Speech Recognition, MLM: Masked Language Modeling, NSP: Next Sentence Prediction

Since the parameters from the pretrained model is not provided, I first

[1]https://github.com/gentaiscool/end2end-asr-pytorch

pretrain this language model on the dataset Librispeech [35] based on the given code, and save the best model during pretraining. Then the decoder of the model is changed from character prediction to binary classification for further fine tuning on my dataset.

Model	**Q1**	**Q2**	**Q3**
Pretrained language model	0.64	0.66	0.65

Table 5.4: The accuracy of the pretrained language model on evaluation set task Q1, Q2, Q3 individually.

Table 5.4 shows the accuracy of this pretrained language model. It's interesting that the model does not have better performance than previous models, which might indicate these two models both are not very capable of extracting feature from audio, or the audio itself does not contain much context and thus is harder to provide signals for final prediction.

Chapter 6

Conclusion and future work

All the speech models get similar results, among which hierarchical LSTM model with token-level MFCC encoding is a bit better than others, which demonstrates that token information does include valuable information, and considering pauses could improve the model performance. However, compared with the text model, none of the speech models have better performance than the BERT model on the text side, which have an accuracy of 73%. From those results, one can conclude that, for prediction of early Alzheimer's Disease, the text model has a better performance because it has richer information on the context of the speech; the speech model, though does not do well in the prediction, could still capture many other aspects such as tones and pauses that are ignored by the text model.

For future work, I'll first try to further improve the text model because BERT model is only a baseline, and text contains much more information

than audio that has potentials to be further extracted. Besides, I'll combine the text model and speech model together to capture both context and audio information at the same time, and see if this could further improve the model accuracy on this task. And if I could get a promising result from this ensemble model of text and audio, I'll try to build a same model for the dataset DementiaBank [8] to check if the model is robust enough to produce a high accuracy on similar task.

Appendix A

Appendix

A.1 Speech Task Protocol

Task	Instruction
Q1	I would like you to describe to me everything we did from the moment we met today until now. Please try to recall as many details as possible in the order the events actually happened where we met, what we did, what we saw, where we went, and what you felt or thought during each of these events.
Q2	I would like you to describe everything that you see in this room.
Q3	I am going to show you a picture and ask you to describe what you see in as much detail as possible. You can describe the activities, characters, and colors of things you see in this picture.

Table A.1: The same instructions of the three speech tasks Q1, Q2 and Q3 provided to each subject.

Table A.1 shows the instructions that guide each subject for the speech protocol. For Q3, the picture shown to the subjects is shown in Figure A.1, copyrighted by the *McLoughlin Brothers* as part of the Juvenile Collection.

Figure A.1: Picture of "The Circus Procession" used in Q3 of the speech task protocol.

Bibliography

[1] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, and Gerald Penn. Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition. In *2012 IEEE international conference on Acoustics, speech and signal processing (ICASSP)*, pages 4277–4280. IEEE, 2012.

[2] Sabah Al-Hameed, Mohammed Benaissa, and Heidi Christensen. Simple and robust audio-based detection of biomarkers for alzheimer's disease. In *7th Workshop on Speech and Language Processing for Assistive Technologies (SLPAT)*, pages 32–36, 2016.

[3] Marilyn S Albert, Steven T DeKosky, Dennis Dickson, Bruno Dubois, Howard H Feldman, Nick C Fox, Anthony Gamst, David M Holtzman, William J Jagust, Ronald C Petersen, et al. The diagnosis of mild cognitive impairment due to alzheimer's disease: recommendations from

the national institute on aging-alzheimer's association workgroups on diagnostic guidelines for alzheimer's disease. *Alzheimer's & dementia*, 7 (3):270–279, 2011.

[4] Meysam Asgari, Jeffrey Kaye, and Hiroko Dodge. Predicting mild cognitive impairment from spontaneous spoken utterances. *Alzheimer's & Dementia: Translational Research & Clinical Interventions*, 3(2):219–228, 2017.

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[6] Aparna Balagopalan, Benjamin Eyre, Frank Rudzicz, and Jekaterina Novikova. To bert or not to bert: Comparing speech and language-based approaches for alzheimer's disease detection. *Proc. Interspeech 2020*, pages 2167–2171, 2020.

[7] Joseph E Beck and Beverly Park Woolf. High-level student modeling with machine learning. In *International Conference on Intelligent Tutoring Systems*, pages 584–593. Springer, 2000.

[8] James T Becker, François Boiler, Oscar L Lopez, Judith Saxton, and

Karen L McGonigle. The natural history of alzheimer's disease: description of study cohort and accuracy of diagnosis. *Archives of Neurology*, 51(6):585–594, 1994.

[9] William B Cavnar, John M Trenkle, et al. N-gram-based text categorization. In *Proceedings of 3rd annual symposium on document analysis and information retrieval*, volume 161175. Citeseer, 1994.

[10] Steven Davis and Paul Mermelstein. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing*, 28(4):357–366, 1980.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

[12] Steven H Ferris and Martin Farlow. Language impairment in alzheimer's

disease and benefits of acetylcholinesterase inhibitors. *Clinical interventions in aging*, 8:1007, 2013.

[13] Kathleen C Fraser, Jed A Meltzer, and Frank Rudzicz. Linguistic features identify alzheimer's disease in narrative speech. *Journal of Alzheimer's Disease*, 49(2):407–422, 2016.

[14] Kathleen C Fraser, Kristina Lundholm Fors, Marie Eckerström, Fredrik Öhman, and Dimitrios Kokkinakis. Predicting mci status from multimodal language data using cascaded classifiers. *Frontiers in aging neuroscience*, 11:205, 2019.

[15] Michael Freitag, Shahin Amiriparian, Sergey Pugachevskiy, Nicholas Cummins, and Björn Schuller. audeep: Unsupervised learning of representations from audio with deep recurrent neural networks. *The Journal of Machine Learning Research*, 18(1):6340–6344, 2017.

[16] Denise C Fyffe, Shubhabrata Mukherjee, Lisa L Barnes, Jennifer J Manly, David A Bennett, and Paul K Crane. Explaining differences in episodic memory performance among older african americans and whites: the roles of factors related to cognitive reserve and test bias. *Journal of the International Neuropsychological Society: JINS*, 17(4):625, 2011.

[17] Harold Goodglass, Edith Kaplan, and Sandra Weintraub. *Boston naming test*. Lea & Febiger Philadelphia, PA, 1983.

[18] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *2017 IEEE international conference on acoustics, speech and signal processing*, pages 131–135. IEEE, 2017.

[19] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[20] Armand Joulin, Édouard Grave, Piotr Bojanowski, and Tomáš Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, 2017.

[21] Sweta Karlekar, Tong Niu, and Mohit Bansal. Detecting linguistic characteristics of alzheimer's dementia by interpreting neural models. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 701–707, 2018.

[22] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, 2020.

[23] Mikyong Kim and Cynthia K Thompson. Verb deficits in alzheimer's disease and agrammatism: Implications for lexical organization. *Brain and language*, 88(1):1–20, 2004.

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[25] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. Recurrent convolutional neural networks for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29, 2015.

[26] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020.

[27] Renxuan Albert Li, Ihab Hajjar, Felicia Goldstein, and Jinho D. Choi. Analysis of hierarchical multi-content text classification model on B-SHARP dataset for early detection of Alzheimer's disease. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 358–365, December 2020.

[28] Chen Liang, Yue Yu, Haoming Jiang, Siawpeng Er, Ruijia Wang, Tuo Zhao, and Chao Zhang. Bond: Bert-assisted open-domain named entity recognition with distant supervision. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1054–1064, 2020.

[29] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[30] Beth Logan. Mel frequency cepstral coefficients for music modeling. In *In International Symposium on Music Information Retrieval*. Citeseer, 2000.

[31] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26: 3111–3119, 2013.

[32] Amish Mittal, Sourav Sahoo, Arnhav Datar, Juned Kadiwala, Hrithwik Shalu, and Jimson Mathew. Multi-modal detection of alzheimer's disease from speech and text. *arXiv preprint arXiv:2012.00096*, 2020.

[33] Ziad S Nasreddine, Natalie A Phillips, Valérie Bédirian, Simon Charbonneau, Victor Whitehead, Isabelle Collin, Jeffrey L Cummings, and Howard Chertkow. The montreal cognitive assessment, moca: a brief screening tool for mild cognitive impairment. *Journal of the American Geriatrics Society*, 53(4):695–699, 2005.

[34] Marjorie Nicholas, Loraine K Obler, Martin L Albert, and Nancy Helm-Estabrooks. Empty speech in alzheimer's disease and fluent aphasia. *Journal of Speech, Language, and Hearing Research*, 28(3):405–410, 1985.

[35] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015*

IEEE international conference on acoustics, speech and signal processing (ICASSP), pages 5206–5210. IEEE, 2015.

[36] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. PMLR, 2013.

[37] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[38] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, 2018.

[39] Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer, 2003.

[40] Adam Roberts, Colin Raffel, Katherine Lee, Michael Matena, Noam Shazeer, Peter J. Liu, Sharan Narang, Wei Li, and Yanqi Zhou. Exploring the limits of transfer learning with a unified text-to-text transformer. Technical report, Google, 2019.

[41] Md Sahidullah and Goutam Saha. Design, analysis and experimental evaluation of block based transformation in mfcc computation for speaker recognition. *Speech communication*, 54(4):543–565, 2012.

[42] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv preprint arXiv:1402.1128*, 2014.

[43] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer, 2019.

[44] László Tóth, Ildikó Hoffmann, Gábor Gosztolya, Veronika Vincze, Gréta Szatlóczki, Zoltán Bánréti, Magdolna Pákáski, and János Kálmán. A speech recognition-based solution for the automatic detection of mild cognitive impairment from spontaneous speech. *Current Alzheimer Research*, 15(2):130–138, 2018.

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30: 5998–6008, 2017.

[46] Sebastian Wankerl, Elmar Nöth, and Stefan Evert. An n-gram based approach to the automatic diagnosis of alzheimer's disease from spoken language. In *INTERSPEECH*, pages 3162–3166, 2017.

[47] Genta Indra Winata, Andrea Madotto, Chien-Sheng Wu, and Pascale Fung. Code-switched language models using neural based synthetic data from parallel sentences. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 271–280, 2019.

[48] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in Neural Information Processing Systems*, 28:649–657, 2015.