INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

WavBERT: Exploiting Semantic and Non-semantic Speech using Wav2vec and BERT for Dementia Detection

Youxiang Zhu[1]*, Abdelrahman Obyat*[1]*, Xiaohui Liang*[1]*, John A. Batsis*[2]*, and Robert M. Roth*[3]

[1]Department of Computer Science, University of Massachusetts Boston, MA, USA
[2] School of Medicine, University of North Carolina, Chapel Hill, NC, USA
[3] Geisel School of Medicine at Dartmouth, Lebanon, NH, USA

{Youxiang.Zhu001, Abdelrahman.Obyat001, Xiaohui.Liang}@umb.edu
John.Batsis@unc.edu, Robert.M.Roth@hitchcock.org

Abstract

In this paper, we exploit semantic and non-semantic information from patient's speech data using Wav2vec and Bidirectional Encoder Representations from Transformers (BERT) for dementia detection. We first propose a basic WavBERT model by extracting semantic information from speech data using Wav2vec, and analyzing the semantic information using BERT for dementia detection. While the basic model discards the non-semantic information, we propose extended WavBERT models that convert the output of Wav2vec to the input to BERT for preserving the non-semantic information in dementia detection. Specifically, we determine the locations and lengths of inter-word pauses using the number of blank tokens from Wav2vec where the threshold for setting the pauses is automatically generated via BERT. We further design a pre-trained embedding conversion network that converts the output embedding of Wav2vec to the input embedding of BERT, enabling the fine-tuning of WavBERT with non-semantic information. Our evaluation results using the ADReSSo dataset showed that the WavBERT models achieved the highest accuracy of 83.1% in the classification task, the lowest Root-Mean-Square Error (RMSE) score of 4.44 in the regression task, and a mean F1 of 70.91% in the progression task. We confirmed the effectiveness of WavBERT models exploiting both semantic and non-semantic speech.

Index Terms: Speech analysis, automatic speech recognition, non-semantic information, dementia detection

1. Introduction

Researchers have exploited spontaneous speech for early detection of Alzheimer's disease, as the collection of speech data is more practical and less costly compared to conventional cognitive assessment methods such as neuropsychological evaluation [1] and Magnetic Resonance Imaging (MRI) [2]. In the 2020 Alzheimer's Dementia Recognition through Spontaneous Speech (ADReSS) challenge, researchers studied the spontaneous speech dataset [3] and demonstrated that transcript-based models are more effective in dementia detection than audio-based models [4, 5, 6, 7]. We envisioned that the low performance of the audio-based models is due to the large variance and hard-to-interpret nature of audio signals [8, 9]. In comparison, transcript-based models were built using the manual transcripts from human transcription, which takes advantage of the human transcriber's knowledge, including rules of the description task and information units within the picture. However, human transcription is a costly and impractical process, which prevents the speech-based evaluation from being a fully automatic approach. The 2021 ADReSS speech only (ADReSSo) challenge thus aims at the development of fully automatic models that detect dementia using only speech data [10].

Automatic Speech Recognition (ASR) aims to generate ASR transcripts automatically from speech audio data. ASR transcripts can be used as the inputs to transcript-based models when manual transcripts are not available. However, we have two concerns. First, ASR might generate a transcript with uncertain errors, especially for the speech from patients with cognitive impairment. Such uncertain errors might negatively affect the performance of transcript-based models. Second, the integration of ASR could analyze both semantic and non-semantic information for a more accurate dementia detection, while transcript-based models focus on the analysis of semantic information of the transcripts only. Previous research demonstrated the usefulness of non-semantic information in dementia detection, such as filled and silent pauses [11, 12, 13], paralinguistic features [10, 14], and Mel Frequency Cepstral Coefficient (MFCC) [15, 8, 9]. As such, we seek an effective integration of ASR and transcript-based models for enhanced dementia detection. For the transcript-based models, Bidirectional Encoder Representations from Transformers (BERT) dominates the Natural language processing (NLP) research due to its power of self-supervised training and transfer learning strategy [16]. It consists of two steps: i) pre-training a model with unlabeled data and self-supervised training strategy, and ii) fine-tuning the model with downstream data and tasks. In the ADReSS 2020 challenge, it was proven that transcript-based model BERT outperforms traditional machine-or-deep learning models using handcrafted features in dementia detection [17].

We recently explored transfer learning over audio dataset directly for dementia detection but achieved limited performance [8, 9]. Our conclusion is that the selected pre-trained audio models did not extract good representation from the audio data of the dementia task from the dementia detection perspective. However, transfer learning has been effective in ASR; researchers proposed Wav2vec model [18] and the corresponding ASR achieved state-of-the-art Word Error Rate (WER) on the LibriSpeech dataset [19]. We thus applied the Wav2vec ASR model to generate ASR transcripts from the 2020 ADReSS dataset, compared the ASR transcripts with the manual transcripts, and found the high similarity of the two transcripts.

In this paper, we first propose a basic WavBERT model to generate ASR transcripts via Wav2vec, use the ASR transcripts and dementia-related labels to fine-tune BERT, and derive the dementia detection results of the testing dataset using the fine-tuned BERT. While the basic WavBERT model discards the non-semantic information, we propose two extended WavBERT models that utilize the intermediate results of the

http://dx.doi.org/10.21437/Interspeech.2021-332

ASR, containing the non-semantic information. Specifically, the **first method** is to derive the locations and lengths of inter-word pauses by counting the blank tokens and separate tokens from the intermediate results of the Wav2vec ASR. Furthermore, we do not manually set the thresholds for pauses. Instead, we propose automatic methods that use BERT or training samples to determine the thresholds. The **second method** is to convert the Wav2vec output embedding to the BERT input embedding using a pre-trained embedding conversion network. The module is pre-trained with a large-sized audio dataset and its corresponding ASR transcripts and assists the fine-tuned BERT to detect dementia using both semantic and non-semantic information from the speech data. Our contributions are three-fold:

First, we propose a basic WavBERT model that concatenates Wav2vec ASR with BERT, enabling an automatic process of dementia detection.

Second, we extend the WavBERT model to determine the locations and lengths of pauses using the ASR intermediate results. The thresholds for setting pauses are automatically generated. The extended WavBERT model achieves the highest accuracy of 83.1% in the classification task and the lowest Root-Mean-Square Error (RMSE) score of 4.44 in the regression task.

Third, we extend the WavBERT model by converting the Wav2vec output embedding to the BERT input embedding for preserving non-semantic information. The extended WavBERT achieves the highest accuracy of 70.91% in the progression task.

2. ADReSSo dataset

The ADReSSo challenge consists of three tasks, an AD classification task, a Mini-Mental State Examination (MMSE) regression task, and a cognitive decline progression task [10]. The first two tasks share the same data, including 237 audio files, which were collected using a Cookie Theft picture description task from the Boston Diagnostic Aphasia Exam [20]. The data is balanced with class, age, and gender. The data for the cognitive decline progression task was collected from a category fluency task, including 105 audio files. The first-round data was provided as the baseline, and the second-round data was collected in two years and used for inferring cognitive decline. The data of this task is unbalanced; the non-decline samples are significantly more than the decline samples. In the challenge, 70% of both datasets were used for training and 30% for testing [10].

3. Basic WavBERT

We propose a basic WavBERT model consisting of Wav2vec ASR and BERT, labeled to path 1 in the Figure 1. The basic model converts speech data to ASR transcripts and inputs the ASR transcripts to BERT for dementia detection.

Wav2vec aims to learn the representations from speech data using self-supervised training [18]. As shown on the left of the Figure 1, Wav2vec first inputs speech data into a Convolutional Neural Network (CNN) to obtain the latent representations, which are then inputted into a transformer encoder. The transformer encoder generates context representations in the output embedding and employs a pre-training task following the self-supervised training strategy [18]. After pre-training, Wav2vec uses a fine-tuning process with a character inference component, optimized with a Connectionist Temporal Classification (CTC) loss. The character inference component consists of a 1D convolutional layer and a softmax layer. The convolutional layer convolutes according to the time dimension using both kernel size and stride set to 1. The output of the character inference component can be 26 English letters, single quote ', blank token <s>, and separator token |. Finally, Wav2vec merges consecutively repeated characters, removes blank tokens, and uses separator tokens to separate words. The transcript has no punctuation, contains the semantic information of the speech data, and can be inputted to transcript-based models.

BERT derives the general representation of the language model by employing a pre-training process with large-scale datasets (i.e., BooksCorpus and Wikipedia) [16]. BERT generally includes the four steps depicted on the right of the Figure 1. Given a transcript, BERT first pre-processes the transcript with the WordPiece tokenizer [21], splits words into sub-word-level tokens, and then adds special tokens [CLS] and [SEP]. All tokens are converted to an input embedding, which is further inputted to a transformer encoder [22] to obtain the output embedding. Two pre-training tasks were adopted: Masked Language Model (MLM) and Next Sentence Prediction (NSP). In MLM, given that some of the input tokens are masked, the classification objective is to use the embedding of the unmasked tokens to infer the masked tokens. In NSP, given that a single [SEP] token is inserted between two selected sentences, a binary classification objective is to infer whether the first sentence is followed by the second in the transcript.

Inference layers. For the dementia detection task, we use the BERT output embedding of all the tokens except for the [CLS] token as the input of the inference layers. Specifically, we use a 1D convolutional layer with both kernel size and stride set to 1. The number of neurons is equal to the hidden size of the BERT. We use a Global Average Pooling (GAP) layer to calculate an averaged vector according to the time dimension, and use a Fully Connected (FC) layer with softmax or LeakyReLU for the classification or regression tasks, respectively.

4. Extended WavBERT

We extend WavBERT models with ASR pause preservation and embedding conversion, shown in paths 2 and 3 of the Figure 1.

4.1. ASR pause preservation

Force alignment methods are often used to align transcripts and audio data and determine both inter-word and inter-sentence pauses [13]. However, as the ASR produces uncertain errors, force alignment between the ASR transcript and audio data might not be effective. In our model, we exploit the CTC property of Wav2vec for determining the pauses. Wav2vec produces blank tokens and separate tokens as intermediate results. As such, we modify the Wav2vec post-processing as follows. First, we merge the consecutively repeated letters and single quotes. Second, we remove the blank token between English letters and the single quote. In this way, the blank tokens within any word were removed. Last, we combine the remaining blank tokens with the separate tokens, count the number of the blank and separate tokens between words, and use that number of tokens to determine the lengths of pauses between words.

BERT requires the input as transcripts and punctuation marks. Therefore, we convert pauses to the punctuation marks, i.e., periods and commas. Specifically, we design automatic methods to determine the thresholds ϵ_p and ϵ_c, which are used to set sentence-level pauses and in-sentence pauses.

Sentence-level pause. BERT has extensive prior knowledge of sentence-level pauses from large-scale pre-training datasets. Thus, we use BERT to determine the threshold ϵ_p for sentence-level pauses. Specifically, we aim to maximize

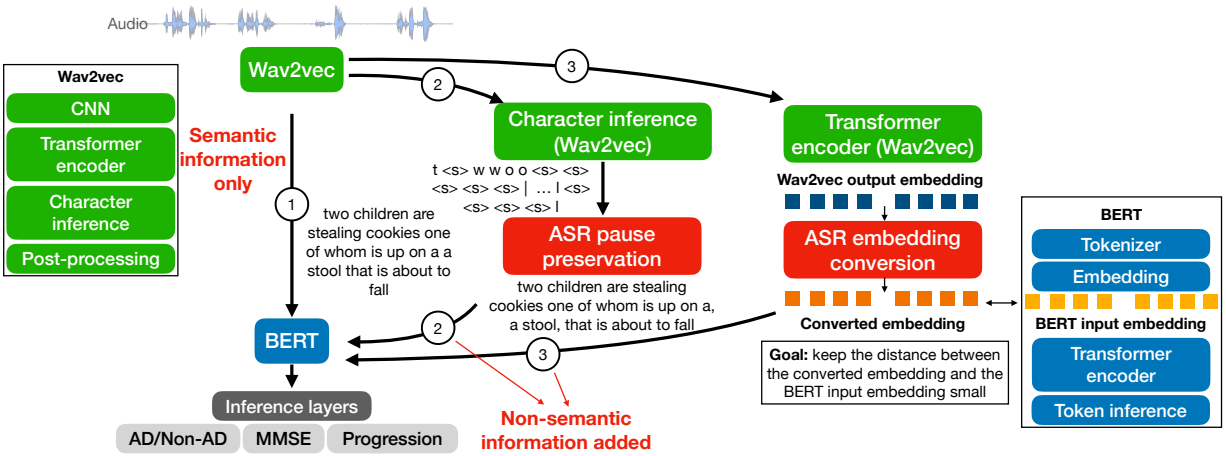

Figure 1: *The basic WavBERT model (Path 1) and extended WavBERT models (Paths 2&3)*

the sample-level cross-entropy difference between the AD and non-AD samples from the training dataset. We consider n_a AD samples $\mathcal{X}_a$ and n_{na} non-AD samples $\mathcal{X}_{na}$. These samples are generated using Wav2vec and do not include any punctuation marks. Using the BERT tokenizer, we obtain k tokens of a transcript as $x_i = \{x_{i,1}, \cdots, x_{i,k}\}$. Given a threshold ϵ_p, we determine the locations of pauses that have lengths $\geq \epsilon_p$, then insert punctuation marks of periods at these locations, and finally obtain t tokens of the transcript as $x_i^{\epsilon_p} = \{x_{i,1}^{\epsilon_p}, x_{i,2}^{\epsilon_p}, \ldots, x_{i,t}^{\epsilon_p}\} \in \mathbb{R}^{t*v}$ $(t \geq k)$. Each token $x_{i,j}^{\epsilon_p} \in \mathbb{R}^v$ is a v-size vector where v is the vocabulary size of BERT. Then we input $x_i^{\epsilon_p}$ to BERT to obtain the corresponding self-supervised token inference after softmax activation $z_i^{\epsilon_p} \in \mathbb{R}^{t*v}$. Then, we remove the tokens of the same indexes of added punctuation marks from $z_i^{\epsilon_p}$ and obtained $\bar{z}_i^{\epsilon_p} \in \mathbb{R}^{k*v}$. We aim to find ϵ_p to maximize the target:

$$\underset{\epsilon_p}{\operatorname{argmax}} | \underset{x_i \in \mathcal{X}_a}{\operatorname{Med}}^{n_a} (\ell_i^{\epsilon_p}) - \underset{x_i \in \mathcal{X}_{na}}{\operatorname{Med}}^{n_{na}} (\ell_i^{\epsilon_p}) | \tag{1}$$

where $\mathrm{Med}()$ is the median function, $\ell_i^{\epsilon_p}$ is the cross-entropy loss of the sample x_i:

$$\ell_i^{\epsilon_p} := \frac{1}{k} \sum_{j=1}^{k} \sum^{v} -log(\bar{z}_{i,j}^{\epsilon_p}) * x_{i,j} \tag{2}$$

In-sentence pause. BERT has no prior knowledge of in-sentence pauses. Thus, we design a statistical method to determine the threshold for locating the in-sentence pauses. Specifically, we measure the lengths of selected pauses in both AD and non-AD samples where we set the maximum length of pauses as ϵ_p. Then, we count the number of pauses with length β as $\pi_{\beta,a}$ and $\pi_{\beta,na}$ for $1 \leq \beta < \epsilon_p$ for AD and non-AD samples, respectively. Finally, we aim to find ϵ_c to maximize the target:

$$\begin{aligned} \underset{\epsilon_c}{\operatorname{argmax}} | \frac{1}{n_a} \sum_{\beta=1}^{\epsilon_c} \pi_{\beta,a} - \frac{1}{n_{na}} \sum_{\beta=1}^{\epsilon_c} \pi_{\beta,na} | + \\ | \frac{1}{n_a} \sum_{\beta=\epsilon_c+1}^{\epsilon_p} \pi_{\beta,a} - \frac{1}{n_{na}} \sum_{\beta=\epsilon_c+1}^{\epsilon_p} \pi_{\beta,na} | \end{aligned} \tag{3}$$

After we determine pauses using ϵ_p and ϵ_c, we insert periods for sentence-level pauses and commas for in-sentence pauses in the ASR transcripts and input the transcripts to BERT.

4.2. ASR embedding conversion

An embedding conversion network converts Wav2vec output embedding to BERT input embedding. The network design faces two challenges: i) the Wav2vec output embedding is at the character-level, while the BERT input embedding is at the sub-word-level; and ii) in order to utilize the pre-training parameters of BERT, we should make the converted embedding close to the BERT input embedding.

For the first challenge, we design a mapping method. We generate an ASR transcript from an audio sample, use the BERT tokenizer to derive sub-word tokens from the ASR transcript, and generate the BERT input embedding for each token. Then, we use the characters of the token to find the corresponding Wav2vec output embedding. Although the characters generated from Wav2vec can be repeated, we can effectively identify any repeated characters using the property of CTC. For example, the Wav2vec output embedding of "t <s> w w o o" corresponds to the BERT input embedding of token "two." Finally, we average the Wav2vec output embedding according to the time dimension to obtain the averaged embedding as e_w and map e_w to the BERT input embedding e_b of the corresponding token. Special tokens such as [CLS] and [SEP] were excluded in this process.

For the second challenge, we design an embedding conversion network with an aim to convert the Wav2vec output embedding e_w to the BERT input embedding e_b. The network consists of two 1D convolutional layers with a layer-norm in between. The 1D convolutional layers use both kernel size and stride set to 1. The number of neurons is equal to the hidden size of BERT. We further design a pre-training process. First, we run the Wav2vec ASR on LibriSpeech [19] to obtain the ASR transcripts and the Wav2vec output embedding, and then we use the BERT tokenizer to obtain the BERT input embedding from the ASR transcripts. We input the Wav2vec output embedding into the embedding conversion network and optimize its outputs with BERT input embedding using l_1 loss. In the training step, after we obtain the converted embedding, we add the embedding of tokens of punctuation marks (from the pause preservation) and special tokens, and finally, input the integrated embedding to BERT.

5. Evaluation

We implemented five models: M_b uses the ASR transcripts as input to BERT, M_{p1} extends M_b by adding sentence-level pauses to ASR transcripts, M_{p2} extends M_b by adding both sentence-level and in-sentence pauses to ASR transcripts, M_e extends M_b with embedding conversion, and M_{e+p2} extends M_b with embedding conversion and sentence-level/in-sentence

Table 1: *Results of classification, regression, and progression tasks over ADReSSo testing dataset. The design of the baseline linguistic model and the definitions of precision, recall, F1, mean F1, accuracy, and RMSE can be found at the baseline paper [10].*

Task	1. Classification (%)						2. Regression	3. Progression (%)					
	Class	Precision	Recall	F1	Mean F1	Accuracy	RMSE	Class	Precision	Recall	F1	Mean F1	Accuracy
Baseline [10]	non-AD	80.00	77.80	78.87	78.87	78.87	5.28	non-decline	83.30	68.20	75.00	66.67	68.75
	AD	77.80	80.00	78.87				decline	50.00	70.00	58.30		
M_b	non-AD	71.79	77.78	74.67	73.16	73.24	4.60	non-decline	64.00	72.73	68.09	39.92	53.13
	AD	75.00	68.57	71.64				decline	14.29	10.00	11.76		
M_{p1}	non-AD	80.00	88.89	84.21	83.02	**83.10**	4.45	non-decline	62.96	77.27	69.39	34.69	53.13
	AD	87.10	77.14	81.82				decline	0	0	0		
M_{p2}	non-AD	77.50	86.11	81.58	80.19	80.28	**4.44**	non-decline	64.29	81.82	72.00	36.00	56.25
	AD	83.87	74.29	78.79				decline	0	0	0		
M_e	non-AD	78.95	83.33	81.08	80.25	80.28	4.46	non-decline	79.17	86.36	82.61	69.08	75.00
	AD	81.82	77.14	79.41				decline	62.50	50.00	55.56		
M_{e+p2}	non-AD	77.78	77.78	77.78	77.46	77.46	4.47	non-decline	81.82	81.82	81.82	**70.91**	75.00
	AD	77.14	77.14	77.14				decline	60.00	60.00	60.00		

pauses. We report the results of the five models in Table 1.

5.1. Implementation and training strategy

We trained the five models with the ADReSSo training dataset and reported the performance of five models over the provided testing dataset. Considering the random states of the models and the limited size of the dataset, we trained each model for 10 rounds and submitted the average results of the 10 rounds. For the classification and progression tasks, we averaged the probabilities from the softmax activation. For the regression task, we averaged the output of MMSE scores. For the classification task, the corresponding models output the non-AD class only if its probability is ≥ 0.5. For the progression task, we implemented classification models for the progression task by treating decline samples as AD samples and non-decline as non-AD. Considering the unbalanced classes of the training dataset, the corresponding models output non-decline class only if its probability is ≥ 0.79, based on the class ratio of the training dataset.

We implemented the five models with PyTorch[1], employing the "bert-base-uncased" and "wav2vec-vox-960h-pl" settings. We filtered out one ASR transcript that has < 20 words in the progression training dataset, which could be caused by either the failure of ASR or inaudible samples. In training, we unfroze all BERT layers and inference layers while freezing the embedding conversion network. We used batch size 8 and learning rate 10^{-6} with the Adam optimizer [23]. We used the cross-entropy loss for the first and third tasks, and we used the mean squared error for the second task. We trained our models with a maximum of 2000 epochs and stopped the training if the loss is smaller than 10^{-6}. Besides, we used a similar setting as above for the pre-training of the embedding conversion network with LibriSpeech [19], but changed the learning rate to 5×10^{-5} and the maximum number of epoch to 100. One-round training using the ADReSSo training dataset took less than 6 hours with one V100 GPU, and one-round pre-training of the embedding conversion network took less than a day with six V100 GPUs.

5.2. Experimental results on testing dataset

The Wav2vec models outperformed the baseline model in all three tasks. Our observations are as follows:

Classification. As shown in Table 1, the basic WavBERT M_b achieved an accuracy of 73.24%. With non-semantic information added into the analysis, M_{p1} and M_e achieved 83.10% and 80.28%, respectively. These accuracy improvements confirmed that the effectiveness of our models, which utilized pause preservation and embedding conversion for non-semantic information. However, the results of M_{p2} and M_{e+p2} exploiting in-sentence pauses were worse compared to the M_{p1} and M_e. We consider that the in-sentence pauses produced a negative impact because the in-sentence pauses were learned from the limited training datasets, which may lead to overfitting. In comparison, the sentence-level pauses were automatically derived using BERT, which provided a positive impact.

Regression. The basic WavBERT M_b produced an RMSE score of 4.60, lower than 5.28 of the baseline model. All extended WavBERT outperformed the basic WavBERT. Specifically, M_{p1} with sentence-level pauses produced an RMSE score of 4.45, M_{p2} with sentence-level/in-sentence pauses further lowered the RMSE score to 4.44, and M_e with embedding conversion produced an RMSE score of 4.47. The performance improvements confirmed that both pause preservation and embedding conversion produced a positive impact. Lastly, M_{e+p2} produced a slightly larger RMSE score, which may have resulted from the limited training dataset and overfitting problem.

Progression. M_b, M_{p1} and M_{p2} resulted in poor performance. By checking the ASR transcripts, we found that the transcripts have a word-misspelling problem for two reasons. First, Wav2vec is a character-level model, and thus the transcripts may have added or missed characters of words. Second, the progression dataset was collected from a category fluency task, significantly different from the training dataset of Wav2vec ASR, thus downgrading the ASR performance. However, M_e and M_{e+p2} with embedding conversion, achieved mean F1 scores 69.08%, 70.91%, outperforming 66.67% of the baseline. We considered that the embedding conversion network effectively mitigated the word-misspelling problem by inputting embedding, not misspelled transcripts, to BERT.

6. Conclusions

We propose WavBERT models by integrating Wav2vec ASR with BERT for an automatic process of dementia detection. While the basic Wav2BERT used ASR transcripts and focused on semantic information, the extended Wav2BERT exploits non-semantic information with a pause preservation module and an embedding conversion network. Our experimental results confirmed that the extended WavBERT models outperformed the baseline linguistic model. Our future goal includes exploring the transformer encoder of BERT for pre-training the embedding conversion network.

7. Acknowledgements

This research is funded by the US National Institutes of Health National Institute on Aging, under grant No. 1R01AG067416.

[1]Codes are available at https://github.com/billzyx/WavBERT

8. References

[1] M. D. Lezak, D. B. Howieson, D. W. Loring, J. S. Fischer *et al.*, *Neuropsychological assessment*. Oxford University Press, USA, 2004.

[2] E. E. Bron, M. Smits, W. M. Van Der Flier, H. Vrenken, F. Barkhof, P. Scheltens, J. M. Papma, R. M. Steketee, C. M. Orellana, R. Meijboom *et al.*, "Standardized evaluation of algorithms for computer-aided diagnosis of dementia based on structural MRI: the CAD Dementia challenge," *NeuroImage*, vol. 111, pp. 562–579, 2015.

[3] S. Luz, F. Haider, S. de la Fuente, D. Fromm, and B. MacWhinney, "Alzheimer's Dementia Recognition through Spontaneous Speech: The ADReSS Challenge," *arXiv preprint arXiv:2004.06833*, 2020.

[4] N. Cummins, Y. Pan, Z. Ren, J. Fritsch, V. S. Nallanthighal, H. Christensen, D. Blackburn, B. W. Schuller, M. Magimai-Doss, H. Strik *et al.*, "A comparison of acoustic and linguistics methodologies for Alzheimer's dementia recognition," in *Interspeech 2020*. ISCA-International Speech Communication Association, 2020, pp. 2182–2186.

[5] J. Koo, J. H. Lee, J. Pyo, Y. Jo, and K. Lee, "Exploiting Multi-Modal Features From Pre-trained Networks for Alzheimer's Dementia Recognition," *arXiv preprint arXiv:2009.04070*, 2020.

[6] E. Edwards, C. Dognin, B. Bollepalli, M. Singh, and V. Analytics, "Multiscale System for Alzheimer's Dementia Recognition through Spontaneous Speech," *Proc. Interspeech 2020*, pp. 2197–2201, 2020.

[7] R. Pappagari, J. Cho, L. Moro-Velazquez, and N. Dehak, "Using state of the art speaker recognition and natural language processing technologies to detect alzheimer's disease and assess its severity," *Proc. Interspeech 2020*, pp. 2177–2181, 2020.

[8] Y. Zhu and X. Liang, "Exploiting fully convolutional network and visualization techniques on spontaneous speech for dementia detection," *arXiv preprint arXiv:2008.07052*, 2020.

[9] Y. Zhu, X. Liang, J. A. Batsis, and R. M. Roth, "Exploring Deep Transfer Learning Techniques for Alzheimer's Dementia Detection," *Frontiers in Computer Science*, vol. 3, p. 22, 2021.

[10] S. Luz, F. Haider, S. de la Fuente, D. Fromm, and B. MacWhinney, "Detecting cognitive decline using speech only: The adresso challenge," *medRxiv*, 2021.

[11] K. C. Fraser, F. Rudzicz, N. Graham, and E. Rochon, "Automatic speech recognition in the diagnosis of primary progressive aphasia," in *Proceedings of the fourth workshop on speech and language processing for assistive technologies*, 2013, pp. 47–54.

[12] L. Tóth, G. Gosztolya, V. Vincze, I. Hoffmann, G. Szatlóczki, E. Biró, F. Zsura, M. Pákáski, and J. Kálmán, "Automatic detection of mild cognitive impairment from spontaneous speech using ASR," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[13] J. Yuan, X. Cai, Y. Bian, Z. Ye, and K. Church, "Pauses for Detection of Alzheimer's Disease," *Frontiers in Computer Science*, vol. 2, p. 57, 2020.

[14] F. Haider, S. De La Fuente, and S. Luz, "An Assessment of Paralinguistic Acoustic Features for Detection of Alzheimer's Dementia in Spontaneous Speech," *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 2, pp. 272–281, 2019.

[15] K. C. Fraser, J. A. Meltzer, and F. Rudzicz, "Linguistic features identify Alzheimer's disease in narrative speech," *Journal of Alzheimer's Disease*, vol. 49, no. 2, pp. 407–422, 2016.

[16] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[17] A. Balagopalan, B. Eyre, F. Rudzicz, and J. Novikova, "To BERT or Not To BERT: Comparing Speech and Language-based Approaches for Alzheimer's Disease Detection," *arXiv preprint arXiv:2008.01551*, 2020.

[18] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "Wav2vec 2.0: A framework for self-supervised learning of speech representations," *arXiv preprint arXiv:2006.11477*, 2020.

[19] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[20] J. T. Becker, F. Boller, O. L. Lopez, J. Saxton, and K. L. McGonigle, "The natural history of alzheimer's disease: description of study cohort and accuracy of diagnosis," *Archives of Neurology*, vol. 51, no. 6, pp. 585–594, 1994.

[21] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.