Deep learning for fusing speech and text for detection of Alzheimer's Disease

Name: Tan Jun Xian

Abstract

Alzheimer's Disease (AD) is one of the most common forms of dementia which occurs mainly in elderlies. Extensive research has been done to find a cure. However, early intervention is just as important. Impaired speech, which occurs in the early stages of AD, could be used as a biomarker for early detection of AD. Impaired speech could provide useful information, such as audio and text features, to be used for detection of AD.

In this project, we will be using speech and text separately to detect AD. Then, speech and text will be combined to detect AD. Audio features like extended Geneva Minimalistic Acoustic Parameter Set (eGeMAPS) and text features like document embeddings, word embeddings, will be extracted. These features will then be used for detection of AD, using different machine learning and deep learning methods, such as Decision Tree, Random Forest, Support Vector Machine, Logistic Regression, Neural Networks.

To obtain better results, the best models achieved from using speech and text will be combined. The fusion of speech and text will be done by using fusion mechanisms such as Concatenation and Bilinear Pooling. Our results indicated that by using the bilinear pooling mechanism, it produced better results compared to the concatenation mechanism.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1. Background

Alzheimer's Disease (AD) is a common form of dementia that affects the patient's cognitive skills, memories, and motor skills. This in turn will affect their ability to perform the most basic tasks used in daily life [1]. This disease will worsen over time, and there have not been any proven cure for AD [2], which is why early intervention is important. Early intervention of AD can allow treatments to start earlier, which can slow down AD.

In the early stages of AD, speech is impaired and could be used as a biomarker for early detection of AD [3]. Current methods used by doctors to diagnose AD includes a combination of cognitive tests, brain imaging and structured interviews which can be time-consuming and expensive [4]. Therefore, by using speech and text to detect AD, it can be more efficient. Impaired speech could provide useful information, such as audio and text features, which could be used for detection of AD.

1.2. Objectives

The objective of this project is to develop a predictive model by using fusion mechanisms to combine audio and text features. In order to detect AD, speech will be converted into text first, then, predictive models will be developed for speech and text individually for AD classification. Finally, fusion mechanisms such as bilinear pooling and concatenation will be used to combine audio and text features.

1.3. Summary of Work Done

For speech, audio data pre-processing and feature extraction was done. The extended Geneva Minimalistic Acoustic Parameter Set (eGeMAPS) features was extracted. The features will be fed as input to machine learning classifiers and a neural network.

As for text, text transcripts were generated using Hidden Unit Bi-directional Encoder Representations from Transformers (HuBERT). Embeddings were extracted using Doc2Vec, these embeddings were used as input to machine learning classifiers and a neural network. BERT-base was also used to extract embeddings. The BERT embeddings were used as input to a neural network.

We then picked the best model from speech and text to be used for fusion.

For fusion, fusion mechanisms such as Bilinear Pooling and Concatenation was used. Then we compared the performance using these methods.

1.4. Summary of Results

For speech, the experiments showed that the eGeMAPS features with a neural network classifier performed better than machine learning classifiers.

For text, the Doc2Vec embeddings with a neural network classifier also performed better than other machine learning classifiers. However, the BERT embeddings with a neural network classifier performed the best.

For fusion, the bilinear pooling method performed the best out of all the methods used.

2. Background

Previous work [6] performed AD classification using speech and text separately, with early and late fusion of audio and text features.

The authors used different audio features, namely MFCC, GeMAPS, eGeMAPS, INTERSPEECH 2016 Computational Paralinguistics Challenge Feature Set (ComParE-2016) [18], and INTERSPEECH 2010 Paralinguistics Challenge Feature Set (IS10-Paraling) [19]. For MFCC, the first 13 MFCC bands from 0 to 12, with 13 delta MFCCs, and 13 delta-delta MFCCs were extracted. The GeMAPS features contains 18 low level descriptors (LLD), which includes frequency, energy, and spectral parameters such as jitter, pitch, loudness. The eGeMAPS features are an extension of GeMAPS, which has an addition of 7 LLDs of dynamic and cepstral parameters. The ComParE feature set contains 6373 static features which were computed over LLD contours. IS10-Paraling feature set has 38 LLDs with 38 delta coefficients, with 21 functionals being applied over these LLDs.

As for the text features, the authors have used DeepSpech from Mozilla, and Google Cloud-based Speech-to-Text to perform automatic speech recognition (ASR) of the audio recordings to generate text transcripts. It was observed that the transcripts generated by Google Cloud ASR was better because it produced better results in their experiments.

To extract text features, Linguistic Inquiry and Word Count (LIWC), and BERT was used. LIWC can be used for text analysis, and it has been used for other tasks such as twitter analysis. For each transcript, 64-dimensional vectors were extracted using LIWC.

The authors used the pretrained BERT base uncased model from the Huggingface Transformers library. Embeddings were computed over each word ($BERT_{word}$) and then aggregated by pooling functions such as minimum, maximum, average, and standard deviation. Embeddings were also calculated for each transcript ($BERT_{sent}$), which produced a 768-dimensional vector.

In their experiments, Leave One Subject Out (LOSO) cross validation was used with Logistic Regression classifier, with grid search to optimize two hyper-parameters, regularization strength between 5e-5 and 1e2, and penalty was chosen from L1 and L2 . The best results obtained for AD classification using speech only was by using eGeMAPS features. For text, it was by using Google Cloud's ASR with $BERT_{word}$ features.

To combine audio and text features, the authors applied a straightforward early fusion by concatenating features into a single vector. Late fusion (decision level) was also applied by using three different rules, (i) majority voting of predicted class labels; (ii) average fusion of predicted class probabilities; (iii) weighted average fusion of class probabilities. By using $BERT_{word}$ and eGeMAPS features with early fusion, they were able to achieve an accuracy of 75.90, and with average late fusion, an accuracy of 77.71.

Another previous work [21] also performed AD classification using speech and text separately. Audio features, eGeMAPS was also used, as for the text features, the authors have used Google Cloud ASR. The transcripts were converted into CHAT format, lexical and morphological descriptions were assigned to all the words using an automated MOR function. Additional commands, EVAL and FREQ was used to create a composite profile of 34 measures, to compute the Moving Average Type Token Ratio [22]. LOSO cross validation was done with 5 different machine learning classifiers, such as decision trees, K-nearest neighbour, linear discriminant analysis, tree bagger, and support vector machine. The best results achieved for audio features were 78.92 by using decision trees, and 75.90 for text features by using tree bagger.

Doc2Vec [12] is an unsupervised algorithm which can learn feature representations for sentences or documents. The embeddings can represent each document to be used for other natural language tasks, such as document classification. In a related work [23], doc2vec was also used in a similar task of AD detection, where the embeddings were extracted to be used in a support vector machine classifier.

Bilinear pooling has been used mainly in computer vision tasks such as fine-grained recognition, scene categorization, texture recognition, and visual question answering tasks [17]. It takes the outer product of features from two Convolutional Neural Networks (CNN), to generate localized features. This has proven to be effective in boosting the performance of CNNs. Another work [24] experimented with a similar method called factorized bilinear pooling to fuse audio and visual features for an Audio-Video Emotion Recognition task to boost the performance of their experiments. But we will be using just bilinear pooling in our experiments.

3. Methodology

3.1. Audio Features

For speech, openSMILE [10] will be used to extract audio features. As AD patients have impaired cognitive skills, we believe that silence in the audio recordings may be useful in detection of AD. Therefore, silence in the audio recordings will not be removed. Audio recordings will also be resampled to 16kHz. All features extracted will be normalized by subtracting the mean and dividing by the standard deviation.

3.1.1. extended Geneva Minimalistic Acoustic Parameter Set (eGeMAPS)

The eGeMAPS features will be extracted, it consists of features such as semitone, loudness, spectral flux, MFCC, jitter, shimmer, frequency, bandwidth, amplitude, alpha ratio, hammarberg index, slope, and other statistical functions. This will generate a total of 88 features, which are computed over the entire audio recording. These features will be used as inputs to machine learning classifiers and a neural network.

3.2. Text Features

For text, we will first be generating text transcripts using HuBERT. All text transcripts have been lowercased. As AD patients tend to have repetitions in their speech [11], these could be useful indicators in detection of AD, therefore no stop words were removed, and no other text pre-processing has been done.

3.2.1. Automatic Speech Recognition (ASR)

To generate text transcripts, the HuBERT [9] model will be used to perform ASR. This is an open-sourced alternative to Google's ASR. However, there will not be any fine-tuning done as no ground truth for the text transcription was provided. The figure below shows the process to generate text transcripts.

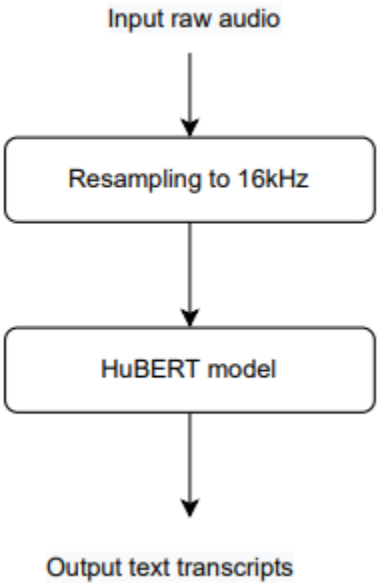

Figure 1 ASR using HuBERT to generate text transcripts

As the HuBERT model was pre-trained on audio files with 16kHz sampling rate, all our audio recordings have been resampled to 16kHz to ensure that the best results are produced. Each audio recording will be transcribed based on speaker identifiers, interviewer (INV) and participant (PAR), where PAR are patients with AD or Non-AD. We will be using only the PAR text transcripts.

3.2.2. Doc2Vec

As some of the text transcripts are long, they could be considered paragraphs or documents. Doc2Vec [12] will be used to generate document embeddings for each text transcript. These document embeddings are a numeric representation of each text transcript. The Doc2Vec has a vector size of 512 to represent each text transcript. The window size parameter, which will determine the size of the context, has been set to 8. The extracted embeddings will be used for different machine learning classifiers and a neural network.

3.2.3. BERT

BERT base uncased, one of the language models for Natural Language Processing [13] will be used to extract embeddings from text transcripts. The BERT architecture is more complex than Doc2Vec as it uses transformers. It has also been proven to perform well in natural language tasks, such as sentiment analysis, text, or sentence classification. The maximum length has been set to 512, and any tokenized text shorter or longer than 512 will be padded or truncated. As BERT uses word-piece tokenization, some information will be lost when the truncation function is applied. There are different ways to extract embeddings using BERT as shown in the figure below. The Second-to-Last hidden layer for each token will be averaged, producing a single 768-dimensional vector. These embeddings will be used as inputs to a neural network classifier.

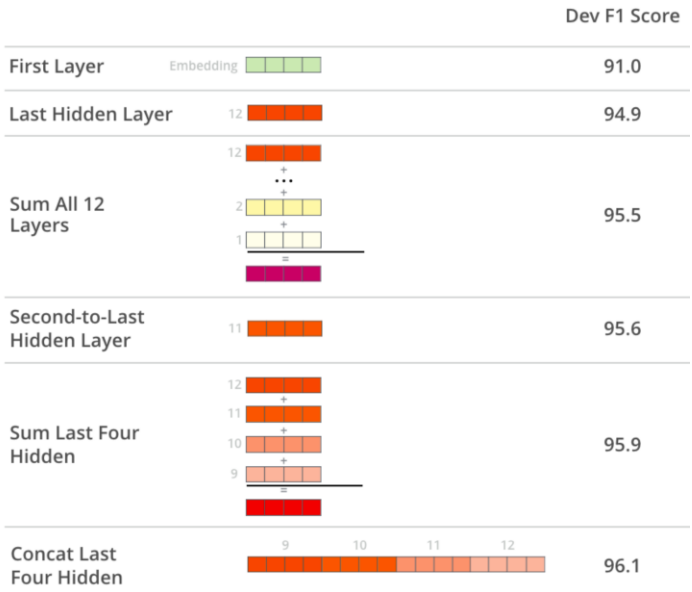

Figure 2 Different strategies to extract embeddings in BERT [20]

3.3. Detecting AD using Speech

AD classification will be done using the extracted eGeMAPS features.

3.3.1. Machine Learning Classifiers

The eGeMAPS features will be used with machine learning classifiers, and a neural network.

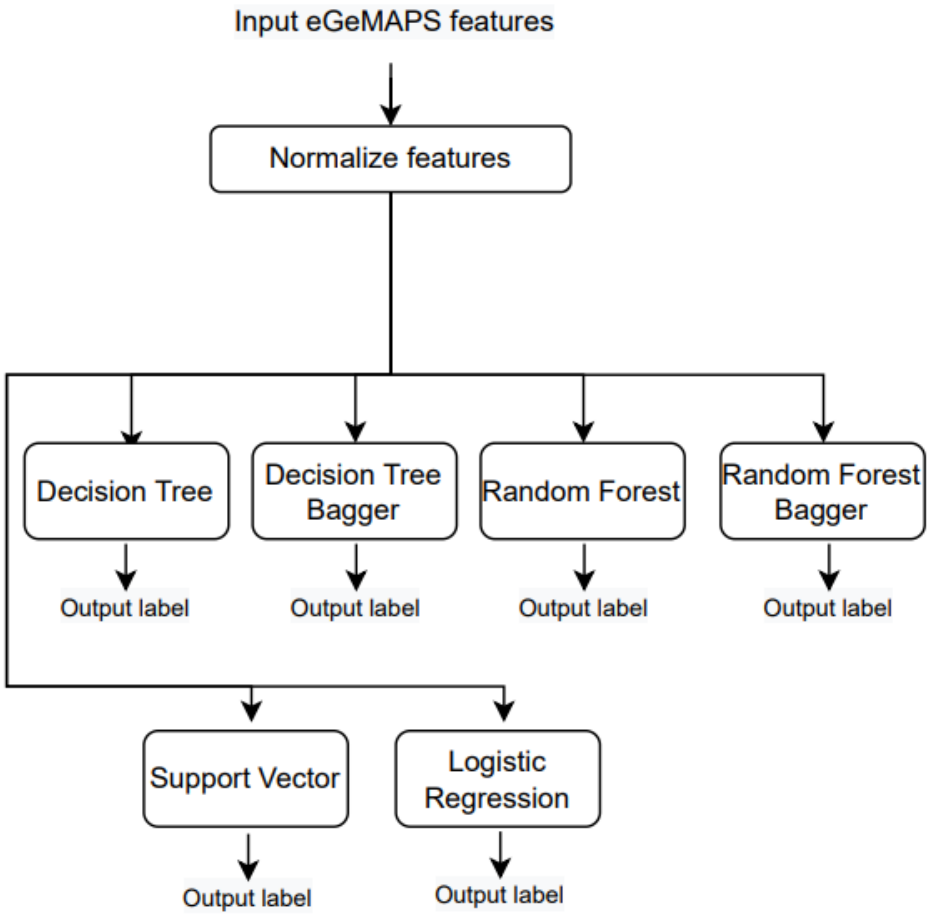

Figure 3 Machine learning classifiers with eGeMAPS features

The machine learning classifiers used are decision trees, decision trees bagger, random forest, random forest bagger, support vector machine, logistic regression, each with their own hyper-parameters tuned using grid search.

3.3.2. Neural Network Architecture

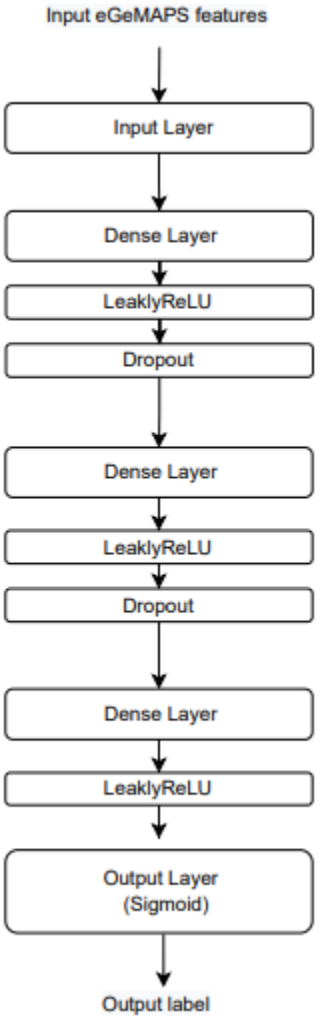

Figure 4 Neural network with eGeMAPS features

The neural network architecture consists of (i) an input layer; (ii) 3 sets of dense layers, with leaky rectified linear unit activation function, and a dropout layer; (iii) an output layer with sigmoid activation function. The hyper-parameters, number of neurons, batch size, dropout rate, learning rates were tuned using grid search.

3.4. Detecting AD using Text

The text transcripts generated using HuBERT will be used for AD classification. Embeddings that were extracted using Doc2Vec and BERT will be fed as inputs to machine learning classifiers and neural networks.

3.4.1. Machine Learning Classifiers

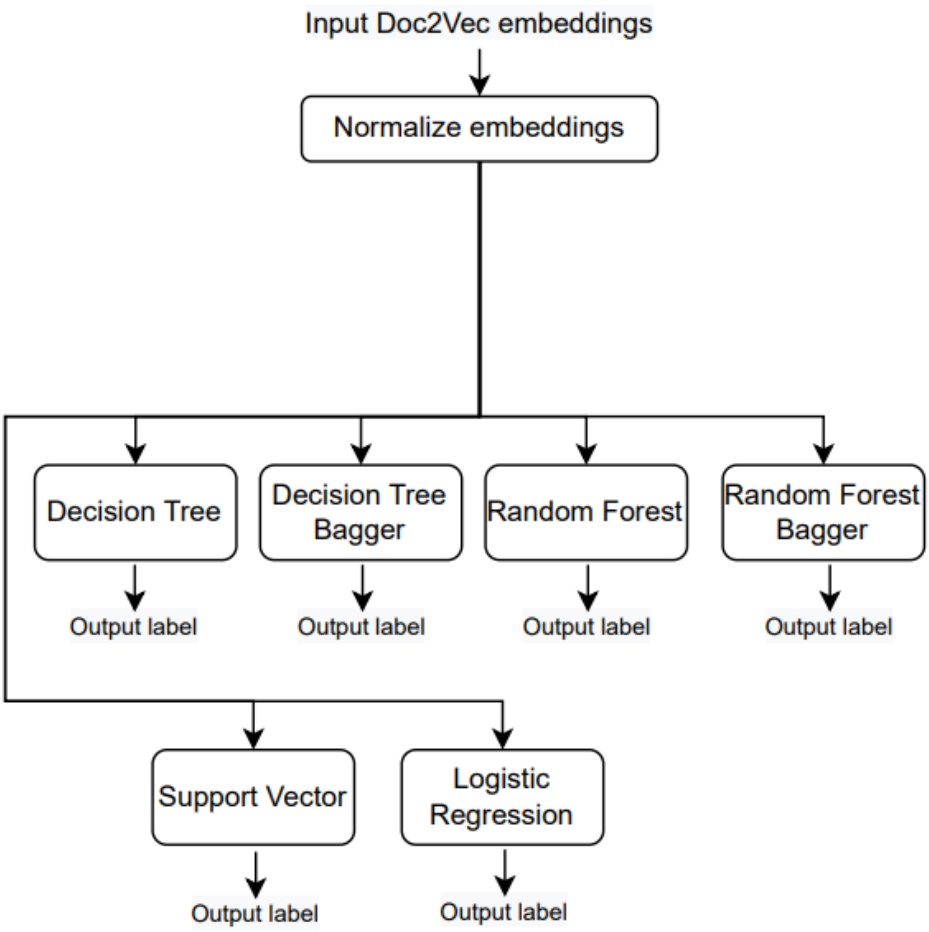

Figure 5 Machine learning classifiers with Doc2Vec embeddings

The Doc2Vec embeddings will be used as input to different machine learning classifiers. Likewise, each classifier's hyper-parameters will be tuned using grid search.

3.4.2. Neural Network Architecture

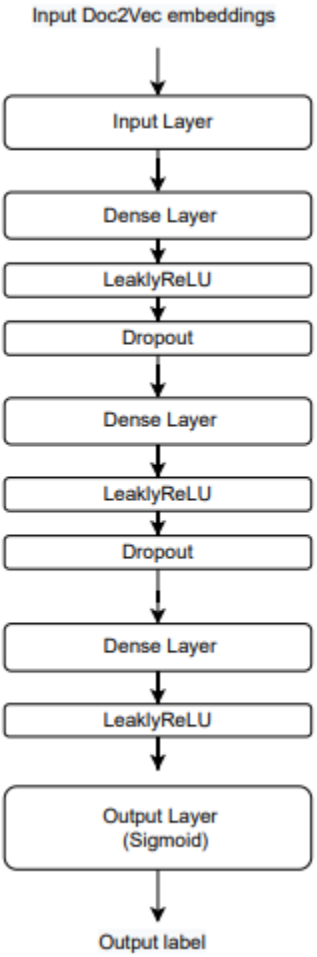

Figure 6 Neural network with Doc2Vec embeddings

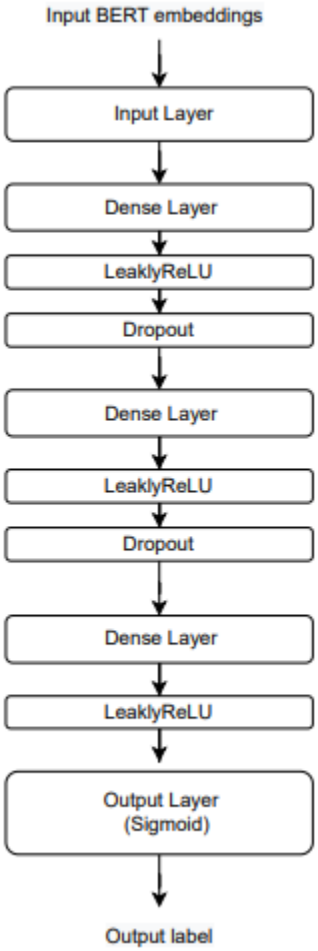

Figure 7 Neural network with BERT embeddings

There will be two similar neural network architectures used, (i) one for Doc2Vec embeddings, (ii) one for BERT embeddings.

The Doc2Vec and BERT embeddings will be fed into two separate 4-layered neural network, which consists of (i) an input layer; (ii) 3 sets of dense layers, with leaky rectified linear unit activation function, and a dropout layer; (iii) an output layer with sigmoid activation function

The hyper-parameters for these networks, such as number of neurons in the dense layer, dropout rate, learning rate, batch size, will be tuned using grid search.

3.5. Fusion Mechanisms

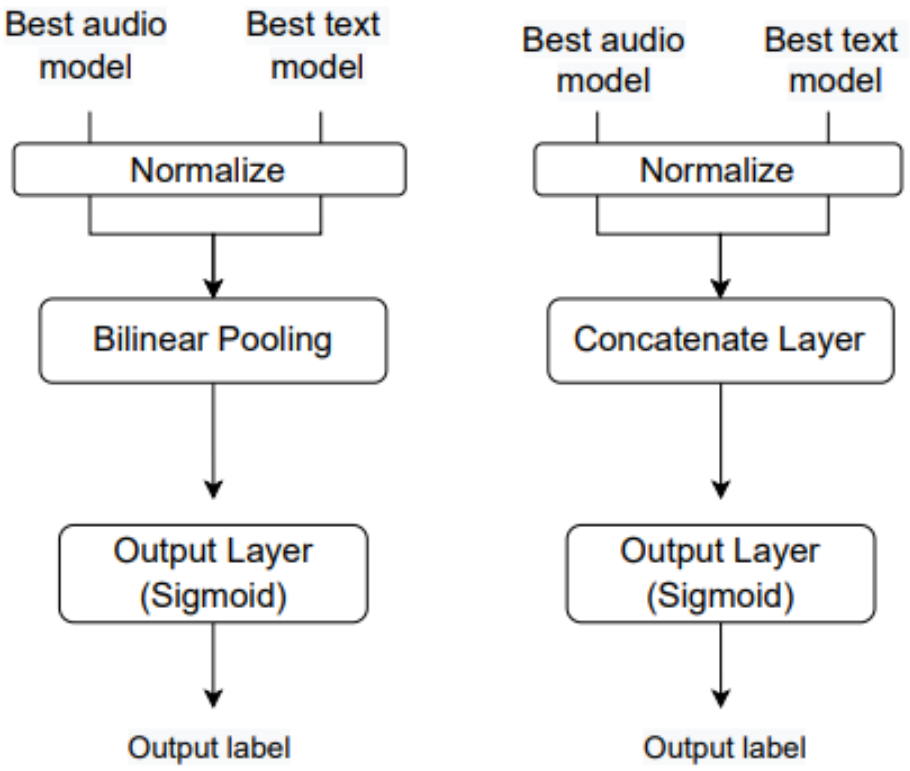

Figure 8 Architecture of fusion mechanisms

There are two different fusion mechanism which was applied in our experiments: (i) Bilinear Pooling [16]; (ii) Concatenation, both with different ways of combining audio and text features. The best models from speech and text which achieved the best accuracies, were neural networks, therefore we will use them and apply these fusion mechanisms.

3.5.1. Bilinear Pooling

Figure 9 Outer product of features

The bilinear pooling layer is trainable, and it takes the outer product of two vectors [16].

The two vectors will be the audio and text features. These features will be taken from the best model saved when performing AD classification. The outputs of the best audio and text models will then be connected to the bilinear pooling layer.

Let $U, U \in R^{d}$ be the matrix for audio features and $V, V \in R^{d}$ be the matrix for text features. The weights W, $W \in R^{n \times d}$ will be randomly initialized with a normal distribution. d represents the dimensions for the audio and text feature. n represents the output dimensions of the bilinear operation.

The outer product of these features will be $Z = W(U \otimes V^{T}) + b, Z \in R^{n}$, where b is for bias.

After the computation of bilinear pooling is done, it will be fed to an output layer with the sigmoid activation function (σ), to be used for classification. The full operation can be denoted as

$$Y = \sigma(\ (W(U \otimes V^{T}) + b))$$

where Y represents the output label.

3.5.2. Concatenation

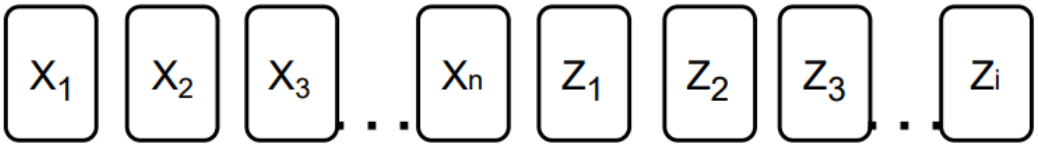

Figure 10 Concatenation of features

For the concatenation layer, output from the audio and text models will be connected to the concatenation layer. The features will then be joined together. Let X be the audio features with n dimension, and Z be the text features with i dimension, the concatenated features will be

$$[X_1 X_2 X_3 \dots X_n Z_1 Z_2 Z_3 \dots Z_i]$$

The combined features will be fed to the output layer with a sigmoid activation function (σ). The hyper-parameters will also be tuned using grid search. The full operation can be denoted as

$$Y = \sigma([X_1 X_2 X_3 \dots X_n Z_1 Z_2 Z_3 \dots Z_i]),$$

where Y is the output label.

4. Experiment and Results

In our experiments, 10-fold cross validation was done on the ADReSSo dataset [5]. Grid search was used to tune the hyper-parameters for each machine learning classifier and neural network model. The dataset consists of a set of audio recordings of picture descriptions, given by AD and Non-AD patients, who were asked to describe the Cookie Theft picture taken from the Boston Diagnostic Aphasia Examination [14]. The dataset has been matched for age and gender to reduce the risk of bias when performing AD classification. There is a total of 166 audio samples.

For the machine learning classifiers, the decision tree and decision tree bagger were trained with maximum depth ranging from 1 to 10. Random forest and random forest bagger were trained with number of estimators of 50, and the maximum depth ranging from 1 to 10. The support vector classifier was trained with the C value ranging from 0.5 to 1.5, in intervals of 0.1, and its kernel chosen from linear, rbf. The logistic regression was trained with its C value ranging from 0.5 to 1.5 in intervals of 0.1.

For the 4-layer neural networks, grid search was also used to tune the hyper-parameters. The number of neurons were chosen from 16, 32. Dropout rate values were chosen from 0.3, 0.5, and batch size was chosen from 1, 2, 4. We set the default value of alpha in leaky rectified linear unit to 0.2. The models were trained for 50 epochs. The text model was trained with a learning rate of 1e-4, while the audio model was trained with a learning rate of 1e-3.

The bilinear model was trained with batch size of 8 for 50 epochs, with learning rate of 1e-4. The concatenate model was trained with batch size of 1 for 50 epochs, with learning rate of 1e-5.

From the table below, the support vector performed better than the other machine learning classifiers. But when using a 4-layer neural network, it performed the best, slightly better than the support vector classifier. The neural network, decision tree, random forest, took the longest time to train, followed by the support vector and logistic regression which took the fastest to train.

Table 1 Results in mean and std for Detection of AD using Speech

	Features	Model	Accuracy
Audio	eGeMAPS	Decision Tree	63.23 ± 10.72
		Decision Tree Bagger	68.12 ± 8.99
		Random Forest	68.71 ± 8.29
		Random Forest Bagger	65.66 ± 13.4
		Support Vector	71.10 ± 8.77
		Logistic Regression	70.58 ± 9.75
		4 Layer Neural Network	**72.42 ± 10.05**

For text, the logistic regression performed the best compared to the other machine learning classifiers. But the Doc2Vec features with a neural network performed better than the other machine learning classifiers. However, the performance is only slightly better than logistic regression.

When using the BERT features with a neural network, the accuracy improved. As BERT uses a much more advanced architecture which includes using transformers, this allows better features to be extracted, which produced better results. Likewise, neural networks took the longest to train, followed by the decision tree, random forest, with the support vector and logistic regression taking the shortest time.

Table 2 Results in mean and std for Detection of AD using Text

	Features	Model	Accuracy
Text	Doc2Vec	Decision Tree	61.47 ± 15.00
		Decision Tree Bagger	64.48 ± 9.15
		Random Forest	65.00 ± 12.42
		Random Forest Bagger	64.37 ± 10.80
		Support Vector	67.97 ± 15.43
		Logistic Regression	69.22 ± 15.45
		4 Layer Neural Network	70.95 ± 8.41
	$BERT_{base}$	4 Layer Neural Network	**74.52 ± 12.47**

When comparing the models between audio and text, the text model, BERT with a neural network did better than the best audio model.

By using the concatenate method for fusion, the accuracy was not as good. The concatenation method simply joins features together, which could explain the poor performance.

We will compare the results with the previous work [6] which used similar feature sets. Overall, we see that the fusion methods achieved better results compared to using audio or text alone.

The average and concatenate fusion method from the previous work performed better than our concatenation method.

However, when using bilinear pooling, it produced the best results overall. But we must consider that we are using 10-Fold cross validation compared to the previous work's LOSO cross validation. Using LOSO will take up much more time as there would be a total of 166 splits, resulting in thousands of iterations, especially when doing grid search for hyper-parameters tuning, which is why LOSO was not implemented here.

Table 3 Results in mean for Detection of AD with fusion of speech and text, compared to other methods

	Method	Features	Accuracy
Late Fusion	Bilinear Pooling	BERT + eGeMAPS	**83.30 ± 10.40**
	Concatenate	BERT + eGeMAPS	75.55 ± 22.21
Audio	-	eGeMAPS	72.42 ± 10.05
Text	-	BERT	74.52 ± 12.47
Late Fusion	Average [6]	BERT + eGeMAPS	77.71
Early Fusion	Concatenate [6]	BERT + eGeMAPS	75.90

5. Conclusion & Future work

Overall, the fusion mechanisms were able to produce better results, compared to when using audio or text models alone. Bilinear pooling has shown that it performs better than the simple method of concatenating features.

The experiments have also proved that most deep learning methods performed better than classic machine learning methods.

The type of features used would also determine the performance, as we have seen that BERT features gave better results compared to Doc2Vec.

In our experiments, only BERT base model was used to extract the text features. There is also a BERT large model which could be used to extract text features, which could possibly produce better results. However, the BERT model has a limitation of maximum number of tokens, which is 512 and information might be lost when truncating the sequence of tokens.

The type of ASR used is also important, as the transcripts are generated using a neural model, it is not as accurate as manual transcribing. However, we did not perform manual transcribing for the audio recordings, as it was time consuming. Using manual transcripts or exploring other open-sourced ASR, such as DeepSpeech could be used.

References

[1] Desai, A.K., and Grossberg, G.T.: 'Diagnosis and treatment of Alzheimer's disease', Neurology, 2005, 64, (12 suppl 3), pp. S34.

[2] "Treatments," Alzheimer's Disease and Dementia. [Online]. Available: https://www.alz.org/alzheimers-dementia/treatments, [Accessed Jan. 22, 2022].

[3] Robin, J., Xu, M., Kaufman, L.D., and Simpson, W.: 'Using Digital Speech Assessments to Detect Early Signs of Cognitive Impairment', Frontiers in Digital Health, 2021, 3.

[4] Porsteinsson, A.P., Isaacson, R.S., Knox, S., Sabbagh, M.N., and Rubino, I.: 'Diagnosis of Early Alzheimer's Disease: Clinical Practice in 2021', The Journal of Prevention of Alzheimer's Disease, 2021, 8, (3), pp. 371-386

[5] "DementiaBank," TalkBank. [Online]. Available: https://dementia.talkbank.org/access. [Accessed: Feb. 9, 2022].

[6] Chen, J., Ye, J., Tang, F., and Zhou, J.: 'Automatic Detection of Alzheimer's Disease Using Spontaneous Speech Only' (2021. 2021)

[7] Graves, A., Jaitly, N., and Mohamed, A.: 'Hybrid speech recognition with Deep Bidirectional LSTM', in Editor (Ed.): 'Book Hybrid speech recognition with Deep Bidirectional LSTM' (2013, edn.), pp. 273-278

[8] El-Moneim, S.A., Nassar, M.A., Dessouky, M.I., Ismail, N.A., El-Fishawy, A.S., and Abd El-Samie, F.E.: 'Text-independent speaker recognition using LSTM-RNN and speech enhancement', Multimedia Tools and Applications, 2020, 79, (33), pp. 24013-24028

[9] Hsu, W.-N., Bolte, B., Tsai, Y.-H.H., Lakhotia, K., Salakhutdinov, R., and Mohamed, A.: 'Hubert: Self-supervised speech representation learning by masked prediction of hidden units', IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2021, 29, pp. 3451-3460.

[10] "OpenSMILE," openSMILE Documentation. [Online]. Available: https://audeering.github.io/opensmile/. [Accessed: Jan. 22, 2022].

[11] "Repetition," Alzheimer's Disease and Dementia. [Online]. Available: https://www.alz.org/help-support/caregiving/stages-behaviors/repetition, [Accessed Jan. 23, 2022].

[12] Le, Q., and Mikolov, T.: 'Distributed representations of sentences and documents', in Editor (Ed.): 'Book Distributed representations of sentences and documents' (PMLR, 2014, edn.), pp. 1188-1196.

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

[14] Roth, C.: 'Boston Diagnostic Aphasia Examination', in Kreutzer, J.S., DeLuca, J., and Caplan, B. (Eds.): 'Encyclopedia of Clinical Neuropsychology' (Springer New York, 2011), pp. 428-430.

[15] Staudemeyer, R.C., and Morris, E.R.: 'Understanding LSTM--a tutorial into long short-term memory recurrent neural networks', arXiv preprint arXiv:1909.09586, 2019.

[16] Fukui, A., Park, D.H., Yang, D., Rohrbach, A., Darrell, T., and Rohrbach, M.: 'Multimodal compact bilinear pooling for visual question answering and visual grounding', arXiv preprint arXiv:1606.01847, 2016

[17] Lin, T.-Y., RoyChowdhury, A., and Maji, S.: 'Bilinear cnn models for fine-grained visual recognition', in Editor (Ed.): 'Book Bilinear cnn models for fine-grained visual recognition' (2015, edn.), pp. 1449-1457

[18] B. Schuller, S. Steidl, A. Batliner, J. Hirschberg, J. K. Burgoon, A. Baird, A. Elkins, Y. Zhang, E. Coutinho, K. Evanini et al., "The interspeech 2016 computational paralinguistics challenge: Deception, sincerity & native language," in 17TH Annual Conference of the International Speech Communication Association (Interspeech 2016), Vols 1-5, 2016, pp. 2001–2005.

[19] B. Schuller, S. Steidl, A. Batliner, F. Burkhardt, L. Devillers, C. Muller, and S. S. Narayanan, "The interspeech 2010 paralinguistic challenge," in Eleventh Annual Conference of the International Speech Communication Association, 2010.

[20] Alammar, "The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)", *Jalammar.github.io*, 2018. [Online]. Available: http://jalammar.github.io/illustrated-bert/. [Accessed: 17- Mar- 2022].

[21] Luz, S., Haider, F., de la Fuente, S., Fromm, D., and MacWhinney, B.: 'Detecting cognitive decline using speech only: The ADReSSO Challenge', medRxiv, 2021, pp. 2021.2003.2024.21254263.

[22] M. Covington and J. McFall, "Cutting the Gordian Knot: The Moving-Average Type–Token Ratio (MATTR)", *Journal of Quantitative Linguistics*, vol. 17, no. 2, pp. 94-100, 2010.

[23] Martinc, M., and Pollak, S., "Tackling the ADReSS challenge: a multimodal approach to the automated recognition of Alzheimer's dementia," in *Proceedings of Interspeech 2020* (Shanghai), 2020, 2157–2161.

[24] H. Zhou, D. Meng, Y. Zhang, X. Peng, J. Du, K. Wang, et al., "Exploring emotion features and fusion strategies for audio-video emotion recognition", *In 2019 International Conference on Multimodal Interaction*, pp. 562-566, 2019.