8-2024

Unveiling Key Features: A Comparative Study of Machine Learning Models for Alzheimer's Detection

Cailean Bushnell
Utah State University

Recommended Citation

Bushnell, Cailean, "Unveiling Key Features: A Comparative Study of Machine Learning Models for Alzheimer's Detection" (2024). *All Graduate Reports and Creative Projects, Fall 2023 to Present*. 42.
https://digitalcommons.usu.edu/gradreports2023/42

Unveiling Key Features: A Comparative Study of Machine Learning Models for Alzheimer's Detection

By: Cailean Bushnell

A thesis submitted in partial fulfillment

of the requirements for the degree

of

MASTER OF SCIENCE

In

Economics

Approved:

Carly Fox, Ph.D.	Todd Griffith, Ph.D.
Major Professor	Committee Member
Pedram Jahangiry, Ph.D.	UTAH STATE UNIVERSITY
Outside Committee	Logan, Utah
Member	2023

Unveiling Key Features: A Comparative Study of Machine Learning Models for Alzheimer's Detection

Cailean Bushnell

Department of Economics and Finance

Utah State University

Abstract

This thesis rigorously evaluates the application of an array of natural language processing (NLP) techniques and machine learning models to identify linguistic signatures indicative of dementia, as sourced from the DementiaBank Pitt corpus. Utilizing a binary classification paradigm, this study meticulously integrates sophisticated embedding methods—including Doc2Vec, Word2Vec, GloVe, and BERT—with traditional machine learning algorithms such as Random Forest, Multinomial Naïve Bayes, ADA boost, KNN classifier, and Logistic Regression, alongside deep learning architectures like LSTM, Bi-LSTM, and CNN-LSTM. The efficacy of these methodologies is evaluated based on their capacity to differentiate between transcribed speech impacted by dementia and that from control subjects. To enhance interpretability, this research also employs feature importance analysis through LIME, SHAP, permutation importance, and integrated gradients, shedding light on the variables most instrumental in driving model predictions. The results of this comprehensive analysis not only illuminate the robust potential of these combined NLP and machine learning approaches in the context of medical screening but also contribute additional valuable insights to the field of NLP and dementia screening specifically.

Keywords: *dementia detection, machine learning, deep learning, Pitt corpus, feature importance.*

I acknowledge and thank Carly Fox for her guidance and for serving as my thesis committee chairman. I also thank Todd Grifith and Pedram Jahangiry for serving as thesis committee members.

SECTION I: Introduction

Alzheimer's disease (AD), a progressive neurodegenerative disorder that causes atrophy of the brain (the loss of neurons and connections between neurons), is the most common form of dementia, making up 60%-70% of all dementia cases [10]. AD is characterized by a decline in cognitive function and impacts the daily life of a diagnosed individual significantly. It is currently the 7th leading cause of death and a leading cause of disability within the elderly population[11]. It relentlessly robs individuals of their memories, reasoning skills, and ultimately, their independence. Currently, over 55 million individuals are living with AD, with a disproportionate percentage of that 55 million (60%), living in low- and middle-income countries[11].

The economic cost of AD is staggering, with healthcare costs for this disease exceeding those of heart disease and cancer combined in the United States. As populations live longer, the societal impact of AD is projected to be immense. Recent studies predict AD will triple worldwide by 2050[9]. In 2019 the global cost of dementia was estimated to be 1.3 trillion US dollars[11]. A significant portion of that estimated cost is attributed to formal and informal caregiving.

At this point in time, there is no cure for AD. It is one of the only major diseases where once symptoms manifest, medical professionals are unable to reverse or slow down the progression of the disease in any meaningful way. Symptoms include, but are not limited to, memory loss, problems with every-day tasks, difficulty walking and maintaining bodily functions, and reduced language capabilities.

Individuals with mild symptoms can continue to function independently, with minor help from friends, family, or caregivers. They are able to continue to work, drive, and participate in their favorite hobbies. Dementia is a degenerative disease and as patients continue to decline from mild AD to severe AD speech inevitably declines. The table presented in this section delineates the progression of symptoms associated AD across three stages: mild, moderate, and severe.

Mild AD	Moderate AD	Severe AD
• General memory loss • Mood changes • Communication problems • Orientation issues • Misplacing items	• Increased confusion • Difficulty with complex tasks • Change in sleep patterns • Wandering and getting lost • personality and behavioral changes that can lead to aggressiveness • Suspicion • Irritability	• Severe memory loss that may inhibit a patient's ability to communicate coherently • Unable to recognize loved ones • Require full time assistance and personal care for daily activities • Increased susceptibility to infections

As Alzheimer's disease progresses, it leads to neuronal damage and brain atrophy, which impact various cognitive functions including the ability to use and understand language. Changes in speech can be attributed to varying stages of **Memory Loss**: as patients struggle with recalling words or remembering the thread of a conversation. **Executive dysfunction**: the disease affects the frontal lobe of the brain, which is responsible for executive functions like planning, decision-making, and moderating social behavior. **Anxiety and depression:** which are common comorbidities in Alzheimer's patients and can exacerbate communication difficulties. **Reduced Processing Speed**: Neurological impairments slow down cognitive processing speed, impacting the patient's ability to process incoming speech quickly, respond appropriately, and keep up with conversations. **Muscle Weakness**: Alzheimer's can affect motor control, leading to weakened muscles around the mouth and throat, which are essential for articulation. This can make speech less clear and more difficult to understand. **Auditory Processing Issues**: There can also be a decline in the patient's ability to process auditory information correctly, making it challenging to respond appropriately in conversations. **Progression of Aphasia**: This condition, often associated with Alzheimer's and other dementias, specifically affects a person's ability to communicate. Patients may develop expressive aphasia, where they know what they want to say but struggle to express thoughts verbally, or receptive aphasia, where they cannot understand spoken or written language well.

The decline of speech capabilities can be categorized into early, moderate and severe stages. In the early stages of AD linguistic changes typically include slower speech, loss of verbal fluency, difficulties with formal and informal writing, finishing sentences, and recalling words. As a patient transitions into the moderate stage, changes in speech can include conversation that is difficult to comprehend, repeated slurring, stammering, or stuttering, inappropriate use of words or phrases, and trouble forming basic sentences. Finally in the severe stage, complications with speech present itself as the inability to think and speak clearly, conversations that are unconnected to a situation, and the repetition of other peoples' words.

Subtle changes in an individual's language capabilities, such as the use of incorrect words, word comprehension, repetitive speech, verbal fluency, and talking at inappropriate times, commonly occur among AD patients at varying levels of mild cognitive impairment. Increased cognitive impairment is strongly correlated with an increased display of aphasia or the impaired ability to understand and produce coherent speech. Given these factors, researchers are looking for cost effective and minimally invasive ways to detect and treat this disorder. Methods from the field of computation, such NLP, can be used as a tool of analysis for the interpretation and detection of changes in a patient's speech. Recent advances in NLP and deep learning technologies have highlighted the potential to detect dementia from spoken language.

Motivation

The interest of this paper lies primarily in the early screening of AD. Early diagnosis is imperative in order to minimize impact on the quality of a patient's life and manage their symptoms. Alzheimer's disease (AD) presents one of the most pressing challenges in contemporary medical science. With millions of individuals affected worldwide, the urgency to combat this neurodegenerative condition is undeniable. The pathology of Alzheimer's disease often begins decades before the manifestation of clinical symptoms, underscoring the critical need for early detection. Early diagnosis has the potential to significantly enhance the quality of life for patients by allowing for the timely management of symptoms.

This thesis is motivated by the possibilities that machine learning, deep learning, and natural language processing offer for of early Alzheimer's detection. Language processing emerges as a particularly promising area due to the subtle linguistic and cognitive changes that precede more overt symptoms of AD. Despite there being no cure for AD, the capacity to predict its onset early through linguistic markers offers a transformative approach to pre-symptomatic intervention.

In response to this opportunity, the proposed paper conducts a comprehensive analysis aimed at testing and validating the efficiency of various computational models for AD prediction. This research evaluates conventional machine learning models such as Decision Tree, Random Forest, Logistic Regression, ADA boost classifier, and KNN classifier, alongside sequential deep learning models including Long Short-Term Memory networks (LSTM), Bidirectional LSTM (Bi-LSTM), and Convolutional Neural Network-LSTM (CNN-LSTM) hybrids coupled with various word embedding methods such as GLOVE, Word2Vec, Doc2Vec, and BERT input embeddings.

By integrating advanced ML and NLP methodologies, this thesis strives to forge a path toward impactful clinical applications, contributing to the burgeoning field of digital biomarkers for AD. The ultimate goal is to leverage technology not just to predict but also to prepare—transforming how we approach this formidable disease long before its symptomatic onset.

Contributions

This thesis contributes to the interdisciplinary fields of dementia detection and cognitive health through a series of applications and evaluations of machine learning and NLP techniques for the early detection of Alzheimer’s disease. Recognizing the pivotal role of early diagnosis in mitigating the progression of AD, this research leverages linguistic analysis and computational intelligence as a non-invasive screening tool to identify early markers of cognitive decline. The following contributions are highlighted:

- The role of conventional ML models: Logistic Regression, Decision Tree, Random Forest, ADABoost Classifier, KNN Classifier, and Multinomial Naive Bayes classifier was investigated for the early detection of linguistic characteristics of Alzheimer’s patients.

- The performance of sequential deep learning models was investigated: LSTM, Bi-LSTM, and a hybrid of CNN-LSTM for automatic detection of AD.

- Multiple embedding techniques/vector representations were obtained from the speech transcripts and passed to the aforementioned deep learning and machine learning models for evaluation. These techniques include: TF-IDF, pre-trained GloVe embeddings, Word2Vec, Doc2vec, and BERT input embeddings.

- This paper contributes to the existing literature of evaluating the interpretability of machine learning models and deep learning models used in the early detection of Alzheimer's disease through the meticulous application of Local Interpretable Model-agnostic Explanations (LIME), SHapley Additive exPlanations (SHAP), permutation importance and Integrated Gradients. By employing these techniques, this thesis draws valuable insights into localized feature importance, providing a granular analysis of how individual features may contribute to model predictions.

The remainder of this paper is structured to provide a comprehensive exploration of machine learning and deep learning applications in the early detection of AD, systematically presenting the research from foundational theories to practical implications. **Section II** delves into the existing body of knowledge with an extensive literature survey and review of related work, establishing the context and justification for this study. **Section III** discusses the DementiaBank Pitt corpus dataset and details the preprocessing steps undertaken to prepare the data for analysis, highlighting the rigor in data handling and preparation. **Section IV**, the methodology section, outlines the diverse array of machine learning models and NLP techniques employed, including the implementation specifics and the rationale behind the choice of models. **Section V** presents the results and a thorough analysis of the findings, evaluating the performance and feature importance of the models in detecting early linguistic markers of Alzheimer's disease. The thesis concludes with **Section VI**, where the conclusions are drawn, and the implications of the findings are discussed. This final section also outlines potential avenues for further research and the future scope of this study, suggesting directions for continuing advancements in the field.

II. Related Work

Prior work has analyzed the early detection of Alzheimer's Disease using text transcripts. During the past decade or so, research efforts have focused on utilizing multi-modal information (i.e., text and audio) to detect AD using speech and language features, as well as oral transcripts, in order to detect patterns of mild cognitive impairment. A plethora of machine learning and deep learning techniques have been used to detect anomalies or irregularities in the narrative language or speech patterns of patients with varying degrees of severity of AD.

Text-Only Detection Studies

For example, *Accurate Detection of Dementia from Speech Transcripts Using RoBERTa* Model uses DementiaBank's Pitt corpus to run multiple experiments evaluating the prediction accuracy of Hugging Face's RoBERTa model. They ran three experiments evaluating RoBERTa's performance against existing research depending on what preprocessing steps are taken to prepare the linguistic transcripts. The first experiment removed almost all of the coded information found in the .cha files. This included spoken words (words spoken by the participant), filler words (words, sounds, and phrases people use to fill empty spaces of communication), pauses, coughing, and any repeated or redundant speech. The second experiment only removed the filled pauses (*uh, um, er*. These pauses are preceded by the ampersand and hyphen mark.), and finally, in the third experiment both the filled pauses and repeated speech were left in the transcript. Using all three experiments, (Matošević, 2022) evaluated RoBERTa against BERT as a baseline model. RoBERTa outperformed BERT with both 256 and 512 tokens across all three experiments. BERT512 achieved and accuracy of 86.26%, 85.03%, and 86.29% across experiments one, two and three. Similarly, BERT256 achieved 85.29%, 86.42%, and 85.22%. They evaluated two RoBERTa models. One using a maximum length parameter of 512 and the other using a maximum length of 256. Of the two, the model set to 512 performed better across all three experiments. Matošević highlights that their 'models benefit from a larger text span'. The second experiment produced their best trained model, which achieved a 90.16% accuracy. (Matošević, 2022) attribute this to the inclusion of repeated speech in the second experiments transcripts, a common characteristic of dementia. Their model was able to isolate this pattern as one correlated with cognitive decline, and outperform the models that had removed the repeated speech. In the third experiment, surprisingly, the inclusion of filled pauses in the transcripts did not affect the performance of the model, indicating they do not lead to better performance.

The *Comparative study of Deep Classifiers for Early Dementia Detection using Speech Transcripts* used a combination of deep learning models, transformer models, and vector representations to detect the dementia class or control group from the Pitt Corpus transcripts. Their aim was to use a variety of methods for binary classification using deep learning and NLP techniques that had not been previously studied. The Pitt Corpus was cleaned by removing punctuation marks, capitalization, and unwanted annotations. Different vectorization techniques were used to obtain embeddings, such as, doc2vec, word2vec, GloVe, BERT, RoBERTa, and ALBERT. The embeddings were then fed to a series of deep learning models consisting of LSTM, BiLSTM, and GRU. Their highest performing accuracy was achieved by the BiLSTM model with BERT embeddings at 81%. The second highest performing model was GRU with RoBERTa embeddings at 80%. The highest precision obtained was 96% by the LSTM model with BERT embeddings[2].

Multi-Modal Detection Studies

Multiple research papers have utilized both the audio files and text transcripts for dementia detection. One such work is *Machine learning of transcripts and audio recordings of spontaneous speech for diagnosis of Alzheimer's disease.* Their objective was to evaluate the performance of manual transcripts vs automatic ones, noised vs denoised recordings, and the different methods of speech recognition models. This work explored machine learning methods to detect AD using Dementia Bank's Pitt corpus by evaluating three different approaches. It utilized both the speech and text data available. The original linguistic transcripts were evaluated alongside automatically generated transcripts based on denoised voice recordings and generated transcripts that had not been denoised. Transcript generation was achieved using the Python library *noisereduce*. Transcribing the audio into text data was done by CMUSpinx pocketsphinx, a program that reads audio from standard input files and attempts to recognize speech in it, and MozillaDeepSpeech, an open-source speech-to-text engine. Transformer model BERT was used to create feature vectors with contextual representation of the transcripts. A neural network model was constructed to perform the classification analysis. This process utilized a 10-fold cross validation and was repeated 20 times. The average results were reported. (Liu, 2021) found features extracted from denoised MozillaDeepSpeech transcripts performed best overall with a

92.72% accuracy. Overall performance between raw and denoised speech recordings was incongruous. Results were dependent on which speech recognition software was used. Features from the pocketsphinx transcripts with noisy recording outperformed the denoised ones, which is opposite relationship observed with the MozillaDeepSpeech transcripts[3].

Computational Intelligent Models for Alzheimer's Prediction Using Audio Transcript Data like (Nambiar, 2022), performs a comprehensive analysis of machine learning (Decision Tree, and Random Forest), deep learning (LSTM, Bi-LSTM, and CNN-LSTM), and transfer learning models (BERT and XLNet) for AD prediction. DementiaBank's Pitt Corpus was also utilized for this study. Data preprocessing steps included tokenization, case correction, stemming, and lemmatization (punctuation, spaces, capitalization, symbols, stop words, and non-ascii characters were removed from the original transcripts).

The statistical method known as term frequency-inverse document frequency (TF-IDF) was then performed to generate vectors. TF-IDF gives a higher weight to words that appear more frequently within a document. The TF-IDF formula is:

$$TF(t,d) = \frac{\text{number of times } t \text{ appears in } d}{\text{total number of terms in } d}$$

$$IDF(t) = \log \frac{N}{1+df}$$

$$TF - IDF(t,d) = TF(t,d) \cdot IDF(t)$$

It is particularly useful at helping detect semantic relevance and identifying which words are important and which words are not within a document. The TF-IDF parameters were as follows: Vector dimensions = 316, max df = 0.9, min df = 5. From the embeddings, vectors were calculated, and an output embedding vector for each tokenized vector was created and ultimately used as an embedding layer for the deep neural networks evaluated. To fine tune hyperparameters for the models being evaluated (Khan, 2022) utilized the simplest algorithm for tuning, the grid search technique. Model performance was evaluated using the metrics: testing accuracy, validation accuracy, ROC curve, and F1-score. Out of the machine learning models, SVM performed the best with an accuracy rate of 85%. Among the deep learning models the CNN-LSTM hybrid achieved a high classification performance with a 90% accuracy. Both transfer learning models, BERT and XLNet outperformed the deep learning and machine learning models at 92% and 93% accuracy rates, respectively. The AUC values for XLNet and BERT were both 97%. Further showing that the transfer learning models outperformed their machine learning and deep learning counterparts.

Overall, BERT obtained the best testing accuracy of 93% and a validation accuracy of 95% when fed the vectors generated by TF-IDF and the model utilizing the fine-tuned parameters. Moreover, these findings are unsurprising given the nature of this classification particular problem. The statistical ML classifier's treat each word independently and do not discriminate by position within the text sequence. The deep learning methods applied in (4) better captured the meaning of the transcript sequences due to their sequential nature. Though, was outperformed by BERT and XLNet's ability to identify contextual relationships within the transcripts. These results highlight the different assessment methods for AD classification, and ultimately show transformer learning based methods achieved higher accuracy and less validation loss. Recent research and practical applications suggest that while text-only models provide significant insights, particularly in linguistic analysis, multimodal models tend to deliver superior performance in terms of accuracy and comprehensiveness. This allows for a more holistic assessment of a patient, a richer more diverse set of features from both the audio and text, and models that are less susceptible to overfitting[4].

Feature Analysis in AD Classification

The purpose of (Meghanani, 2021) is to evaluate the classification problem of AD transcripts and MMSE scores prediction (MMSE score refers to a regression task, created to evaluate a participants Mini Mental Status Examination score (MMSE) based on speech and/or language data. This score quantifies cognitive impairment). (6) evaluates two models for AD classification, a FastText model and a CNN model with a single convolutional layer to capture n-gram-based linguistic information. FastText can be used as a CBOWs model or Skip-gram model specifically for learning text representations for text classifiers. Representations created by FastText can be used for a plethora of applications from data compression, as features into additional models, for candidate selection, or as initializers for transfer learning. The FastText model uses a bag of bigrams and trigrams in order to catch word orderings. The CNN model captures different n-grams (2, 3, 4, and 5) by adapting the kernel size to n. For both models, embeddings are created using pretrained GloVe vectors.

FastText models outperform several baseline methods historically used in AD classification. They are shallow in nature and tend to perform training and evaluation significantly faster than other deep learning models. Other research has shown that language impairment, including repetitive speech, lexical retrieval, and loss of verbal fluency, is often indicative of mild to severe cognitive impairment. A CNN model was chosen specifically to capture linguistic information in the n-grams present in an input transcript. Meghanani expounds on this by saying, “Any n × d CNN filter, where n is the number of sequential words looked over by the filter and d is the dimension of word embedding, can be viewed as a feature detector looking for a specific n-gram in the input that can capture the language impairments associated with AD.” Both models are used because of their ability to capture linguistic features from n-grams[6].

Text transcripts from the Alzheimer’s Dementia Recognition through Spontaneous Speech (ADReSS) dataset are used for evaluation. Like the Pitt Corpus, transcripts contain a conversational task between a participant and investigator. Transcripts include supplemental information, such as filler words, coughing, and pauses in speech. Each transcript is treated as a single datapoint with a correlating AD label and MMSE score, both of which were provided by the ADReSS challenge. After preprocessing steps were taken, the transcripts were broken up into two categories. One includes solely the speech from the participant, the other features speech from both the investigator and the participant.

For the AD prediction problem, the models were trained for 100 epochs with a batch size of 16. Adam optimizer is used with a learning rate of 0.001. Conversely, the MMSE task trained for 1500 epochs. The AD classification problem used binary cross-entropy as a loss function and the MMSE prediction task utilized a fully connected linear activation function.

The bootstrap method, bagging, was used to predict the final labels and MMSE scores for all test samples across both tasks.

RMSE (root mean square errors) and accuracy were used to evaluate the models. On the MMSE prediction task the CNN model scored a 4.38 and the fastText model scored 4.28. The best performing model was fastText with bi-grams on the transcripts that included both participants and investigators speech. The cross-validated accuracy was 86.09%. Across the different CNN models evaluated, tri+4+5 grams gave the best accuracy rates between both sets of transcripts. Overall, (Meghanani, 2021) found both models benefitted from keeping in utterances from the investigator. This is likely due to the variance between the speech of an AD participant and investigator.

Dementia Detection using Transformer-Based Deep Learning and Natural Language Processing Models performed the binary text classification with five pre-trained transformer models including: BERT, ALBERT, XLNet, RoBERTa, and ELECTRA. They utilized the Pitt corpus, the ADReSS challenge, an augmented set of the ADReSS challenge, and UW semi-structured interviews in an effort to compare transcripts to semi-structured interviews, as well as evaluate the effect of data augmentation on prediction. The UW transcripts used Type-Token Ratio for manual linguistic analysis. Data was split using an 80:20 train to test set ratio, and 20% of the training data was held back for the validation set fine tuning for each model. The ADReSS set was augmented using Random Deletion (RD) of the Easy Data Augmentation (EDA). Performance was evaluated using validation accuracy and prediction, precision, F1-score, recall, and Mcc. The results found across models; the UW interviews predictions were not consistent. The BERT model evaluating the augmented ADReSS challenge achieved the highest F1-score at 90%. Their results did show some performance gains can be made using an augmented dataset. Though, the overall performance of the original ADReSS challenge and the Pitt Corpus lagged slightly behind other foundational research[8].

Alternative Methods for AD Classification

Works such as (Bouazizi, 2023), evaluated alternative approaches to traditional deep learning and transfer learning methods for AD classification. Advanced language models have been utilized for multiple speech and cognitive classification problems the last few years. This is not without cost. LLMs require a large amount of data and computational power to train. This is often outside the realm of possibility for many researchers and is unnecessary for many classification tasks. While pretrained models minimize that cost, they do little to address the overfitting problem often found when using small datasets. This makes work unreliable at best and irreproducible at worst. Relying heavily on pretrained models also overlooks the fact that not all information for text classification problems can be found in the vocabulary and grammar related contexts. To assess this claim, Bouazizi extracts the topics subjects are talking about and how often they change from one sentence to the next or within one particular sentence for the Pitt Corpus image description task. This builds on previous work evaluating the importance of grammar and vocabulary in AD detection. This paper used unconventional methods to classify AD. The cookie theft image from the Pitt Corpus was split into regions of interest in order to identify which areas of the picture a participant was discussing. Independent from the data provided by DementiaBank, words that could be associated with each area of interest were collected from an online thesaurus and NLTK in order to collect synonyms and hypernyms to ultimately enrich the dataset. The transcripts were split into sentences, each sentence was checked for the presence of words associated with that region, and then that number was counted. The goal was to observe how the topics of sentences changed over time. An LSTM model was then trained for the binary classification task and MMSE regression. The LSTM was meant to capture trends associated with both the control and dementia group, and did so moderately compared to existing literature. (Bouazizi, 2023) also augmented samples of data using GPT-4. They were able to achieve an 83.56% accuracy at the subject level and 82.07% accuracy at the sample level on the DementiaBank cookie theft task. The results show control subjects spend more time describing certain aspects of the cookie theft image compared to their AD counterparts. Dementia subjects also tended to jump from one topic to another, having a higher degree of variance in their sentence subjects than members from the control group. Overall this highlights alternative methods for perusing AD classification[5].

Feature Importance

The work conducted by (Vimbi et al., 2024) provides a systematic review on the application of LIME (Local Interpretable Model-agnostic Explanations) and SHAP (Shapley Additive exPlanations) in the detection of Alzheimer's Disease (AD) using artificial intelligence (AI) (Vimbi et al., 2024). This work is pivotal in the field of explainable artificial intelligence (XAI), particularly within the context of medical diagnosis where understanding the reasoning behind AI predictions is crucial for clinical acceptance and trust. The study emphasizes the necessity of XAI in enhancing the transparency and trustworthiness of AI-based diagnostic systems, making them more acceptable for clinical use. It adheres to rigorous systematic review methodologies (PRISMA and Kitchenham's guidelines) to collate and analyze data from multiple studies, ensuring comprehensive coverage of the current landscape of XAI in AD detection. The paper delves into the capabilities, benefits, and challenges of using LIME and SHAP frameworks. These tools help interpret AI-driven decisions in AD diagnostics, demonstrating how they can provide meaningful insights into the decision-making processes of complex models. By integrating these XAI frameworks, the paper highlights their role in improving the fidelity of clinical decision-support systems, aiming to facilitate early intervention and better management of Alzheimer's disease[23].

Training Models on Oversampled Data and a Novel Multi-class Annotation Scheme for Dementia Detection by Nadine Abdelhalim, Ingy Abdelhalim, and Riza Batista-Navarro explores an advanced approach for classifying dementia through text analysis of patient-doctor conversations (Abdelhalim, 2023). This study is crucial as it employs a novel three-class annotation scheme that categorizes patients into Healthy Control (HC), Mild Cognitive Impairment (MCI), and Dementia stages, enhancing the granularity of dementia detection beyond traditional binary classifications. The study utilizes a dual approach to address data imbalances and increase the robustness of the findings:

The first approach involves oversampling underrepresented classes (MCI and Dementia) in the Pitt dataset to balance the dataset effectively. This is crucial as it allows the models to learn more representative features of these categories without bias towards the more numerous HC class. The second approach combines the Pitt dataset with additional datasets from Holland and

Kempler, introducing a richer variety of text data and discussion topics which help in generalizing the model's application.

For model development, the authors employed bidirectional transformers based on BERT, RoBERTa, and DistilBERT architectures, renowned for their effectiveness in text classification tasks. These models were trained to classify transcripts into the three specified classes, employing stratified 10-fold cross-validation to ensure robust evaluation.

The models achieved impressive classification accuracy and macro-averaged F1 scores, particularly, the DistilBERT model trained on the oversampled Pitt dataset and the combined dataset exhibited state-of-the-art performance with an accuracy of 98.8% and a macro-averaged F1-score of 98.6%, respectively.

LIME was employed to analyze the output of the bidirectional transformer-based models (BERT, RoBERTa, and DistilBERT) that were trained to classify text into three dementia-related categories: Healthy Control (HC), Mild Cognitive Impairment (MCI), and Dementia. The main goal of using LIME in this context was to provide explanations for individual predictions, highlighting the specific words or phrases in the patient-doctor conversation transcripts that influenced the models' classifications.

The paper does not detail specific results from the LIME analysis, such as the exact words or phrases identified as influential. However, it emphasizes that LIME was instrumental in manually inspecting the models' outputs, which involved saliency highlighting to visually represent the importance of different parts of the text. This technique helps in pinpointing linguistic features that are indicative of dementia and could be used as diagnostic markers. The use of LIME allowed Abdelhalim, to better understand model behavior, enhance model trust, and identify any potential biases[24].

In the paper *Explainable Identification of Dementia from Transcripts Using Transformer Networks* by Loukas Ilias and Dimitris Askounis, the authors address the challenges of identifying Alzheimer's dementia using transformer-based networks while emphasizing the importance of model interpretability (Ilias, 2022). They integrate the prediction of dementia and the evaluation of its severity (Mini-Mental State Exam scores) into a unified approach rather than treating them as separate tasks. The paper uses several transformer-based models, including BERT, which demonstrated the highest accuracy of 87.50%. They also developed an interpretable siamese network architecture, which achieved an accuracy of 83.75%. Additionally, the study introduced two multi-task learning models that simultaneously addressed dementia identification (binary classification) and severity assessment (multiclass classification), achieving accuracy up to 86.25%.

The ADReSS Challenge Dataset was used, which focuses on minimizing bias by matching for gender and age and includes spontaneous speech recordings along with their transcriptions from both Alzheimer's patients and non-demented controls.

LIME was specifically utilized to interpret the best performing model's predictions, enhancing the understanding of the linguistic differences between dementia and non-dementia groups. Part-of-speech tagging was involved in detailed linguistic analyses to discern patterns more effectively. The results from LIME corroborated with linguistic and speech markers, indicating that explanations from LIME aligned well with known linguistic traits associated with dementia, such as simplified vocabulary and grammar structures, which are commonly seen in patients with dementia[25].

This research builds upon and extends the findings from previous studies by employing a comprehensive and innovative approach to the detection and analysis of using speech and language data. While the aforementioned works have laid a strong foundation by exploring various machine learning and NLP techniques, this work introduces further methodological enhancements and integration strategies. By leveraging a nuanced combination of deep learning models, and innovative preprocessing strategies that maintain critical linguistic features often discarded in traditional analyses, this study aims to provide a more accurate and holistic understanding of the linguistic markers associated with AD. Additionally, the application of advanced interpretability tools such as LIME and SHAP enables a deeper insight into the decision-making processes of the models, thus contributing to a more transparent and replicable research framework. Collectively, these efforts not only enhance the capabilities of AD prediction models but also address the critical challenge of reproducibility in computational research, paving the way for future studies to build upon this work with greater confidence and scientific rigor.

III. Dataset

This study utilizes the DementiaBank dataset, specifically the Pitt Corpus, a key component of the TalkBank system (Becker, 1994), which provides valuable resources for the linguistic analysis of cognitive disorders such as AD. A particularly pertinent subset of this dataset is the cookie theft picture description task, a speech based medical examination for neurological diseases, wherein participants were asked to describe everything they could see in a complex scene depicted in an image. A picture description assignment is one of the best techniques for getting an appropriate standardized speech sample across a variety of subjects. The cookie theft picture description task includes 243 control samples and 309 dementia samples. This task is designed to elicit rich linguistic output that captures a range of cognitive functions (including modalities of perception: gestural, auditory, and visual; Functions of processing analysis like: problem solving and comprehension; modalities of response: articulation and writing manipulation, etc.), making it an excellent tool for identifying early linguistic markers of dementia. Participants without neurological impairment are able to perceive and identify every aspect of the picture, whereas, those with cognitive decline may struggle with a range of cognitive skills, such as attention, memory, and description. They may not recall recounting certain scenes and repeat them multiple times, use repetitive language to describe the picture, be unable to relay the information in a logical format or present a cohesive description of the image. The picture included a mother washing dishes and her children stealing cookies from a jar and can be found in Figure One.

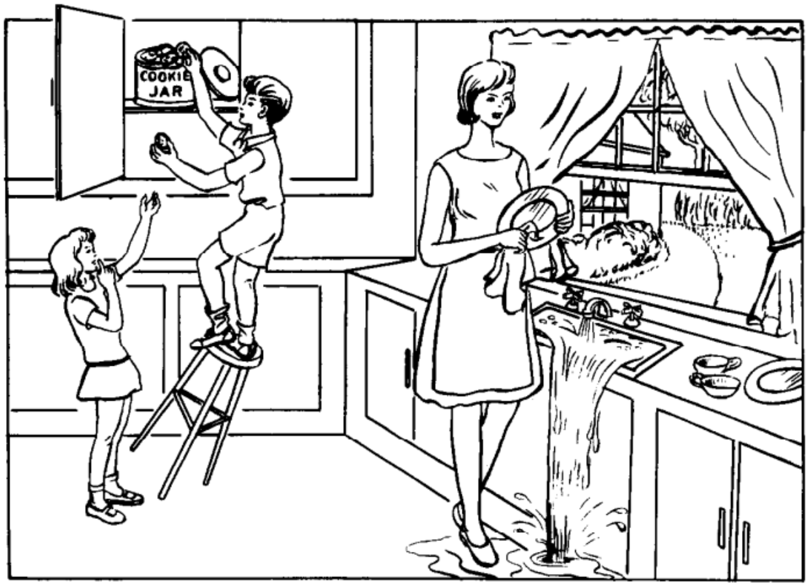


Figure 1: Cookie Theft picture, adapted from Matošević et al. 2022

The participant responses describing the above image were recorded and subsequently transcribed verbatim. The dataset used in this thesis consists of speech transcripts produced in the transcription software CLAN, which allows for detailed annotations and structuring of dialogue data (MacWhinney, 2010). In order to prepare this data for analysis, several preprocessing steps were necessary, many of which were adapted and implemented through the TRESLE preprocessing framework (Li, 2023). TRESLE, which stands for Toolkit for Reproducible Execution of Speech, Text, and Language Experiments, was designed to facilitate the reproducibility of computational models that assess speech and language changes associated with cognitive impairments, such as those caused by AD. TRESTLE, allows options to remove clear throat indicators, parenthesis and brackets, noise indicators, multiple spaces, capitalization and punctuation, and non-alpha numeric characters. Specific steps taken by this paper included:

- **Punctuation Handling**: All punctuation marks were stripped from the transcripts except for periods, which are crucial for maintaining the integrity of sentence structures.

- **Speaker Separation**: The speech of the investigators was removed, ensuring that only the participants' responses were retained for analysis.

- **Whitespace Correction**: Extraneous spaces within the transcripts were removed to standardize the text formatting.

- **Stop Word Retention**: Unlike typical text processing that might remove stop words to focus on more meaningful words, in this context, every word including stop words was kept to preserve the natural flow and subtle linguistic cues in the speech.

- **Character Filtering**: Non-ASCII characters were removed, and all symbols were excluded except alphabetic characters, to focus solely on the linguistic content.
- **Case Normalization**: The text was converted to lowercase to ensure uniformity across the dataset, facilitating more straightforward textual analysis.
- **Pause Notation**: Pauses, indicated in the transcripts, were retained as they provide significant insights into the speech patterns and cognitive load in individuals, which are critical for this analysis.
- **Ampersand Notation and Underscore Replacement**: Any text elements marked with an ampersand (indicating special annotations or actions within speech) are normalized or removed based on the research requirements. Additionally, underscores within the text are replaced with spaces to ensure consistency and readability in the transcript formatting.

Once the dataset was preprocessed, it was divided into three subsets to support different phases of the model development process: training, testing, and validation. Specifically, the dataset was split into proportions of 70%, 20%, and 10%. The training set, constituting 70% of the data, was utilized to train the various models described in this study. The test set, making up 20% of the data, was employed to evaluate the performance of the models post-training. Lastly, the validation set, which comprised the remaining 10% of the data, was crucial for tuning the hyperparameters of the models.

These preprocessing steps were critical for ensuring that the data was clean, structured, and suitable for the subsequent analysis using advanced machine learning models. By standardizing and refining the dataset in this manner, the study aims to isolate and identify linguistic features that are indicative of AD, thereby supporting the effectiveness of the predictive models developed in this research.

SECTION IV: Methodology

The figure presented below illustrates the proposed architecture for this study, and displays a comprehensive outline of the framework employed.

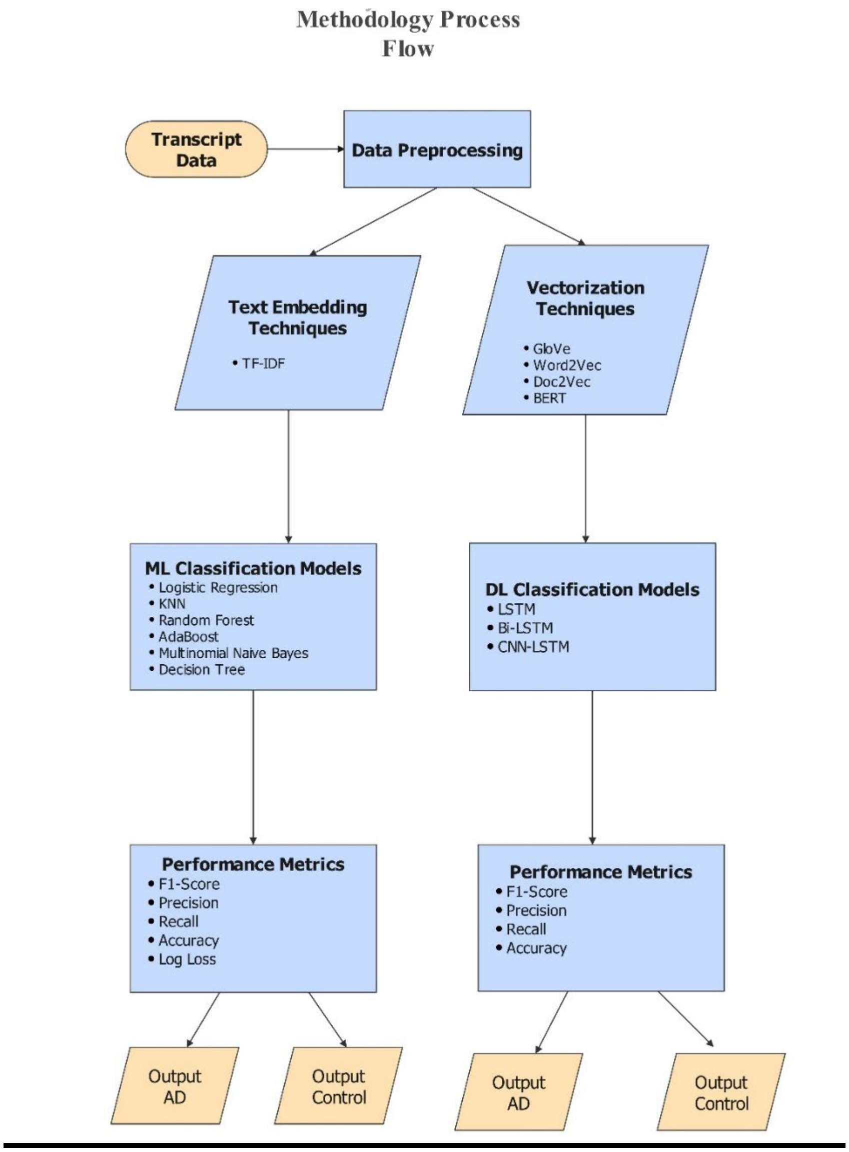


Figure Two: Proposed Architecture

4.1 Vectorization Method: TF-IDF

Vectorization Configuration

Term Frequency-Inverse Document Frequency (TF-IDF) is a quantitative measure used to gauge the relevance of a word to a document within a corpus, relative to its frequency across the corpus. This statistic is crucial for tasks such as search engine optimization and information retrieval, as it helps differentiate between commonly used words and those that provide specific insight into the content of a document. The TF component of the measure calculates the frequency of a word in a single document, while the IDF component scales down this frequency by the number of documents that contain the word. This dual approach helps to attenuate the effect of terms that appear frequently across documents, emphasizing words that are more unique to individual documents.

Configuring the TfidfVectorizer

In this study, the TfidfVectorizer from the Scikit-learn library was strategically configured to optimize the extraction and analysis of linguistic features relevant to Alzheimer's Disease:

- **Stop Words**: The inclusion of a parameter to exclude common English stop words ('the', 'is', 'and', etc.) is crucial as these words are typically poor discriminators in thematic content analysis. Their removal increases the focus on more meaningful terms that are potentially indicative of cognitive patterns specific to Alzheimer's Disease.

- **Maximum Features**: By setting a cap of 1000 on the number of features, the model focuses on the top terms by term frequency, thus simplifying the model and enhancing its computational efficiency. This limitation also helps prevent overfitting by not overly tailoring the model to the noise within the training data.

- **Minimum Document Frequency (min_df)**: This threshold was set at 5 to ensure that only terms that appear in at least five documents are considered. This filter helps to remove anomalies or rare occurrences that might otherwise skew the analysis.

- **Maximum Document Frequency (max_df)**: By excluding terms present in more than 90% of the documents, the model avoids terms that are too common and therefore not useful for distinguishing between different document types or contents.

- **N-Gram Range**: The vectorizer was configured to consider both unigrams (single words) and bigrams (pairs of words). This range allows the model not only to assess the importance of individual words but also to capture the context provided by adjacent word pairs, enhancing the model's ability to recognize more complex linguistic structures which could be critical in identifying subtle cognitive impairments.

To systematically evaluate the impact of different n-gram configurations on model performance, the study conducted separate analyses using unigrams only, bigrams only, and a combination of both. This approach enabled the paper to discern which n-gram configuration most effectively captures the linguistic nuances associated with AD. It was hypothesized that while unigrams might highlight prevalent words, bigrams could reveal more about the relationships between words, potentially identifying patterns that are not apparent from single-word analysis alone. The combined analysis aimed to leverage the strengths of both unigrams and bigrams, potentially offering a more comprehensive view of the text data.

Data Transformation Process

Utilizing the configured *TfidfVectorizer*, text data from the 'text' column of the talk_bank_small dataset was transformed into a TF-IDF matrix. This dataset, containing transcribed speech data from subjects, is pivotal for identifying linguistic markers of AD. The transformation process converts this textual information into a numerical format, where each row of the matrix represents a document, and each column corresponds to a TF-IDF score for a term, facilitating subsequent machine learning analyses.

Matrix Dimensions

The resulting TF-IDF matrix, denoted as X, measures 549 rows by 954 columns for unigrams and bigrams. For bigrams X measured at 549 rows by 610 columns, and for unigrams only X was represented by 549 rows by 344 columns. Each row represents a unique document from the dataset, and each column corresponds to a distinct term identified within the corpus. The dimensionality of X reflects the methodological choices made, capturing the most significant terms as determined by the TF-IDF vectorization process, constrained by the specified maximum features and n-gram range.

4.2 Machine Learning Models Using TF-IDF

In the pursuit of identifying the most effective machine learning model for detecting linguistic markers indicative of AD, several models were evaluated using the TF-IDF vectorized data. This section outlines the specifics of each model's operational framework and their performance based on accuracy, precision, recall, F1 score, and log loss metrics.

Logistic Regression

Logistic Regression is a statistical model that in this context, estimates the probabilities of binary outcomes based on input features derived from TF-IDF scores. It is particularly useful for this kind of binary classification task because it provides a direct probabilistic interpretation for class membership (Alzheimer's vs. non-Alzheimer's). The model demonstrated robust performance with a notable accuracy and a balanced precision-recall trade-off, reflecting its ability to handle linear relationships within the data.

K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) classifies new cases based on a similarity measure (e.g., distance functions). KNN has been included in the analysis to leverage its non-parametric nature, which makes no assumptions about the underlying data distribution. This model is intuitive and effective, particularly when there is little or no prior knowledge about data distribution. In this study, KNN showed a competitive accuracy, suggesting it can effectively capture the complexities in the data introduced by the high-dimensional TF-IDF vectors.

Decision Tree Classifier

Decision Tree Classifier builds a model in the form of a tree structure. It breaks down the dataset into smaller subsets while at the same time, an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes, which is easy to interpret and understand. This model's simplicity can be particularly advantageous for explaining decision-making processes but showed some limitations in handling the sparse matrix from TF-IDF, as reflected in the lower accuracy and higher log loss compared to other models.

Multinomial Naive Bayes

The Multinomial Naive Bayes classifier is a probabilistic learning method commonly used in NLP and document classification. This classifier operates under the foundational assumption of the Naive Bayes theorem, which posits that the presence (or absence) of a particular feature in a class is independent of the presence (or absence) of any other feature. This assumption, while simplistic, allows the model to perform efficiently even with complex datasets. Multinomial Naive Bayes is particularly well-suited for classification tasks involving discrete data, such as text classification where features are typically the frequencies with which words appear in documents. The model calculates the probability of each class based on the input features (word counts or frequencies) and then predicts the class with the highest probability.

Random Forest Classifier

Random Forest Classifier is an ensemble of Decision Trees, typically trained via the bagging method. The general idea of the bagging method is that a combination of learning models increases the overall result. Put simply: Random Forest builds multiple decision trees and merges them together to get a more accurate and stable prediction. Random Forest constructs multiple decision trees, referred to as base models or weak learners, during the training phase. This ensemble technique takes the majority vote from these trees to decide the output class, enhancing prediction accuracy and robustness over a single decision tree. In this study, the Random Forest model was configured with several parameters optimized for textual data analysis: In this methodology, various parameters of random forest, such as n estimators = 100, min sample leaf = 1, min_samples_split=2, max_depth=None, min sample split = 2, criterion = gini, are used. Random Forest showed a strong performance across all metrics, suggesting that it can manage the high feature dimensionality well, providing a balance between overfitting and underfitting.

AdaBoost Classifier

AdaBoost Classifier begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases. This model is used to boost the classification performance by combining multiple weak models to create a strong classifier. AdaBoost's performance in this setup highlights its ability to adaptively focus on challenging cases in the dataset.

Grid Search

Grid search is a widely used technique for hyperparameter tuning in machine learning, aiming to find the optimal combination of parameters for a given model. This method systematically constructs and evaluates a model for each combination of algorithm parameters specified in a predefined grid. Each parameter configuration is validated through techniques like cross-validation to determine which set of parameters achieves the best performance metrics.

In the context of this study, the grid search method was employed to explore a variety of hyperparameter settings across different machine learning algorithms. The parameters varied included regularization techniques and strength for logistic regression, the number of neighbors and the weighting method for K-Nearest Neighbors, criteria for splitting in decision trees and forests, learning rates for boosting methods, and kernel types for Support Vector Machines, among others.

Despite the comprehensive exploration of hyperparameter spaces facilitated by grid search, the results indicated that the default settings of the models generally outperformed the alternative combinations. This outcome suggests that the default parameters, often chosen based on extensive empirical evidence and best practices in the field, are robust across a range of datasets and scenarios. This finding underscores the importance of empirical validation even when theoretical considerations might suggest alternative configurations. It also highlights the effectiveness of the default settings in providing a strong baseline performance, simplifying the model development process without extensive tuning efforts.

4.3 Embedding Methods

GloVe Word Embeddings

GloVe (Global Vectors for Word Representation embeddings) are an unsupervised method of word representation that captures the associations between words through the aggregation of global word-word co-occurrence statistics from a corpus (**source**). The primary idea behind GloVe's approach is to derive semantic relationships between words by examining how frequently words appear together in a context window within a given corpus. Unlike other embedding techniques that focus solely on local context, GloVe constructs an explicit word-context or word-co-occurrence matrix using statistics across the entire corpus, thereby encoding both local and global context information in the embedding space.

In this research, the implementation of GloVe embeddings involved several steps to transform raw text data from the dataset into a format suitable for deep learning models. These vectors were obtained from the GloVe model pre-trained on a large external dataset, allowing the embedding to encapsulate a rich set of semantic relationships found between transcripts. Following tokenization, the GloVe pre-trained word vectors were leveraged to create an embedding matrix that represents each word in the vocabulary by a dense vector of fixed size.

Word2Vec

Developed by Tomas Mikolov and colleagues at Google in 2013, Word2Vec is designed to map words into an embedding space where the geometric relationships between these vectors reflect the semantic relationships between the words themselves (15). Word2Vec employs a shallow neural network model that utilizes one of two architectures: Continuous Bag of Words (CBOW) or Skip-Gram, both of which are grounded in the distributional hypothesis of linguistics—words that appear in similar contexts tend to have similar meanings. Both models optimize the same objective function, which is to maximize the log probability of observing a context word given a word, over all word-context pairs observed in the corpus. This paper implements the Word2Vec model using CBOW. Formally, for Skip-Gram:

$$\sum_{(w,c)\in D} \log P(c|w)$$

Where *D* is the training corpus, *w* is a word, and *c* is a context word. The probability *p*(*c*|*w*) is computed using the softmax function:

$$P(c|w) = \frac{\exp\left(\mathbf{v}_c^{\mathsf{T}}\mathbf{v}_w\right)}{\sum_{c'\in C}\exp\left(\mathbf{v}_{c'}^{\mathsf{T}}\mathbf{v}_w\right)}$$

Here, **v***w* and **v***c* are the vector embeddings of the word and the context, respectively. *C* represents all context words in the vocabulary. In this paper, the Word2Vec model is applied to transform the training data into vector space using the following configuration: Vector Size: 100 dimensions. Window Size: 2, allowing the model to consider words within two places of the target word, focusing on more immediate contextual relationships. Minimum Count: 1, ensuring even rare words are included, which could be crucial in clinical narratives where specific terms may be vital yet infrequent. Post-training, the word vectors are integrated into a neural network workflow via Keras, utilizing a custom function *w2v_to_keras_weights* to transfer the learned word vectors into a format suitable for embedding layers in deep learning models. This process enables the subsequent application of the embeddings in predictive modeling, specifically focusing on classifying transcripts for AD. The application of Word2Vec in this research is particularly aimed at enhancing the model's ability to discern subtle linguistic cues that might indicate cognitive impairments like Alzheimer's. The dense embeddings generated by Word2Vec provide a rich, nuanced representation of text data. The embedding matrix created from these Word2Vec embeddings was directly passed as an embedding layer in deep learning models.

Doc2vec

Doc2Vec, also known as Paragraph Vector, is an unsupervised algorithm for generating vector representations of variable-length pieces of texts such as sentences, paragraphs, and documents. Developed by Le and Mikolov in 2014, Doc2Vec extends the Word2Vec methodology to allow for the embedding of entire documents (16). This capability is crucial for tasks where the context provided by the entire document is necessary for understanding, such as in document classification, sentiment analysis, and patient record analysis. Doc2Vec is fundamentally designed to overcome the limitations of averaging word vectors (Word2Vec) loses the ordering of words and hence the meaning encoded in sequence. It introduces a document-level vector which serves as a unique tag for each document in the corpus. Distributed Memory (DM) preserves the word order in a document, acting similarly to a memory that remembers what is missing from the current context — or the paragraph vector. Distributed Bag of Words (DBOW) ignores the context words in the input but forces the model to predict words randomly sampled from the paragraph in the output. In both cases, words are projected into a continuous vector space along with the document itself. The document vector is trained to predict the words in a small context window. Each document's vector is unique in the model, allowing it to capture the essence of the document. During training, word vectors and paragraph vectors are trained using either the context of the words (DM) or by predicting words randomly sampled from the paragraph (DBOW). The loss function optimized during training is similar to that in Word2Vec, typically using negative sampling or hierarchical softmax. Shown as:

$$\text{Objective} = \text{argmin}_{\theta} \sum_{d \in D} \sum_{w \in d} \log\ P(w|\theta)$$

In this research, the Doc2Vec model is employed to capture the nuanced linguistic characteristics inherent in the training transcripts related to the classification of AD. The approach involves: Tokenizing the training data and tagging them with unique identifiers, setting vector size to 100, window size to 2, and minimum count =1. After training, document vectors are extracted and used to create an embedding matrix which serves as input to subsequent deep learning models for classifying AD.

BERT Input Embeddings

BERT (Bidirectional Encoder Representations from Transformers) is a groundbreaking method in the field of NLP introduced by researchers at Google AI in 2018 (17). BERT's primary innovation lies in its ability to train language models based on the entire set of words in a sentence or document. This approach allows BERT to capture the context of a word based on all of its surroundings (both left and right of the word). BERT utilizes the Transformer, an attention mechanism that learns contextual relations between words (or sub-words) in a text. The multi-headed attention mechanism allows the model to capture various aspects of the data in parallel, improving its contextual understanding of the text. It is a way to learn the inherent relation between single sentences or different related sentences in order to obtain a more robust representation of attention vectors for each word. These are in turn then used to compute a final attention vector consisting of a weighted average. To do this, the input vectors are linearly projected multiple times to form sets of queries (Q), keys (K), and values (V). This is illustrated by the equation:

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

Essentially, for each attention "head," there is a distinct set of projection matrices. Within each head, attention scores are calculated using the scaled dot-product attention. The result is then summed up for each query, producing an output vector for each head. The output vectors from all heads are concatenated, and once concatenated the output undergoes one final linear transformation through another learned weight matrix. This step integrates information from all the heads into a single vector for each token. The multi-head attention mechanism allows the model to attend to different parts of the input sequence differently. For example, one head might focus on the syntactic role of a word, while another might focus on its semantic role. This is particularly useful in the context of AD detection due to the complexity of analyzing speech.

In its essence, the Transformer includes two separate mechanisms—an encoder that reads the text input and a decoder that produces a prediction for the task. Unlike directional models, which read the text input sequentially (left-to-right or right-to-left), the BERT model is deeply bidirectional. BERT's architecture is composed of a multi-layer bidirectional Transformer encoder. Each layer aggregates information from both directions and all positions of the text sequences. To understand the order of words, positional encodings are added to the input embeddings to provide some information about the relative or absolute position of the tokens in the sequence. The model is pre-trained on a large corpus of text in two unsupervised tasks. First the model was trained as a Masked Language Model where random words are masked (hidden), and the objective is to predict the masked word based only on its context. Then trained in Next Sentence Prediction, where model predicts whether a sentence naturally follows a given sentence, which helps it understand relationships between consecutive sentences.

In the context of this this thesis, BERT is utilized to generate robust input embeddings for the training transcripts. The implementation involves the following steps: Tokenizing the transcripts into tokens that BERT can understand. Each token is converted into IDs that are fed to BERT to obtain corresponding embeddings. The BERT model is then used to transform these tokenized inputs into embeddings. Each token ID is converted into a dense vector of fixed size (768 dimensions in the case of BERT base models). BERT embeddings are then employed to enrich the feature set for deep learning models by providing a deep, contextualized representation of the Pitt corpus data.

4.4 Deep Classifiers

The advancements in neural network architectures have led to significant improvements in text classification tasks, particularly in the context of AD detection. As part of the ongoing exploration into more effective methods for detecting AD through binary classification, this study employs sophisticated deep learning models such as Long Short-Term Memory (LSTM), Bidirectional LSTM (Bi-LSTM), and Convolutional Neural Network-LSTM (CNN-LSTM). These models are particularly adept at processing sequential data, like text, where understanding the temporal dynamics is crucial. This section of the thesis details the implementation and configuration of these models, and outlines how they integrate the different embedding techniques discussed above—including GloVe, Doc2Vec, Word2Vec, and BERT—to enhance their predictive capabilities.

LSTM

Long Short-Term Memory (LSTM) models are a type of Recurrent Neural Network (RNN) specifically designed to avoid the long-term dependency problem, allowing them to remember information for extended periods. Unlike standard feedforward neural networks, LSTMs have feedback connections that make them capable of processing entire sequences of data. A key feature of LSTM units is their use of gated cells, which regulate the flow of information. These gates—forget, input, and output gates—determine which parts of a cell state should be retained or discarded, thus enabling the model to learn what to keep from long-term and what to remove from short-term memory.

The LSTM architecture used in this thesis is constructed using Keras. The first layer is an embedding layer, designed to convert input data into dense vectors of fixed size. In this study, the embedding layer is configured to use pre-trained vectors from GloVe, Doc2Vec, Word2Vec, or BERT, varying by each experimental run. The embeddings are set to non-trainable to preserve their pre-learned semantic properties. The models are trained using sequences of 100 tokens for uniformly shaped data, binary crossentropy as the loss function, batch_size = 256, epochs = 45, validation split= .01, metrics = accuracy, and optimizer = 'adam'. The output layer is a dense layer with a single neuron and a sigmoid activation function. One of the major drawbacks of using an LSTM model is that it only captures unidirectional context in a sentence, which is why Bi-LSTM models were also included to enhance robustness.

Bi-LSTM

Bidirectional Long Short-Term Memory (Bi-LSTM) networks are an extension of the traditional LSTM model that can enhance model performance by providing additional context. While standard LSTMs process data from past to future (left to right in a sequence), Bi-LSTMs run two LSTMs simultaneously, one in the forward direction and the other in the backward direction. This allows them to capture information from both past and future states, offering a richer understanding of context, which is particularly beneficial for complex sequence prediction tasks like text classification where context from both directions is crucial. Bi-LSTMs are particularly well-suited for tasks where the entire sequence context (both preceding and following information) is crucial for understanding the current element. This dual-direction processing capability makes Bi-LSTMs adept at handling sequences where the context in both directions is critical for accurate predictions. The outputs from both LSTMs are typically concatenated at each time step, which doubles the dimensionality of the output space compared to a standard LSTM. This concatenation allows the following layers to learn from the complete history of inputs (both past and future relative to a given time step).

The Bi-LSTM is also built using Keras. The first layer is one of the embedding matrices. Input sequences = 100, loss = binary crossentropy, batch_size = 256, epochs = 45, validation split= .01, metrics = accuracy, and optimizer = ‘adam’, activation = sigmoid.

CNN-LSTM

The CNN-LSTM architecture combines the strengths of Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks, creating a powerful model for handling tasks that involve both spatial and temporal dependencies. This architecture is particularly effective in areas such as video frame prediction, audio signal classification, and complex sequence modeling tasks where both local features (extracted by CNNs) and long-range dependencies captured by LSTMs are crucial. Unlike standard LSTMs that only process temporal information, CNN-LSTMs incorporate convolutional layers that precede LSTM layers, allowing the model to first extract local features through convolutions and then analyze the temporal dynamics through recurrent processing. These layers apply a series of filters to the input for feature extraction, which identifies local patterns such as edges in images or key features in textual data. In text classification, these features might correspond to patterns of keywords or phrases that are indicative of certain topics or sentiments. Following convolution, pooling layers reduce the dimensionality of the data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer, thus reducing the computational complexity and controlling overfitting. After processing through CNN and pooling layers, the data is fed into LSTM layers, which can then process the sequence data with an awareness of the temporal dependencies between the extracted features.

The hybrid CNN-LSM is implemented using the Keras-API TensorFlow. The first layer is one of the embedding matrices. Input sequences = 100, loss = binary crossentropy, batch_size = 256, epochs = 45, validation split= .01, metrics = accuracy, and optimizer = ‘adam’, activation = relu, dropout layer= 0.1. The convolutional layer consists of 32 filters with a kernel size of 5, the max pooling layer reduces the spatial dimensions with a pooling size of 4, and the LSTM layers process the feature-mapped sequence output from the CNN layers with 64 LSTM units.

Feature Importance

In the pursuit of enhancing the interpretability of machine learning models in the context of AD classification, this thesis dedicates a section to exploring the significance of various features utilized by both machine learning and deep learning models. Feature attribution attempts to quantify the significance of input characteristics based on deep neural network predictions. For the top-performing machine learning models, including Random Forest with bigrams, Random Forest with both bigrams and unigrams, and Logistic Regression with unigrams and bigrams, feature importance is assessed using SHAP (SHapley Additive exPlanations) and permutation importance. These methods are chosen for their ability to provide a comprehensive, local perspective on the contribution of individual features to the prediction outcomes, highlighting the most influential factors in the classification process. Meanwhile, for the leading deep learning models—specifically those employing Doc2Vec LSTM, Doc2Vec Bi-LSTM, Doc2Vec CNN-LSTM, and BERT CNN-LSTM—feature importance is analyzed through LIME (Local Interpretable Model-agnostic Explanations) and integrated gradients. These methods show how particular features impact model decisions, offering a deeper understanding of the underlying mechanisms driving the models' performances. This comprehensive approach to feature importance not only clarifies the predictive dynamics of advanced classification models but also informs further model refinement and feature engineering efforts, ultimately aiming to enhance the diagnostic accuracy and reliability of AD screening tools.

Machine Learning Feature Importance

SHAP

SHapley Additive exPlanations (SHAP) is a powerful tool for explaining the output of any machine learning model by quantifying the contribution of each feature to the prediction (Lundberg et al., 2017). SHAP is grounded in the principles of cooperative game theory, specifically the Shapley values—a method developed by Lloyd Shapley in 1953. Shapley values provide a fair distribution of payouts (model predictions) among the players (features), considering the contribution of each player to every possible coalition. The Shapley value is calculated for each feature across all possible combinations of features to determine its average marginal contribution to the model's output. The mathematical formulation is as follows:

$$\phi_j = \sum_{S \subseteq N\{j\}} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} [v(S \cup \{j\}) - v(S)]$$

Where:

- N is the set of all features.
- S is a subset of features excluding feature jj.
- $v(S)$ is the prediction function evaluated with the features in set SS.
- ϕj is the Shapley value for feature jj, representing the average contribution of feature jj to the change in the prediction from the baseline prediction (the average prediction over the dataset).

When applying SHAP to text data, such as transcripts from AD assessments, the model's predictions are decomposed into the contributions of individual words or phrases within the transcript. This decomposition allows for an in-depth analysis of how certain terms or linguistic features influence the model's outputs, which is crucial for understanding complex models like those used in AD classification.

Transcript data must first be transformed into a numerical format that machine learning models can process, typically using techniques like TF-IDF, word embeddings, or Doc2Vec. A predictive model is then trained on the transformed text data. This model could be a deep learning model or any other algorithm suitable for text analysis. After training, the SHAP values are computed for each feature (word or phrase) across many possible combinations of features, quantifying each feature's impact on the model's output. SHAP provides clear, actionable insights into which words or phrases in a transcript are driving the model's predictions and identifies whether specific features consistently contribute positively or negatively to outcomes. Meaning, a positive contribution refers to features that increase the probability of the predicted outcome, such as classifying a transcript as indicative of AD. While, a negative contribution refers to features that decrease the probability of the predicted outcome or suggest an alternative classification, such as a control group without (non-AD). Essentially, SHAP values can reveal how much each word or phrase shifts the model's output closer to one class or another, helping to clarify the linguistic patterns that the model relies on for its decisions.

In the realm of machine learning, enhancing the interpretability of algorithms, particularly for unstructured data like text, has emerged as a vital area of research. While numerous methodologies have been developed for interpreting structured data, the complexity of dealing with unstructured datasets such as textual content introduces unique challenges. The SHAP summary plot is a powerful visual tool used to display the impact of each feature across a dataset[19].

Permutation Importance

Permutation importance is a model-agnostic technique used to measure the importance of features in a predictive model. Unlike model-specific methods that rely on internal model parameters, permutation importance provides a straightforward and intuitive understanding of feature significance based on changes in model performance. The method works by evaluating the decrease in a model's accuracy after the values of a particular feature have been randomly shuffled. This shuffling breaks the relationship between the feature and the target, highlighting how much the model's prediction relies on the feature.

In the context of this study, permutation importance was applied to assess the significance of features in a Logistic Regression model trained on unigrams and bigrams from text data. The approach is particularly valuable for unstructured data such as text, where it is crucial to understand which words or phrases (features) are most influential in classifying documents. By randomly shuffling the values of each feature in the test set and observing the deterioration in the model's accuracy, one can infer the relative importance of each feature.

This particular implementation of permutation importance involved computing the mean importance and standard deviation of importance scores across 10 repetitions, ensuring the reliability of the importance estimates by mitigating the random variability in the shuffling process. The logistic regression model was evaluated using a test dataset (*X_test, y_test*), with the aim to identify the top 25 features (unigrams and bigrams) that have the most significant impact on the model's ability to classify Alzheimer's Disease accurately.

Deep Learning Feature Importance

LIME

Local Interpretable Model-agnostic Explanations (LIME) is an innovative technique designed to enhance the interpretability of complex machine learning models (Ribeiro et al., 2016). It is particularly valuable in fields like healthcare, where understanding the reasons behind a model's predictions is crucial for trust and actionable insights. LIME provides explanations for individual predictions, which helps in demystifying complex models that can often seem like "black boxes." LIME is designed to explain the predictions of any classifier or regressor in a faithful way by approximating the model locally with an interpretable model. The core idea behind LIME is to perturb the input data and observe how the predictions change, which provides insight into the behavior of the model near the vicinity of the input being explained.

LIME generates new samples around the vicinity of the input by perturbing it. For tabular data, this might involve slight modifications to feature values; for text, it involves creating similar texts by removing words. Then those perturbed samples are used to train a simple, interpretable model, such as a linear regression or decision tree, restricted to the locality of the original instance. Each of the perturbed samples is weighted according to their proximity to the original instance, with closer samples receiving higher weights. This ensures that the local model is faithful to the area around the data point being explained. The local model is then used to explain each prediction in terms of the contributions of each feature. For a linear model, these contributions can be directly interpreted as the coefficients of the model.

The mathematical rationale for LIME involves fitting a linear model g that approximates the predictions f of the complex model locally. If x represents the original instance and ξ represents the perturbed instances, LIME solves the following optimization problem:

$$\xi = \text{argmin}_{g \in G} L(f, g, \pi x) + \Omega(g)$$

where L is a measure of how unfaithful g is at approximating f over the perturbed dataset, $\pi x \pi x$ is a proximity measure that defines the locality around x, and $\Omega(g)$ is a complexity measure of the model g.

In the medical domain, the demand for explainable models is particularly high. Clinicians need to understand why a model makes certain predictions to trust its reliability and to integrate these insights into their decision-making processes. For example, understanding why a model predicts a high risk of dementia based on certain speech patterns or other biomarkers can inform better treatment plans and patient management strategies.

Feature importances in LIME can be categorized into two types: global and local. Global importance refers to the overall impact of a feature across all predictions, providing a broad view of feature relevance. Local importance, on the other hand, examines the impact of features on individual predictions. This distinction is crucial because features that are globally important may not be significant in specific cases, and vice versa. Local explanations allow researchers to tailor their interpretations and considering unique factors that may of otherwise been overlooked[22].

Integrated Gradients

Integrated Gradients is a method designed for attributing the prediction of a neural network to its input features (Sundararajan et al., 2017). Developed by Sundararajan, Taly, and Yan, this technique leverages the axiomatic approach to ensure robust and meaningful feature attributions which are critical in many applications, particularly in the medical and financial fields where explanations are vital for trust and legality. The core concept behind Integrated Gradients is to connect the input of interest to a baseline input (a starting point with no predictive signals) through a straight path and to compute the gradients of the output prediction with respect to the input features at points along this path. The contribution of each feature is then quantified by integrating these gradients along the path, from the baseline to the input.

Formally, for a given function $F{:}Rn$→[0,1] representing the network, and an input x, the integrated gradient along the ith dimension for x relative to a baseline x' is defined as:

$$IntegratedGrads_i(x)::==(x_i-x_i')\times\int_{\alpha=0}^{1}\frac{\partial F(x'+\alpha\times(x-x'))}{\partial x_i}d\alpha$$

Integrated Gradients offers a significant contribution to the interpretability of deep neural networks, especially in the realm of NLP. This method provides a systematic approach to understanding feature importance across different model architectures, regardless of their complexity. It is particularly vital in scenarios where explainability is crucial, such as medical diagnostics, financial decision-making, and other sensitive domains where stakeholders require transparency and clarity on how decisions are derived.

The principle behind Integrated Gradients is based on attributing the prediction of a deep network to its input features by observing how the predictions change as inputs are varied along a straight path from a baseline to the actual input. The baseline is typically a type of input that represents an absence of features and is often chosen as a zero vector in text models. The path integral of gradients along this path quantifies the contribution of each feature to the final prediction.

For text data, particularly in deep learning models like LSTM or Transformer-based architectures, the baseline often represents a state with no textual input, such as a zero-embedding vector. The gradients calculated reflect how each component (word or sub-word token) of the input text contributes to the final prediction, offering insights into which words or phrases are most influential, and potentially why certain decisions are made by the model.

Integrated Gradients stands out for its simplicity, requiring only basic gradient computations, and for its robustness, adhering to desirable axiomatic properties such as sensitivity and implementation invariance. It enhances our understanding of complex NLP models and also supports efforts to make AI more accountable and transparent. By methodically breaking down the contribution of each input feature, integrated gradients helps validate model behavior and helps ensure that model decisions are grounded in observable and justifiable patterns in the data. This makes it a powerful tool for evaluating the behavior of complex models, particularly in high-stakes applications where understanding model reasoning is as crucial as the accuracy of its predictions[27].

The application of integrated gradients for feature importance analysis exemplifies a meticulous approach to understanding the contributions of individual input features to the predictions made by deep learning models. This section details the implementation methodology and integrated gradients in this paper, by enhancing transparency and interpretability within the predictive modeling process. integrated gradients, as a technique, provides a structured method for attributing the prediction of a model to its input features by examining how the prediction changes when inputs are interpolated between a baseline and the actual input. For this study, it was deployed using the Alibi library, which facilitates the integration of this method with TensorFlow and Keras models.

Implementation of Integrated Gradients

Configuration Parameters

The implementation involved configuring the IG explainer with the following parameters:

- **Model**: The top performing models being explained.
- **Layer**: The initial embedding layer of the model, chosen because it directly handles the input text data, transforming it into a form that subsequent layers can process. In this case that would be the Doc2Vec embedding layer and the BERT input embedding layer.
- **Number of Steps (n_steps)**: Set to 20 to balance computational efficiency with the granularity of the approximation. This parameter dictates the number of steps in the path integral approximation from the baseline to the input.
- **Method**: '*gausslegendre*' was selected for integrating gradients, a method known for its efficiency in numerical integration using Gaussian quadrature.
- **Internal Batch Size**: Set to the last 100 tokens to manage memory usage during the computation of gradients, especially beneficial when handling large input datasets or complex models. This parameter is critical, as it determines the number of samples processed simultaneously when computing gradients. While larger batch sizes can theoretically provide a more stable gradient estimation by reducing variance, they also

significantly increase computational demands. The limitation on computational resources necessitated this choice, as the complete processing of entire transcripts at once would require extensive memory and processing power, potentially making the execution impractical on available systems. Consequently, the chosen batch size represents a compromise that balances the need for accurate gradient computation with the practical limitations of hardware resources, ensuring that the explanatory analysis remains computationally feasible while still yielding meaningful insights into the model's decision-making process.

SECTION V: Results

This section of the thesis provides a quantitative assessment of the distinct approaches implemented in the study, focusing on the performance of both machine learning and deep learning methods applied to transcripts from the DementiaBank database. By systematically evaluating these methods, the study not only highlights the strengths and limitations of each approach but also offers valuable insights into how these models process and analyze complex clinical data. Additionally, this section delves into evaluating the importance of different features, which elucidates their contributions and influence on the predictive power and interpretability of the models. The insights derived from this analysis are instrumental in understanding the underlying dynamics of the models, which in turn can inform improvements in algorithm design and application, ultimately aiding in the development of more accurate and robust tools for dementia diagnosis.

Each model was evaluated using five scoring methods. These indicators included: Testing Precision, Accuracy, F1-Score, Recall, and area under the ROC curve. These were applied to all of the classification tasks in this thesis. The machine learning methods additionally employed the metric log loss. The subsequent section evaluates the model's ability to measure classifying AD or Control. In the evaluation of top-performing models for the classification of AD, particular emphasis was placed on the precision metric due to the critical nature of accurately identifying potential dementia cases in clinical settings. Higher precision minimizes the risk of false positives, which is essential in medical diagnostics where the cost of misdiagnosis can be high both in terms of patient care and subsequent medical treatment. Moreover, additional emphasis was put on the precision scores, as they were used to evaluate the effectiveness of the given model, and then used as a criterion for selecting the top performing models for feature importance.

5.1: Machine Learning Model Evaluation

The Random Forest model using only bigrams (figure four) showed the best overall precision at 83%, with a particularly high precision of 90% for dementia cases, underscoring its strength in correctly identifying true positive cases of dementia. The accuracy stood at 84% and the f1-score was 83% with 80% for control cases and 86% for dementia cases. The focus on bigrams likely helped the model to better understand the contextual dependencies in the speech patterns of dementia patients, which are crucial for accurate classification.

Models Evaluated with Bigrams

Model	Precision	Recall	F1-score	Accuracy
Logistic regression	0.76	0.74	0 .74	0.75
KNN	0.66	0.66	0.66	0.66
Decision Tree	0.66	0.66	0.66	0.66
MN Bayes	0.71	0.71	0.71	0.71
Random Forest	**0.83**	0.84	.83	**0.84**
AdaBoost	0.65	0.64	0.64	0.65

The Random Forest model utilizing both unigrams and bigrams demonstrated strong performance, achieving an accuracy of 81% with a precision of 82% for control cases and 79% for dementia cases. This model exhibited a solid balance between precision and recall, indicating its effectiveness in classifying dementia with a high degree of reliability.

The Logistic Regression model combining unigrams and bigrams also showed robust performance, with a precision of approximately 82.58% across both classes. The model achieved an overall accuracy of 81.82% and a macro-average F1 score of 81%, reflecting a strong balance between precision and recall. In contrast, the Logistic Regression model employing bigrams achieved an accuracy of 75.45% and a precision of 76.05%. These results illustrate that Logistic Regression, while simpler in its approach compared to more complex models, remains an effective tool for capturing linguistic patterns indicative of cognitive decline.

Models Evaluated with Both Unigrams and Bigrams

Model	Precision	Recall	F1-Score	Accuracy
Logistic Regression	**0.82**	0.81	0.81	**0.81**
KNN	0.77	0.77	0.77	0.77
Decision Tree	0.71	0.71	0.71	0.72
MN Bayes	0.78	0.74	0.74	0.74
Random Forest	**0.82**	.80	.81	**.80**
AdaBoost	0.65	0.65	0.65	0.65

As noted in figures four and five, the Multinomial Naive Bayes classifier for bigrams demonstrated a commendable effectiveness in classifying AD from transcript data, achieving an accuracy of 71.81% and a precision of 71.28%. While these metrics indicate a strong capability to identify true AD cases, they are slightly lower compared to the model's performance with unigrams and bigrams, which attained an accuracy of 75.45% and a precision of approximately

78.91%. The recall for the bigram model was notably consistent, with a balanced performance between precision and recall as indicated by an F1 score of 73.00%.

The K-Nearest Neighbors (KNN) classifier utilizing both unigrams and bigrams also performed adequately, recording an accuracy of 77.27% and a precision of 77.15%. This performance did not lag significantly behind the top-performing models, demonstrating its viability as a reliable alternative for text classification in dementia contexts.

Conversely, the worst-performing model in this analysis was the AdaBoost classifier, which registered a lower accuracy of 65% for both unigrams and bigrams. This underperformance could be attributed to AdaBoost's sensitivity to noisy data and outliers, which are prevalent in natural language processing tasks. AdaBoost's algorithmic design, which focuses on increasing the weight of misclassified instances, may not be ideally suited for text classification where semantic nuances and contextual dependencies are crucial. This sensitivity might lead to an overemphasis on hard-to-classify examples, detracting from the model's overall ability to generalize from textual data.

The top performing models had fairly high scoring precision metrics, with the Random Forest model using bigrams only slightly outperforming the others in terms of precision for AD classification. This suggests that the granularity provided by bigrams is particularly useful in capturing the linguistic complexities associated with dementia speech.

5.2: Deep Learning Model Evaluation

The comparative analysis of the scoring metrics for each model is presented in Table six. Examination of these metrics reveals that the Bi-LSTM+Doc2Vec model achieved superior performance, registering the highest Precision score at 90% and an Accuracy score of 89%. Following closely, the CNN-LSTM+Doc2Vec model demonstrated robust results with a Precision score of 89% and an Accuracy of 87%. Subsequent evaluations show that the CNN-LSTM+BERT and LSTM+Doc2Vec models also performed commendably, with precision scores of 86% and 84% respectively. In contrast, the LSTM+Word2Vec model exhibited the least effectiveness, achieving a Precision of 28% and an Accuracy of 55%, thereby ranking as the lowest-performing model in this study.

Models	Precision	Recall	F1-Score	Accuracy
LSTM+GloVe	.79	.79	.82	.79
Bi-LSTM+GloVe	. 72	.71	.76	.72
CNN-LSTM+GloVe	.82	.77	.83	.79
LSTM+Word2Vec	.78	.52	.57	.72
Bi-LSTM+Word2Vec	.61	.51	.71	.56
CNN-LSTM+Word2Vec	0.28	0.50	0.36	0.55

LSTM+Doc2Vec	**.84**	.84	.85	.84
Bi-LSTM+Doc2Vec	**.90**	**.90**	**.89**	**.89**
CNN-LSTM+Doc2Vec	**.89**	**.86**	**.90**	**.87**
LSTM+BERT	. 75	.70	.79	.73
Bi-LSTM+BERT	.74	.61	.75	.65
CNN-LSTM+BERT	**.86**	.85	.88	**.85**

The models' loss and accuracy information were stored in the object history for each epoch. Training and validation accuracy was plotted across the number of epochs in the training process. Figure Eight, found in the appendix, shows Training vs Testing accuracy and loss for the Doc2Vec LSTM, Bi-LSTM and CNN-LSTM.

The analysis of model loss and accuracy graphs plays a crucial role in the assessment and optimization of machine learning models, particularly in deep learning applications where the complexity of models often obscures their functioning. These graphs provide a visual representation of the training and validation processes, offering critical insights into the model's performance across different stages of the learning process.

In the context of this study, the training loss versus testing loss over epochs for the Doc2Vec LSTM, Bi-LSTM, and CNN-LSTM models, as depicted in Figure 10, is essential for evaluating model fit and the generalization ability. The loss function, calculated as a quantitative measure of error between the predicted outputs and actual outputs, is computed over all data items throughout each epoch. By monitoring this loss across epochs, we can observe how well the model learns the dataset's patterns over time.

A key observation from these graphs is the divergence between training loss and testing loss starting around epoch 20. This widening gap is indicative of overfitting—a scenario where the model learns the details and noise in the training data to an extent that it negatively impacts the performance on new data. Essentially, while the training loss continues to decrease, suggesting better performance on the training set, the validation loss begins to increase, reflecting a decline in the model's ability to generalize to unseen data.

Optimization for validation loss is crucial as it helps ensure that the model adjustments enhance its performance on externally valid data, not just the training set. Which was why all deep learning were optimized for validation loss. This approach helps in fine-tuning the model to achieve a balance between underfitting and overfitting, striving for a model that generalizes well on new unseen data, while performing adequately on the training set.

5.3 Feature Importance

This section presents results of the feature importance for the machine learning and deep learning dementia text classifiers. The presentation of the LIME and Integrated Gradients methods are facilitated by visual examples which can be created for individual observations in the data. Unfortunately, not all 500 transcripts can be displayed to the reader in this paper and so a select few are included that neatly illustrate the most important findings. Where possible, the same transcripts are compared from one method and model to another.

Machine Learning Feature Importance

SHAP

In the context of this thesis, the summary plot for Random Forest with Bigrams, which can be found in figure four, is utilized to illustrate the contribution of different bigrams to the predictions made by a Random Forest model trained on the DementiaBank transcripts. The plot highlights the features (bi-grams) that have the most substantial impact on the model’s output. Bigrams such as *'mother drying'*, *'window open'*, and *'girl reaching*', ‘*reaching cookie’*, and ‘*mother washing’* exhibited a high correlation with the control class (non-dementia). This implies that the frequent occurrence of these bigrams is associated with instances more likely to be classified as belonging to the control. This overlaps with the features found in the random forest with unigrams and bigrams model, which consisted of overlapping features like, ‘*mother drying’, ‘window’, ‘action, ‘open’, ‘mother’,* and *‘reaching cookie’,* and can be found illustrated in Figure five. Such bi-grams may occur in more structured and logical narratives typical of individuals without cognitive impairments. These phrases and words may represent coherent activities or scenarios that are less typical in the disjointed narratives of Alzheimer's patients.

In contrast, the random forest model with bigrams categorizes *'cookie jar'*, which showed a high importance with the AD class, and ‘*gonna fall*’ which showed moderate importance suggest a higher association with the AD group as important features for AD detection. It is particularly interesting that within the random forest with unigrams and bigrams model the features ‘*jar’, ‘uh’,* and *‘oh’* are the features ranked with the highest importance for classifying a transcript as AD. The linguistic markers *‘uh’* and *‘oh’* are filled pauses, and can be indicative of cognitive impairments associated with the progression of dementia. These expressions, typically employed in speech to signal hesitation or a search for the correct terminology, can become notably prevalent as the neurological impact of Alzheimer’s disrupts the patient's linguistic abilities. From a cognitive perspective, AD characteristically leads to difficulties in retrieving words and forming coherent thoughts, processes that are heavily influenced by degeneration in temporal and parietal lobes of the brain—regions critical for language and memory. Consequently, affected individuals may rely increasingly on nonspecific filler words or pauses as placeholders during moments of lexical retrieval failure.

SHAP for Random Forest with Bigrams

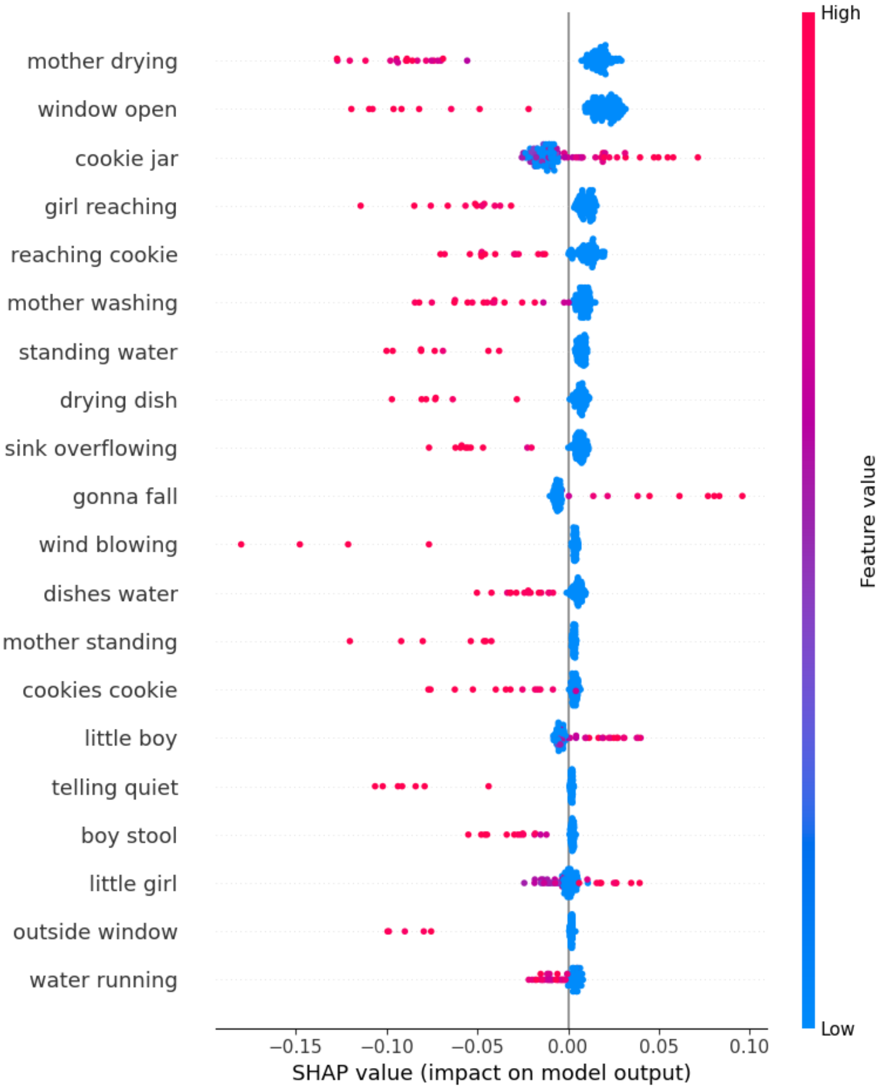


Figure 4

SHAP For Random Forest with Unigrams and Bigrams

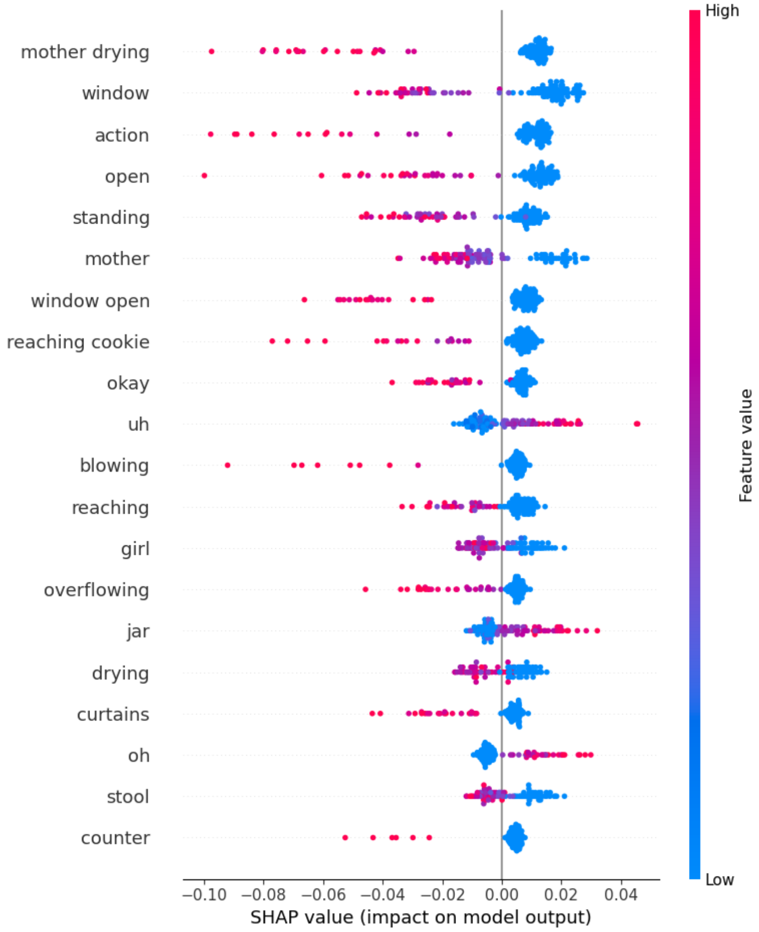


Figure 5

PERMUTATION IMPORTANCE

The sorted feature importance, particularly visualized through a horizontal bar chart found in figure six, not only quantifies the influence of each feature but also incorporates error bars representing the standard deviation of importance scores across repetitions. This visualization highlights the uncertainty and variability in the importance estimates, providing a clearer and more nuanced understanding of each feature's role.

Permutation Importance for Logistic Regression with Unigrams and Bigrams

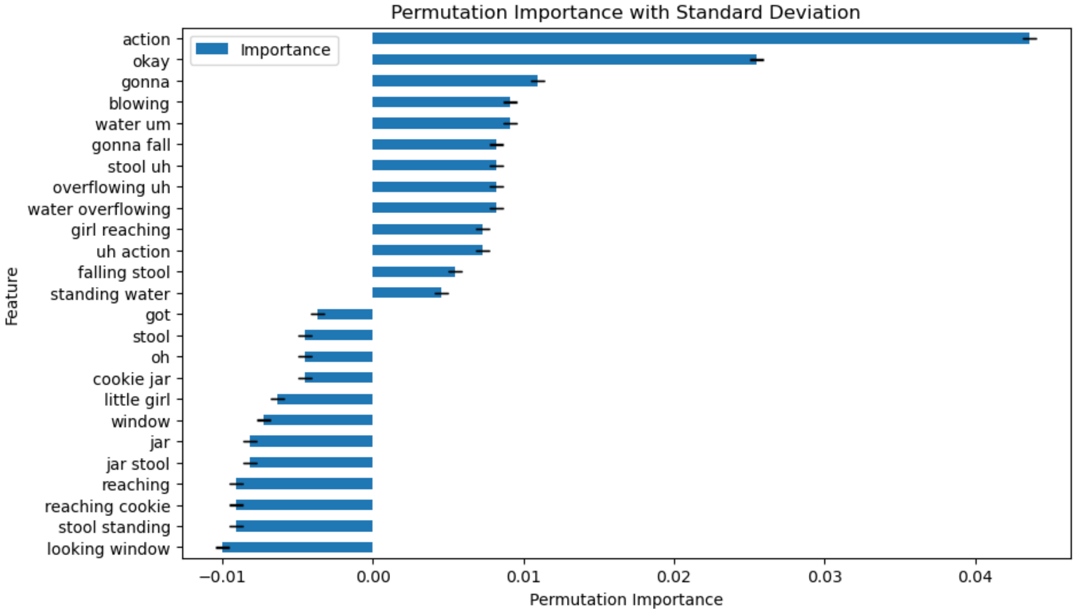


Figure 6

The results obtained from the permutation importance analysis for the Logistic Regression model complement and reinforce the insights gathered from SHAP summary plots for the Random Forest models. For instance, certain linguistic markers that were identified as significant in the SHAP plots, such as ‘*action*’, ‘window’, ‘*ok*’, and ‘*standing*’, also showed high importance in the Logistic Regression analysis, affirming their predictive power. This alignment between the two methods not only strengthens the confidence in the models' assessments but also underscores the consistency of the identified linguistic features across different analytical approaches. By providing a converging validation from both permutation importance and SHAP values, the results reinforce the robustness of the feature selection process and enhance the interpretability of the predictive models used in this study.

Deep Learning Feature Importance

LIME

LIME is particularly beneficial. Text data is inherently high-dimensional and sparse, making traditional feature importance techniques less effective. LIME's ability to provide explanations for individual predictions by highlighting words or phrases that influence the model’s output offers clear, actionable insights into how language use correlates with dementia diagnoses. This is critical in ensuring that the models used are not only accurate but also clinically meaningful.

The LIME explanation framework is initiated specifically for text data by importing *LimeTextExplainer* from the LIME library. This setup helps tailor the LIME output to be intuitive and directly applicable to the task at hand, facilitating easier interpretation. A crucial step in the LIME process is defining a prediction function that the explainer can use to simulate how the model behaves with modified inputs. The *predict_proba* function is defined to take a list of transcripts, preprocess it by tokenizing and padding to fit the model's input requirements, and output the probability predictions for each class. This function uses the trained deep learning model to predict how likely each transcript belongs to the control or dementia The *explain_instance* method is called for a specific instance and the prediction function. It returns an explanation by predicting repeatedly with modified inputs, which features (words or phrases) in the transcript were most influential in the model's prediction and the direction of their impact (increasing or decreasing the probability of belonging to a particular class).

The subsequent figures provide a visual representation of the outputs generated by the Local Interpretable Model-agnostic Explanations (LIME) text explainer for several instances across the top-performing models in this study, specifically the Doc2Vec LSTM, Bi-LSTM, CNN-LSTM, and BERT CNN-LSTM models. These figures are instrumental in understanding the interpretative capacity of the models, showcasing how each model weights and prioritizes different features or phrases within the transcripts to arrive at a diagnostic classification.

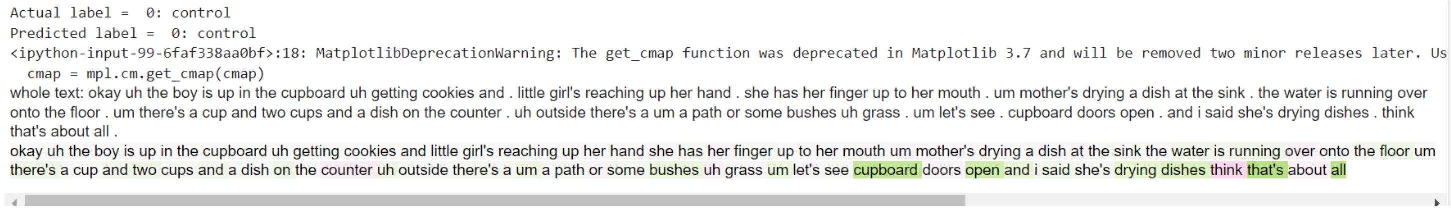
The application of LIME is used to analyze the classification of transcripts from top-performing models—Doc2Vec LSTM, Bi-LSTM, CNN-LSTM, and BERT CNN-LSTM— to provide critical insights into the linguistic patterns associated with AD and control groups. LIME's utility lies in its ability to visually and quantitatively highlight the influence of specific words or phrases on the models' classification decisions, assigning colors to tokens where blue indicates a correlation with the control group and orange signifies a connection with the AD class. The intensity of the color corresponds to the degree of influence, providing a clear indication of feature importance at a local level. The LIME text explainer results offer a nuanced understanding of how certain tokens contribute to the classification accuracy and reliability of the models

The application of LIME to analyze transcripts from Alzheimer's Disease (AD) patients and control groups has provided nuanced insights into the linguistic and speech patterns characteristic of each group. This analysis is particularly important for understanding the cognitive impairments associated with AD and the relative clarity observed in the control group.

Integrated Gradients

The subsequent figure provides a visual representation of the Integrated Gradients method applied to selected examples from this paper. These illustrations encapsulate the attributions of individual features within the last 100 tokens of each transcript (limited due to computational constraints), highlighting the impact of specific words on the model's predictions. Each example within the figure has been meticulously chosen to demonstrate how Integrated Gradients highlights the contributions of various elements of the input data towards the predictive outcomes. The visualization serves as an integral component of this thesis, offering a clear and concise depiction of the model's interpretive processes, and fostering a deeper understanding of the underlying mechanisms driving the predictions.

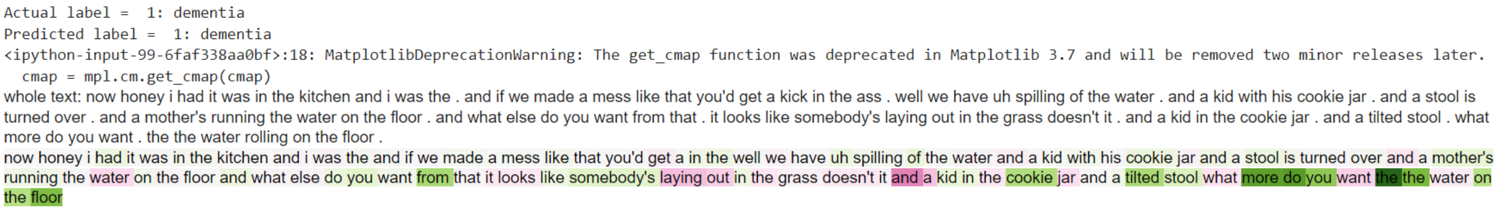
Intgrated Gradients Transcripts



```
Actual label =  0: control
Predicted label =  0: control
<ipython-input-99-6faf338aa0bf>:18: MatplotlibDeprecationWarning: The get_cmap function was deprecated in Matplotlib 3.7 and will be removed two minor releases later. Us
  cmap = mpl.cm.get_cmap(cmap)
```

whole text: okay uh the boy is up in the cupboard uh getting cookies and . little girl's reaching up her hand . she has her finger up to her mouth . um mother's drying a dish at the sink . the water is running over onto the floor . um there's a cup and two cups and a dish on the counter . uh outside there's a um a path or some bushes uh grass . um let's see . cupboard doors open . and i said she's drying dishes . think that's about all .

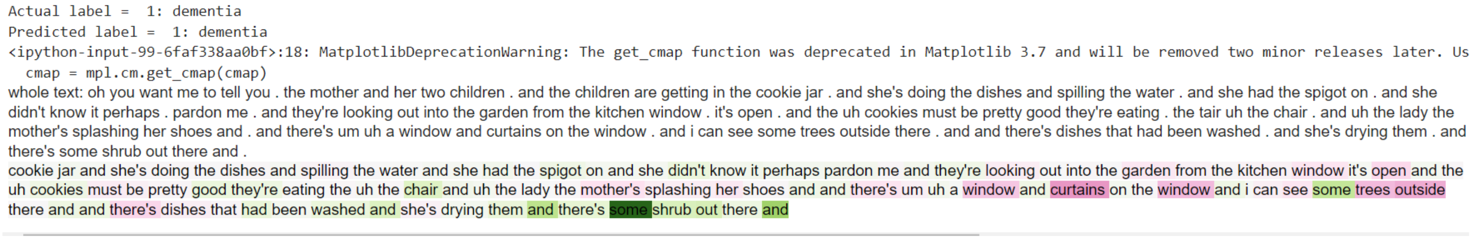
okay uh the boy is up in the cupboard uh getting cookies and little girl's reaching up her hand she has her finger up to her mouth um mother's drying a dish at the sink the water is running over onto the floor um there's a cup and two cups and a dish on the counter uh outside there's a um a path or some bushes uh grass um let's see cupboard doors open and i said she's drying dishes think that's about all



```
Actual label =  1: dementia
Predicted label =  1: dementia
<ipython-input-99-6faf338aa0bf>:18: MatplotlibDeprecationWarning: The get_cmap function was deprecated in Matplotlib 3.7 and will be removed two minor releases later.
  cmap = mpl.cm.get_cmap(cmap)
```

whole text: now honey i had it was in the kitchen and i was the . and if we made a mess like that you'd get a kick in the ass . well we have uh spilling of the water . and a kid with his cookie jar . and a stool is turned over . and a mother's running the water on the floor . and what else do you want from that . it looks like somebody's laying out in the grass doesn't it . and a kid in the cookie jar . and a tilted stool . what more do you want . the the water rolling on the floor .

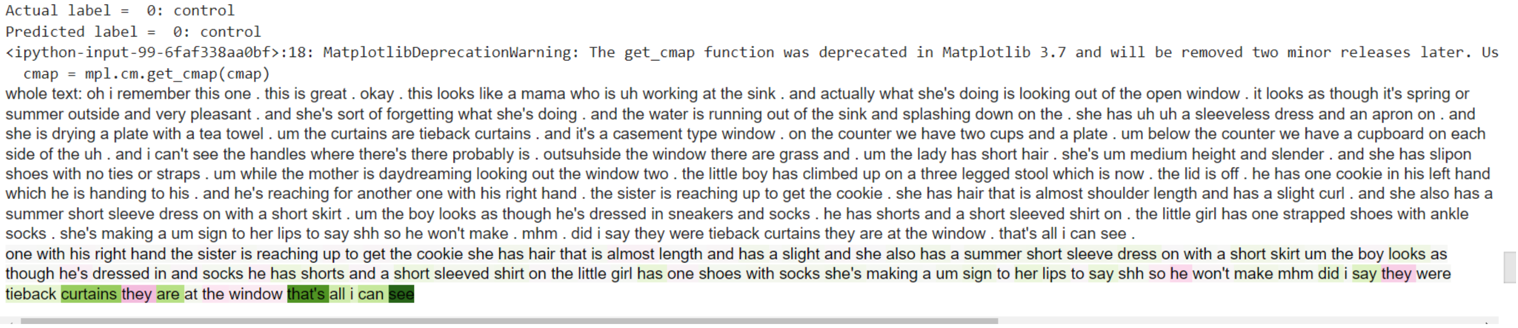
now honey i had it was in the kitchen and i was the and if we made a mess like that you'd get a in the well we have uh spilling of the water and a kid with his cookie jar and a stool is turned over and a mother's running the water on the floor and what else do you want from that it looks like somebody's laying out in the grass doesn't it and a kid in the cookie jar and a tilted stool what more do you want the the water on the floor



```
Actual label =  1: dementia
Predicted label =  1: dementia
<ipython-input-99-6faf338aa0bf>:18: MatplotlibDeprecationWarning: The get_cmap function was deprecated in Matplotlib 3.7 and will be removed two minor releases later. Us
  cmap = mpl.cm.get_cmap(cmap)
```

whole text: oh you want me to tell you . the mother and her two children . and the children are getting in the cookie jar . and she's doing the dishes and spilling the water . and she had the spigot on . and she didn't know it perhaps . pardon me . and they're looking out into the garden from the kitchen window . it's open . and the uh cookies must be pretty good they're eating . the tair uh the chair . and uh the lady the mother's splashing her shoes and . and there's um uh a window and curtains on the window . and i can see some trees outside there . and and there's dishes that had been washed . and she's drying them . and there's some shrub out there and .

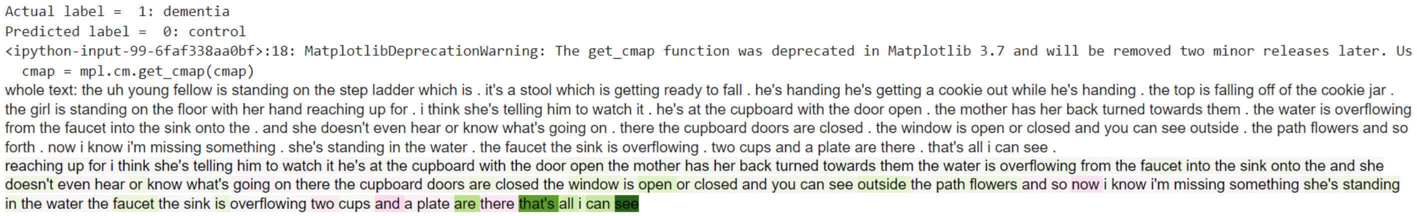
cookie jar and she's doing the dishes and spilling the water and she had the spigot on and she didn't know it perhaps pardon me and they're looking out into the garden from the kitchen window it's open and the uh cookies must be pretty good they're eating the uh the chair and uh the lady the mother's splashing her shoes and and there's um uh a window and curtains on the window and i can see some trees outside there and and there's dishes that had been washed and she's drying them and there's some shrub out there and



```
Actual label =  0: control
Predicted label =  0: control
<ipython-input-99-6faf338aa0bf>:18: MatplotlibDeprecationWarning: The get_cmap function was deprecated in Matplotlib 3.7 and will be removed two minor releases later. Us
  cmap = mpl.cm.get_cmap(cmap)
```

whole text: oh i remember this one . this is great . okay . this looks like a mama who is uh working at the sink . and actually what she's doing is looking out of the open window . it looks as though it's spring or summer outside and very pleasant . and she's sort of forgetting what she's doing . and the water is running out of the sink and splashing down on the . she has uh uh a sleeveless dress and an apron on . and she is drying a plate with a tea towel . um the curtains are tieback curtains . and it's a casement type window . on the counter we have two cups and a plate . um below the counter we have a cupboard on each side of the uh . and i can't see the handles where there's there probably is . outsuhside the window there are grass and . um the lady has short hair . she's um medium height and slender . and she has slipon shoes with no ties or straps . um while the mother is daydreaming looking out the window two . the little boy has climbed up on a three legged stool which is now . the lid is off . he has one cookie in his left hand which he is handing to his . and he's reaching for another one with his right hand . the sister is reaching up to get the cookie . she has hair that is almost shoulder length and has a slight curl . and she also has a summer short sleeve dress on with a short skirt . um the boy looks as though he's dressed in sneakers and socks . he has shorts and a short sleeved shirt on . the little girl has one strapped shoes with ankle socks . she's making a um sign to her lips to say shh so he won't make . mhm . did i say they were tieback curtains they are at the window . that's all i can see .

one with his right hand the sister is reaching up to get the cookie she has hair that is almost length and has a slight and she also has a summer short sleeve dress on with a short skirt um the boy looks as though he's dressed in and socks he has shorts and a short sleeved shirt on the little girl has one shoes with socks she's making a um sign to her lips to say shh so he won't make mhm did i say they were tieback curtains they are at the window that's all i can see



```
Actual label =  1: dementia
Predicted label =  0: control
<ipython-input-99-6faf338aa0bf>:18: MatplotlibDeprecationWarning: The get_cmap function was deprecated in Matplotlib 3.7 and will be removed two minor releases later. Us
  cmap = mpl.cm.get_cmap(cmap)
whole text: the uh young fellow is standing on the step ladder which is . it's a stool which is getting ready to fall . he's handing he's getting a cookie out while he's handing . the top is falling off of the cookie jar .
the girl is standing on the floor with her hand reaching up for . i think she's telling him to watch it . he's at the cupboard with the door open . the mother has her back turned towards them . the water is overflowing
from the faucet into the sink onto the . and she doesn't even hear or know what's going on . there the cupboard doors are closed . the window is open or closed and you can see outside . the path flowers and so
forth . now i know i'm missing something . she's standing in the water . the faucet the sink is overflowing . two cups and a plate are there . that's all i can see .
reaching up for i think she's telling him to watch it he's at the cupboard with the door open the mother has her back turned towards them the water is overflowing from the faucet into the sink onto the and she
doesn't even hear or know what's going on there the cupboard doors are closed the window is open or closed and you can see outside the path flowers and so now i know i'm missing something she's standing
in the water the faucet the sink is overflowing two cups and a plate are there that's all i can see
```

In the analysis of Integrated Gradients for the feature importance section, the significance of words within the transcripts is visually highlighted to discern their impact on model predictions. The color-coding scheme employed provides an intuitive understanding of each word's contribution: words colored greener denote a higher positive attribution toward the predicted class (AD), while those in pink suggest a negative attribution. This method of visualization is particularly effective in distinguishing the salient features that drive the model's decisions.

Despite the methodological constraint of analyzing only 100 tokens per transcript, notable commonalities are observed between features associated with the control and AD groups across both integrated gradients and LIME. Such overlaps in linguistic features are illustrative of the complex nature of language processing in the context of dementia classification. Common tokens such as 'the', 'from', 'cookie jar', 'and', 'some', and 'for' appear across transcripts from both AD groups, suggesting that these words are central to helping identify dementia the descriptions given in the Cookie Theft picture description task.

For the control class, specific phrases like *'a sign to her lips'*, *'that's all I can see'*, '*cupboard*', *'drying dishes'*, '*curtains*', '*window*', and *'trees outside'* are recurrent. These phrases generally depict more structured and detailed observations of the scene, which might imply a higher level of cognitive coherence and linguistic organization typical of control subjects. This observation aligns with linguistic research suggesting that individuals without cognitive impairment tend to provide more detailed and contextually rich descriptions.

The overlaps in tokens between the two classes may be occurring due to the universal nature of the task, where subjects from both groups are asked to describe the same picture. As a result, certain items or actions within the picture (like 'cookie jar' or 'drying dishes') are likely to be mentioned by different subsets of subjects and seems to be correlated with their cognitive status. However, the way these items are described—the detail, coherence, and context provided—can differ significantly, which is why deeper linguistic and semantic analysis beyond surface tokens is crucial for more accurate classification. This exemplifies the importance of employing sophisticated NLP tools and techniques that can capture not just the presence of words, but their contextual relevance and integration into coherent narratives, which are often more telling of a subject's cognitive state.

Discussion

Deep Learning Models

The notable performance of the BERT input embeddings combined with the CNN-LSTM architecture, as demonstrated by a precision of 86%, an accuracy of 85%, recall of 85% and an

F1-score of 88%, demonstrates the synergetic potential of integrating advanced embedding techniques with sophisticated neural network structures. This model configuration leverages the strengths of BERT's deep contextual embeddings and the dynamic processing capabilities of the CNN-LSTM framework, making it particularly adept at handling complex language data inherent in clinical diagnostics such as AD detection.

Performance of the CNN-LSTM+BERT can be attributed to the unique architecture of the model. Unlike traditional word embeddings that generate a single word embedding for each token in the vocabulary, BERT considers the full context of a word by looking at the words that come before and after it in a sentence. Moreover, the convolutional layers act as feature extractors that identify and encode local patterns within the input data. These features might include key phrases or syntax patterns in text that are indicative of cognitive issues. Convolution operations apply filters to the input data, which can be represented mathematically as:

$$f(x) = (x * w) + b$$

where $*$ denotes the convolution operation, x is the input, w represents the weights of the filter, and b is the bias.

The integration of BERT embeddings provides the CNN-LSTM architecture with a rich, pre-trained contextual basis from which to build its feature analyses, thus enhancing the model's ability to discern subtle linguistic cues linked to AD. This model's high F1-score suggests an effective balance between precision and recall, indicating not only its accuracy in identifying relevant cases but also its reliability in minimizing false positives and negatives—an essential attribute in medical applications.

The models that exhibited the least effectiveness were those utilizing Word2Vec embeddings within the CNN-LSTM and Bi-LSTM architectures, with the machine learning models outperforming them. Specifically, the CNN-LSTM with Word2Vec embeddings demonstrated a notably low precision score of 28% and an accuracy of 58%, while the Bi-LSTM with Word2Vec achieved a precision of 61% and an accuracy of 56%. Several factors might contribute to these underwhelming performances, which stand in stark contrast to the more successful models employing different embeddings. This could be attributed to inaccurate contextual representation. Word2Vec, as a method of generating embeddings, focuses primarily on capturing local contextual meanings within a fixed window size around each word, unlike BERT or Doc2Vec that account for broader or entire document context, Word2Vec may not fully capture the necessary contextual clues that indicate cognitive impairments in Alzheimer's Disease, which often manifest in more complex and subtle linguistic patterns beyond local word pairs. The embeddings are also static, meaning that each word is represented by a single embedding, regardless of its varying meanings in different contexts. These static embeddings might be insufficient for the deep learning models, particularly CNN-LSTMs and Bi-LSTMs, which require dynamic and rich input features to effectively model the intricacies of natural language as it pertains to AD symptoms.

The poor performance may also be ascribed to mismatched architecture. The architecture of CNN-LSTMs, which are designed to capitalize on both local feature extraction (through CNN layers) and sequential data processing (through LSTM layers), might not align well with the nature of Word2Vec embeddings. The simplistic and local context focus of Word2Vec may not provide enough depth and variation needed for the CNN layers to extract meaningful features, nor for the LSTM layers to capture long-term dependencies effectively. Further, both CNN-LSTM and Bi-LSTM models with Word2Vec may also suffer from overfitting or underfitting, particularly if the model training did not adequately account for the nuances and variability within the AD-related data.

Of the traditional machine learning models, Random Forest with bigrams and unigrams, and Logistic Regression with unigrams and bigrams, demonstrated competitive, and at times superior, performance compared to more complex deep learning models such as the Word2Vec CNN-LSTM and Bi-LSTM. This observation can be explained by a common heuristic in the field of machine learning regarding the relationship between dataset size and the efficacy of model types.

Deep learning models are renowned for their high capacity and flexibility, enabling them to model complex patterns in large datasets. However, they require substantial amounts of data to generalize effectively without overfitting. According to a widely accepted rule of thumb in machine learning literature, deep learning models typically begin to outperform simpler machine learning models only when the available dataset includes thousands to tens of thousands of samples. Conversely, when the dataset is smaller, as is often the case in specialized medical research studies like AD classification using linguistic data, traditional machine learning models tend to be more effective. These models are less prone to overfitting and require fewer data to reach a reliable level of generalization, making them particularly suitable for studies with datasets comprising fewer than a few thousand examples.

In the context of this study, the relatively smaller dataset size likely contributed to the robust performance of the machine learning models. These models, with their simpler structures, could effectively capture the essential patterns in the data without the extensive data requirements and computational complexity associated with deep learning models. Thus, the Random Forest and Logistic Regression models were able to achieve high accuracy and precision scores, closely matching or even surpassing some of the deep learning approaches, especially in cases where the latter might have struggled with overfitting or insufficient data to learn from effectively. This efficacy denotes the importance of choosing the appropriate modeling technique based on the dataset characteristics and the specific requirements of the study, thereby ensuring optimal performance and utilization of resources.

Surprisingly, the top-performing deep learning models in this analysis consistently utilized Doc2Vec embeddings, particularly leveraging the Distributed Bag of Words (PV-DBOW) variant of Doc2Vec, as the primary embedding layer. This notable pattern suggests a strong correlation between the use of Doc2Vec embeddings and enhanced model performance across different architectures including LSTM, Bi-LSTM, and CNN-LSTM. Two key insights can be inferred from this observation:

Enhanced Semantic Capture: Doc2Vec, especially the PV-DBOW model, excels in capturing the overarching semantic context of the entire document without the computational burden of sequential dependency. This broader understanding likely contributes to the model's ability to discern patterns and nuances in Alzheimer's Disease-related transcripts that are crucial for accurate classification. By training word vectors alongside vector representations of entire documents, Doc2Vec may provide a more comprehensive feature set for the deep learning models, enhancing their predictive capabilities.

Generalization Across Architectures: The success of Doc2Vec embeddings across various deep learning architectures indicates their robustness and adaptability. Unlike other embedding techniques that might favor specific neural configurations, the generic nature of Doc2Vec embeddings appears to complement and enhance the inherent strengths of LSTM, Bi-LSTM, and CNN-LSTM models. Each of these architectures benefits differently from the embeddings—LSTMs leverage temporal patterns, Bi-LSTMs utilize bidirectional context, and CNN-LSTMs capitalize on localized feature extraction followed by sequence modeling.

This analysis stands in direct contrast to findings from other studies, such as the *Comparative Study of Deep Classifiers for Early Dementia Detection using Speech Transcripts* (2), which reported lower metrics for Doc2Vec-enhanced models. In (2) LSTM+Doc2Vec achieved 76% precision and 71% accuracy. Similarly, Bi-LSTM+Doc2Vec had a 71% precision and 69% accuracy. The discrepancy could be attributed to several factors, including differences in dataset preprocessing, model parameterization, and perhaps the integration technique of embeddings within the neural networks. The superior performance in this study may also emphasize the effectiveness of the PV-DBOW architecture in handling diverse and complex datasets like those involved in AD detection, suggesting that the way embeddings are implemented and utilized can significantly influence outcome metrics.

Feature Importance: SHAP

The paper titled *"Classification of Alzheimer's Disease Leveraging Multi-task Machine Learning Analysis of Speech and Eye-Movement Data*" explores a multi-modal machine learning approach to diagnose Alzheimer's Disease (AD) using novel speech and eye-movement tasks (Jang et al., 2021). Jang utilizes the Cookie Theft image description task as part of its multimodal approach to analyze speech and eye-movement data for AD classification[20]. This task is a traditional component of the Boston Diagnostic Aphasia Examination and is commonly used in cognitive assessments for dementia. The Cookie Theft description task contributed to the multimodal dataset that achieved high classification accuracies in distinguishing between AD patients and healthy controls. It provided critical speech and visual engagement data that, when combined with other tasks, enhanced the predictive capabilities of the machine learning models used. The image was split up into areas of interest that correlated with language features such as: cookie, cookie jar, boy, girl, woman, stool, plate, dishcloth, water, window, curtain, dishes, and sink[21]. The AoI's used in (Barral et al., 2021) can be found in the subsequent figure.

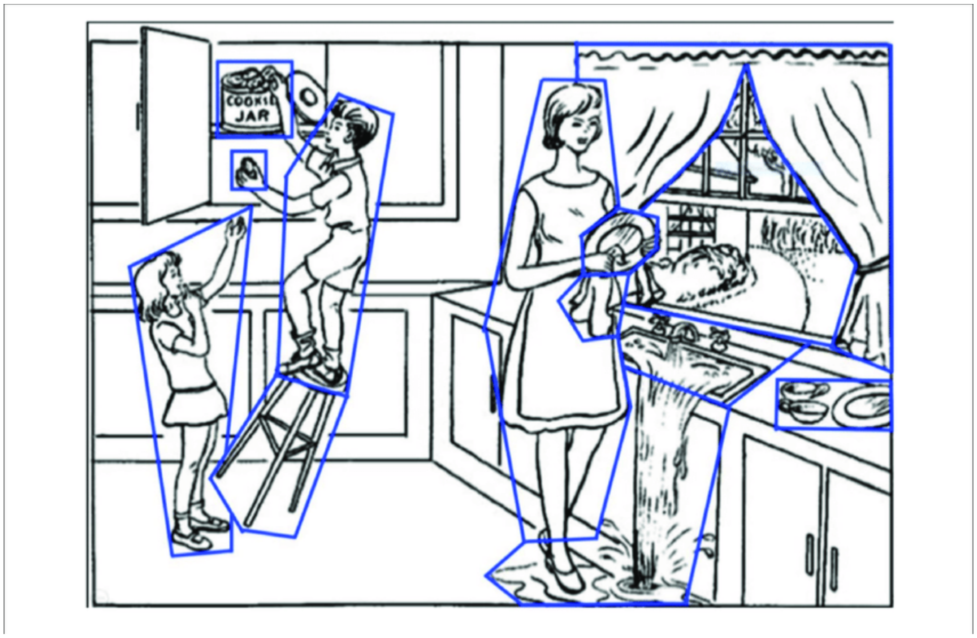


Figure 7: Cookie Theft picture featuring AOIs, adapted from Jang et al., 2021

They found AD patients tended to have shorter fixations, eye movements, and more variation in their eye movements during the task. (Barral et al., 2021), the foundational work of (20), also found AD patients have a higher proportion of visits to the cookie jar. In contrast, controls show more transitions within the window. Healthy controls also have more transitions to the girl (sink to girl, and cookie to girl). Which directly overlaps with the linguistic commonalities found in this thesis between control and AD transcripts.

In the context of the cookie theft image description task, it is possible to conclude that bigrams and unigrams like *'mother drying'*, *'window open'*, *'window', 'action, 'open', 'girl reaching'* and *'mother'* often found in control transcripts could be attributed to control participants being better able to identify areas of importance (AoI).

Feature Importance: LIME

Overlapping Linguistic Features in Dementia Transcripts

In dementia transcripts, certain tokens such as 'and', 'uh', and 'jar' frequently appear. These tokens reveal significant linguistic markers:

- '*And*' is often used repetitively, indicating a difficulty in forming cohesive and complex sentences. Its repeated use is typically seen in run-on sentences, which may signify an attempt by AD patients to hold onto a train of thought or to mask difficulties with word retrieval.
- '*Uh*' reflects moments of hesitation or uncertainty, commonly observed in spontaneous speech as AD patients struggle to find the correct words.

- '*Jar*', a specific noun, repeatedly occurs in the Cookie Theft Picture Description Task, pointing towards a narrowed focus on particular elements within a picture, possibly due to the patient's reduced ability to interpret the overall scene.

These findings align with the broader linguistic patterns noted where AD patients tend to use more interjections such as '*oh*', '*yeah*', and '*well*', and adverbs like *'maybe'* and '*here*'. This usage may compensate for their uncertainty and cognitive impairments, reflecting an underlying struggle with articulating definitive statements.

Overlapping Linguistic Features in Control Transcripts

Conversely, control transcripts often contain more structured and diverse vocabulary. Words like 'window', 'curtains', 'dish', 'mother', 'the', and 'say', not only indicate a higher level of descriptive detail but also demonstrate the ability to engage with and describe the environment more accurately. These words align with the control class features identified in SHAP summary plots and eye tracking findings in (Jang et al., 2021) and (Barral et al., 2021).

An interesting linguistic feature in the control group is the frequent correct usage of the apostrophe 's'. This grammatical element suggests proficiency with the possessive case, which requires cognitive capabilities that might be compromised in AD patients. The clarity of the 'S' sound is important in speech articulation. Proper pronunciation aids in the clarity of speech, and its presence in many common words means that any mispronunciation could lead to misunderstandings, which are less frequent in the control group but a potential issue in AD due to symptoms like slurred speech and stammering.

Parts of speech (POS) in these analyses helps further differentiate and analyze the grammatical tendencies between AD patients and controls. Control transcripts tended to weigh the token 'the' highly. The frequent identification of the definite article 'the' as a significant token merits further discussion. The definite article 'the' is employed in English to refer to specific, known entities or nouns within a discourse context. Its usage indicates that the speaker assumes the listener knows what is being referred to, without needing to specify it explicitly. This linguistic feature is essential in analyzing transcripts involving tasks such as the Cookie Theft Image Description, where clarity and specificity in describing known elements of the picture are indicative of cognitive health.

Transcripts from control participants that heavily utilized '*the*' likely demonstrated greater coherence and connectivity in narrative construction. These participants were more adept at identifying and articulating specific elements within the image, suggesting a more organized and focused cognitive approach to the task. For detailed visualizations, please refer to Appendix.

In contrast, less specific or indefinite articles, which refer to unspecific nouns, might be more prevalent in the speech patterns of individuals experiencing cognitive decline, as seen in Alzheimer's Disease patients.

Therefore, the heavy association of '*the*' with control transcripts by LIME could be indicative of the participants' ability to maintain clear and coherent speech, effectively employing definite articles to construct well-defined and logical narratives about the image. This linguistic capability reflects a higher level of cognitive functioning, where the speaker can navigate and describe their environment accurately, a skill that is often compromised in the progressive stages of dementia.

For instance, the higher usage of nouns and adjectives in control transcripts versus the frequent use of pronouns and conjunctions in AD transcripts could indicate a decline in the ability to name objects or describe them with adjectives, a common symptom in Alzheimer's linguistic degradation.

Limitations

This work, however, is not without its limitations. The models developed are monolingual, focusing solely on English speech transcripts, which may limit their applicability in more linguistically diverse settings. Furthermore, the models' performance is currently bound by hardware limitations, suggesting that with improved computational resources, the efficacy and efficiency of these models could be significantly enhanced.

The choice to utilize a relatively small dataset in this study represents a significant limitation that must be acknowledged, as it can constrain the generalizability and robustness of the findings. Small datasets often provide insufficient variability and volume to fully train and validate complex machine learning and deep learning models, potentially leading to overfitting or under-representation of the broader population characteristics. However, the decision to employ this specific dataset was made strategically to minimize various kinds of biases that could compromise the validity of the proposed approaches. By selecting a smaller, more controlled dataset, this study aimed to maintain higher data integrity and ensure that the findings are as accurate and reliable as possible within the given constraints. Consequently, this methodological choice supports a more focused exploration of the specific features and models under consideration, facilitating a cleaner analysis of the data and contributing to the credibility of the research outcomes

It is essential to contextualize the findings within the broader spectrum of AD classification, linguistic and speech analysis, part-of-speech tagging, and non-verbal behavioral analysis such as eye movement tracking. This paper contributes to the burgeoning field of explainable AI, particularly within the realm of linguistics-based classification of neurodegenerative diseases. By shedding light on how deep learning models process and classify linguistic data in the context of AD, this work takes a significant step towards demystifying the decisions made by AI systems and advances XAI. This endeavor not only advances the technical capabilities of AI models but also enhances their societal acceptance and ethical deployment in sensitive domains such as healthcare diagnostics. While the results presented here align with existing research in these fields, demonstrating common linguistic patterns and keywords associated with AD and control groups, caution must be exercised in interpreting these outcomes.

The methodologies employed—SHAP, LIME, and Integrated Gradients—have provided valuable insights into the features that our models deem significant in classifying AD. However, these results should be approached with a degree of skepticism. There are numerous variables and external factors that could influence why certain words and phrases were highlighted by these explainability tools. For instance, the intrinsic biases in the training data, the limitations inherent in the models themselves, or even the subjective nature of the baseline chosen for Integrated Gradients could skew results in unforeseen ways.

Further research is crucial to validate these findings rigorously. Future studies could focus on expanding the datasets used, incorporating multimodal data, or employing more sophisticated models that can better capture the nuances of human language and cognition. Moreover, interdisciplinary approaches that combine insights from cognitive science, linguistics, and computer science could enhance the robustness and applicability of AD classification systems.

SECTION VI: Conclusion

We have explored the potential of multiple machine learning and deep learning models to detect AD from transcript data, using the DementiaBank dataset. The primary motivation behind this research is the pressing need for early detection of dementia, a condition that, while currently incurable, can have its symptoms managed more effectively with timely diagnosis.

The study employed a variety of vector embeddings and model architectures to differentiate between individuals with and without AD. These ranged from traditional machine learning models, utilizing bag-of-words approaches, to more complex sequential deep learning models that leverage the latest advancements in NLP. The comparative analysis of these models highlighted the nuanced capabilities of each approach, with deep learning models showing promise in capturing sequential and contextual information that escapes simpler, frequency-based models. Through the implementation of various machine learning architectures, and the application of advanced sequential neural networks, this study aimed to compare and contrast the efficacy of different computational approaches in identifying AD. Notably, the Random Forest model with bigrams demonstrated the highest classification performance out of the machine learning models, achieving an accuracy of 84.00% and a precision of 83.00%. Concurrently, we ventured into the domain of deep learning by deploying multiple Long Short-Term Memory (LSTM) architectures enhanced with Doc2Vec embedding layers. Among these, the Bi-LSTM coupled with Doc2Vec embeddings emerged as the strongest performing model, accurately detecting AD patients with an impressive accuracy of 89.00% and a precision of 90.00%.

A significant component of this research involved the examination of feature importance through techniques such as SHAP, LIME, and integrated gradients. This analysis is crucial, as it not only provides transparency into the decision-making processes of the models but also identifies key linguistic markers associated with AD. These insights are invaluable for clinicians and researchers alike, as they highlight potential areas for further diagnostic development, offer a deeper understanding of the linguistic impacts of dementia, and provides valuable insight into local explainability. This level of granularity in explainability is particularly valuable in clinical settings, where understanding the rationale behind a diagnostic prediction or treatment recommendation can influence clinical decisions and patient outcomes. Understanding which words or speech patterns led to this conclusion can help future research focus on specific cognitive aspects when developing models.

The analysis of feature importance within this thesis has illuminated significant linguistic patterns that distinguish between AD patients and healthy controls. Specifically, the frequent use of words such as '*and*,' '*yeah*,' and '*uh*'," and particularly '*and*' at the beginning of utterances, is notable among AD patients. This frequent usage suggests a reliance on certain connective or filler words to structure speech, which may reflect challenges in maintaining coherent and fluent discourse. In contrast, the analysis highlights that healthy individuals tend to use a richer variety of linguistic constructs, including verbs in present participle or gerund forms, nouns, and determiners. This variety denotes more robust cognitive abilities to structure complex sentences and maintain topic coherence.

This contrast in language usage not only deepens our understanding of the cognitive impacts of Alzheimer's but also enhances the potential for developing linguistic-based diagnostic tools. By identifying and quantifying these distinctive linguistic markers, feature importance analysis provides a pathway for more targeted screenings and supports the broader discourse on the capabilities of natural language processing in healthcare screening. The findings from this study thus hold significant implications for both theoretical linguistics and practical applications in medical settings, offering valuable insights into the intersection of language function and neurological health.

Future studies could build on this work by incorporating larger datasets, ideally enhancing the scope and applicability of the results while continuing to address potential biases in data collection and model training; incorporate multilingual datasets and exploring alternative tokenization, embedding, and encoding strategies to enrich transcript representations; Increase computation power, and integrate feature-fusion techniques to further refine the identification processes, potentially leading to more robust models capable of diagnosing dementia across various stages and from more diverse demographic backgrounds.

Overall, this thesis contributes to the body of knowledge in applying NLP techniques to health outcomes research, particularly in the context of AD. By providing an extensive evaluation of different models and their interpretability through feature importance analysis, this study not only advances our understanding of the linguistic characteristics associated with AD but also sets the stage for future innovations in the field. The insights gained here underline the potential of NLP in medical noninvasive medical screening and the importance of continued research in this vital area.

References

1. Matošević, Lovro, and Alan Jović. "Accurate Detection of Dementia from Speech Transcripts Using RoBERTa Model." In 2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO), pp. 1478-1484. IEEE, 2022.
2. Nambiar, Anjana S., Kanigolla Likhita, KVS Sri Pujya, Deepa Gupta, Susmitha Vekkot, and S. Lalitha. "Comparative study of Deep Classifiers for Early Dementia Detection using Speech Transcripts." In 2022 IEEE 19th India Council International Conference (INDICON), pp. 1-6. IEEE, 2022.

3. Liu, Ziming, Lauren Proctor, Parker Collier, Devin Casenhiser, Eun Jin Paek, Si On Yoon, and Xiaopeng Zhao. "Machine learning of transcripts and audio recordings of spontaneous speech for diagnosis of Alzheimer's disease." Alzheimer's & Dementia 17 (2021): e057556.

4. Khan, Yusera Farooq, Baijnath Kaushik, and Bilal Ahmed Mir. "Computational Intelligent Models for Alzheimer's Prediction Using Audio Transcript Data." Computing and Informatics 41, no. 6 (2022): 1589-1624.

5. Bouazizi, Mondher, Chuheng Zheng, Siyuan Yang, and Tomoaki Ohtsuki. "Dementia Detection from Speech: What If Language Models Are Not the Answer?." Information 15, no. 1 (2023): 2.

6. Meghanani A, Anoop CS and Ramakrishnan AG (2021) Recognition of Alzheimer's Dementia From the Transcriptions of Spontaneous Speech Using fastText and CNN Models. Front. Comput. Sci. 3:624558. doi: 10.3389/fcomp.2021.624558

7. Kothari, Muskan, Darshil Vipul Shah, T. Moulya, Swasthi P. Rao, and R. Jayashree. "Measures of Lexical Diversity and Detection of Alzheimer's Using Speech." In ICAART (3), pp. 806-812. 2023.

8. Saltz, Ploypaphat, Shih Yin Lin, Sunny Chieh Cheng, and Dong Si. "Dementia detection using transformer-based deep learning and natural language processing models." In 2021 IEEE 9th International Conference on Healthcare Informatics (ICHI), pp. 509-510. IEEE, 2021.

9. Alzheimer's disease, Scheltens, Philip et al. The Lancet, Volume 397, Issue 10284, 1577 - 1590

10. Zhou, Xianbo, and J. Wesson Ashford. "Advances in screening instruments for Alzheimer's disease." Aging Medicine 2, no. 2 (2019): 88-93.

11. "Dementia." World Health Organization, March 15, 2023. https://www.who.int/news-room/fact-sheets/detail/dementia.

12. Di Palo, Flavio, and Natalie Parde. "Enriching neural models with targeted features for dementia detection." arXiv preprint arXiv:1906.05483 (2019).

13. Li, Changye, Weizhe Xu, Trevor Cohen, Martin Michalowski, and Serguei Pakhomov. "Trestle: Toolkit for reproducible execution of speech, text and language experiments." AMIA Summits on Translational Science Proceedings 2023 (2023): 360.

14. Pennington, Jeffrey, 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)*, pp. 1532-1543. 2014.

15. Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781* (2013).

16. Le, Quoc, and Tomas Mikolov. "Distributed representations of sentences and documents." In International conference on machine learning, pp. 1188-1196. PMLR, 2014.

17. Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

18. MacWhinney, Brian. "Computational models of child language learning: an introduction." Journal of Child language 37, no. 3 (2010): 477-485.

19. Lundberg, Scott M., and Su-In Lee. "A unified approach to interpreting model predictions." *Advances in neural information processing systems* 30 (2017).

20. Jang, Hyeju, Thomas Soroski, Matteo Rizzo, Oswald Barral, Anuj Harisinghani, Sally Newton-Mason, Saffrin Granby et al. "Classification of Alzheimer's disease leveraging multi-task machine learning analysis of speech and eye-movement data." Frontiers in Human Neuroscience 15 (2021): 716670.

21. Barral, Oswald, Hyeju Jang, Sally Newton-Mason, Sheetal Shajan, Thomas Soroski, Giuseppe Carenini, Cristina Conati, and Thalia Field. "Non-invasive classification of Alzheimer's disease using eye tracking and language." In Machine Learning for Healthcare Conference, pp. 813-841. PMLR, 2020.

22. Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. "" Why should i trust you?" Explaining the predictions of any classifier." In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1135-1144. 2016.

23. Vimbi, Viswan, Noushath Shaffi, and Mufti Mahmud. "Interpreting artificial intelligence models: a systematic review on the application of LIME and SHAP in Alzheimer's disease detection." *Brain Informatics* 11, no. 1 (2024): 10.

24. Abdelhalim, Nadine, Ingy Abdelhalim, and Riza Theresa Batista-Navarro. "Training Models on Oversampled Data and a Novel Multi-class Annotation Scheme for Dementia Detection." In *Proceedings of the 5th Clinical Natural Language Processing Workshop*, pp. 118-124. 2023.

25. Ilias, Loukas, and Dimitris Askounis. "Explainable identification of dementia from transcripts using transformer networks." *IEEE Journal of Biomedical and Health Informatics* 26, no. 8 (2022): 4153-4164.

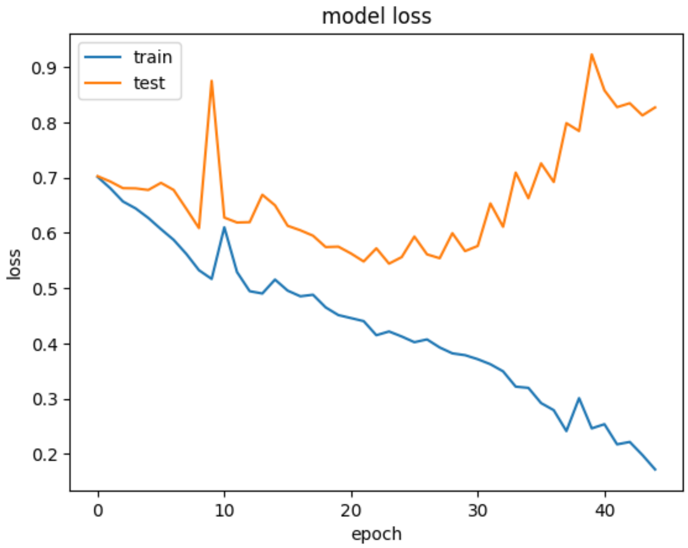
26. DementiaBank (Pitt corpus): Becker, J. T., Boller, F., Lopez, O. L., Saxton, J., & McGonigle, K. L. (1994). The natural history of Alzheimer's disease: description of study cohort and accuracy of diagnosis. *Archives of Neurology*, 51(6), 585-594.

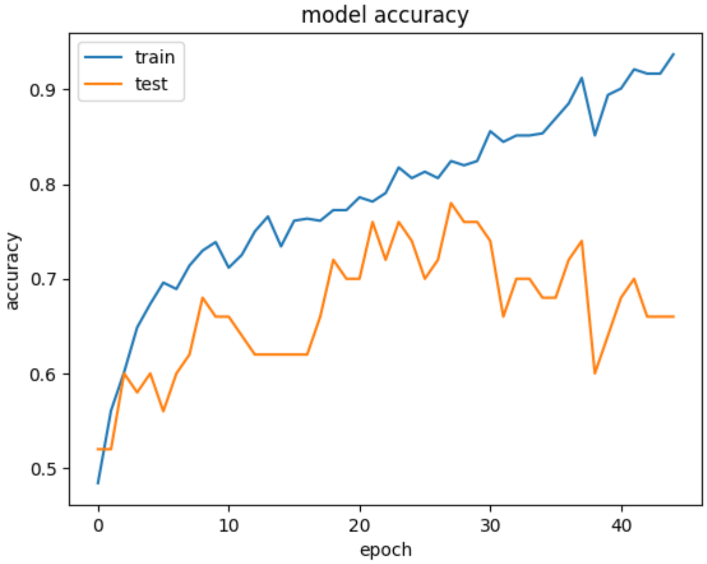
27. Sundararajan, Mukund, Ankur Taly, and Qiqi Yan. "Axiomatic attribution for deep networks." In *International conference on machine learning*, pp. 3319-3328. PMLR, 2017.

Appendix

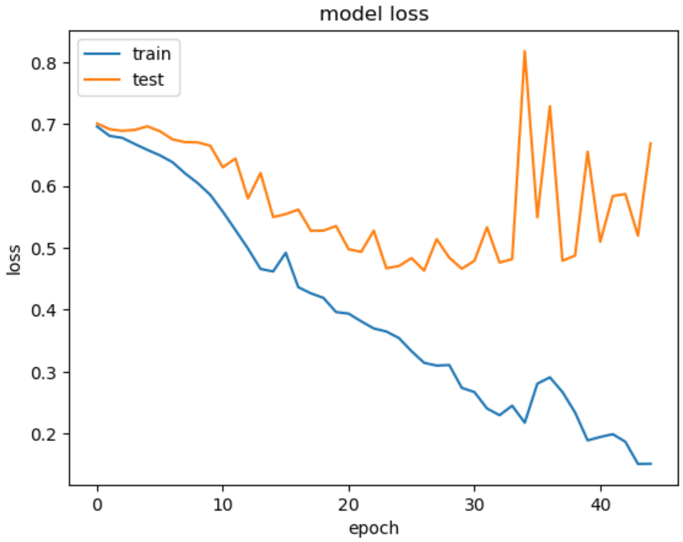
Figure 8

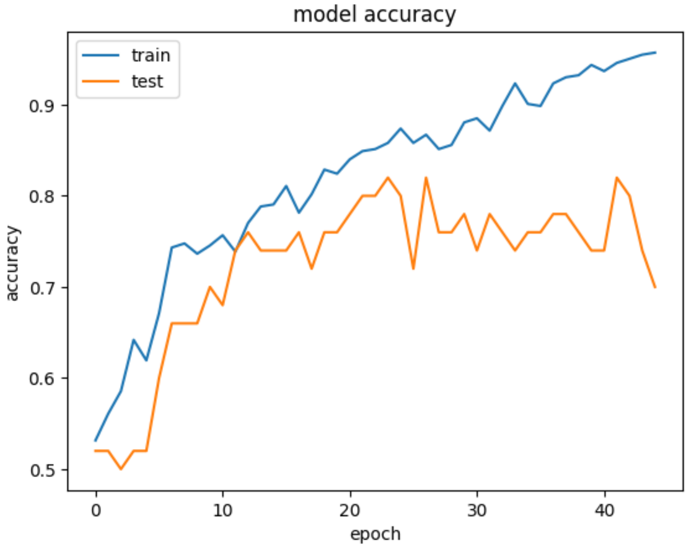
Doc2Vec LSTM



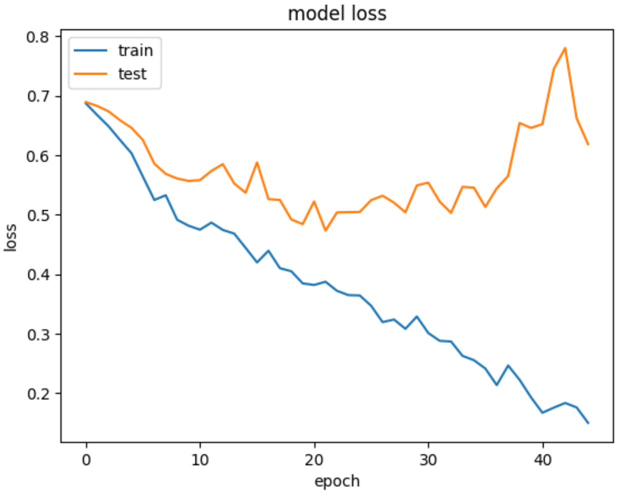


Doc2Vec CNN-LSTM





Doc2Vec Bi-LSTM


model loss
train
test
loss
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
10
20
30
40
epoch

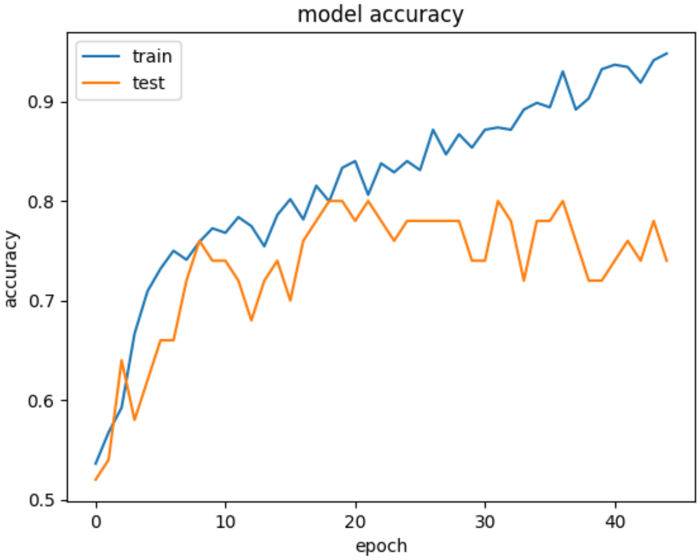
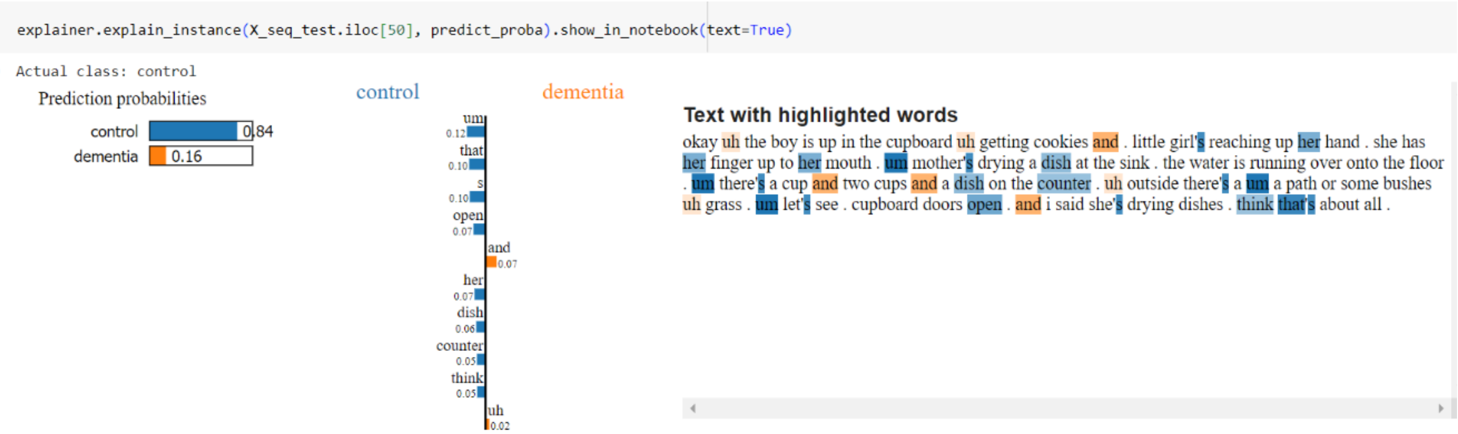

model accuracy
train
test
accuracy
0.9
0.8
0.7
0.6
0.5
0
10
20
30
40
epoch

Figure Nine

LIME

LIME for Doc2Vec LSTM

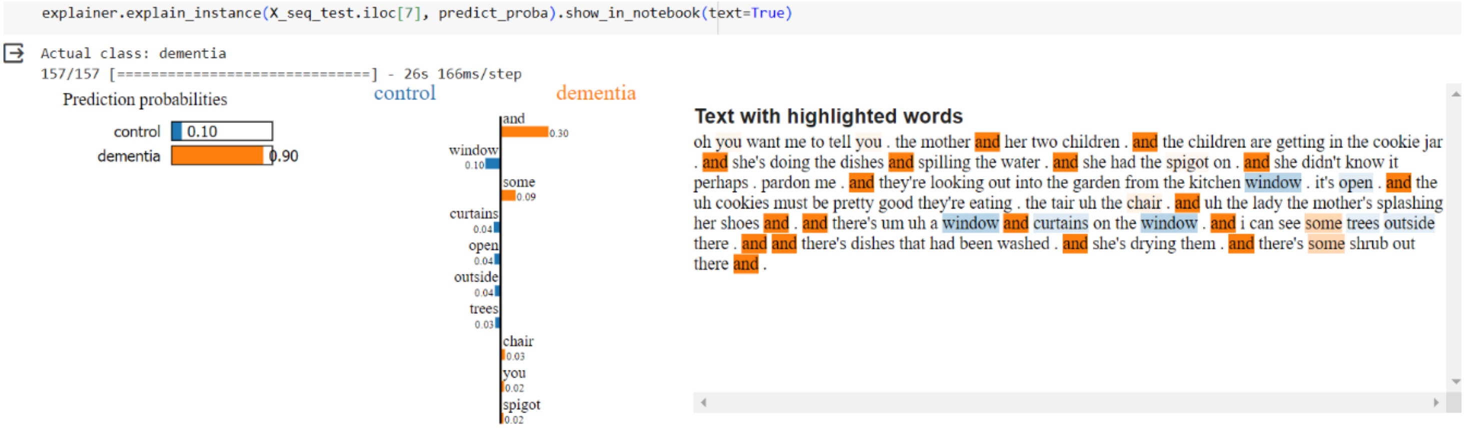


```
explainer.explain_instance(X_seq_test.iloc[50], predict_proba).show_in_notebook(text=True)
```

```
Actual class: control
```

Text with highlighted words

okay uh the boy is up in the cupboard uh getting cookies and . little girl's reaching up her hand . she has her finger up to her mouth . um mother's drying a dish at the sink . the water is running over onto the floor . um there's a cup and two cups and a dish on the counter . uh outside there's a um a path or some bushes uh grass . um let's see . cupboard doors open . and i said she's drying dishes . think that's about all .

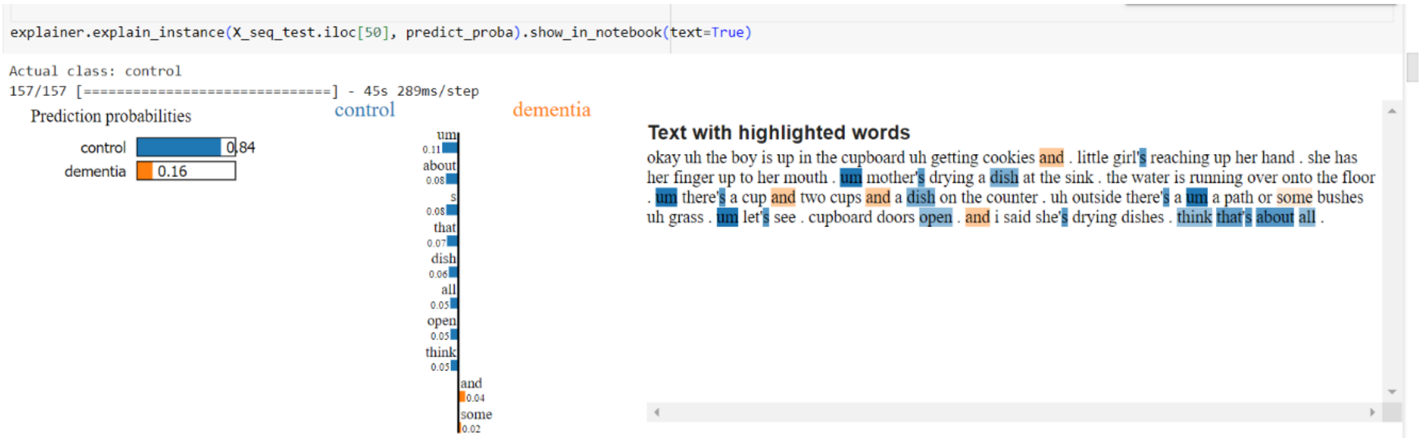


```
explainer.explain_instance(X_seq_test.iloc[7], predict_proba).show_in_notebook(text=True)
```

```
Actual class: dementia
157/157 [==============================] - 26s 166ms/step
```

Text with highlighted words

oh you want me to tell you . the mother and her two children . and the children are getting in the cookie jar . and she's doing the dishes and spilling the water . and she had the spigot on . and she didn't know it perhaps . pardon me . and they're looking out into the garden from the kitchen window . it's open . and the uh cookies must be pretty good they're eating . the tair uh the chair . and uh the lady the mother's splashing her shoes and . and there's um uh a window and curtains on the window . and i can see some trees outside there . and and there's dishes that had been washed . and she's drying them . and there's some shrub out there and .

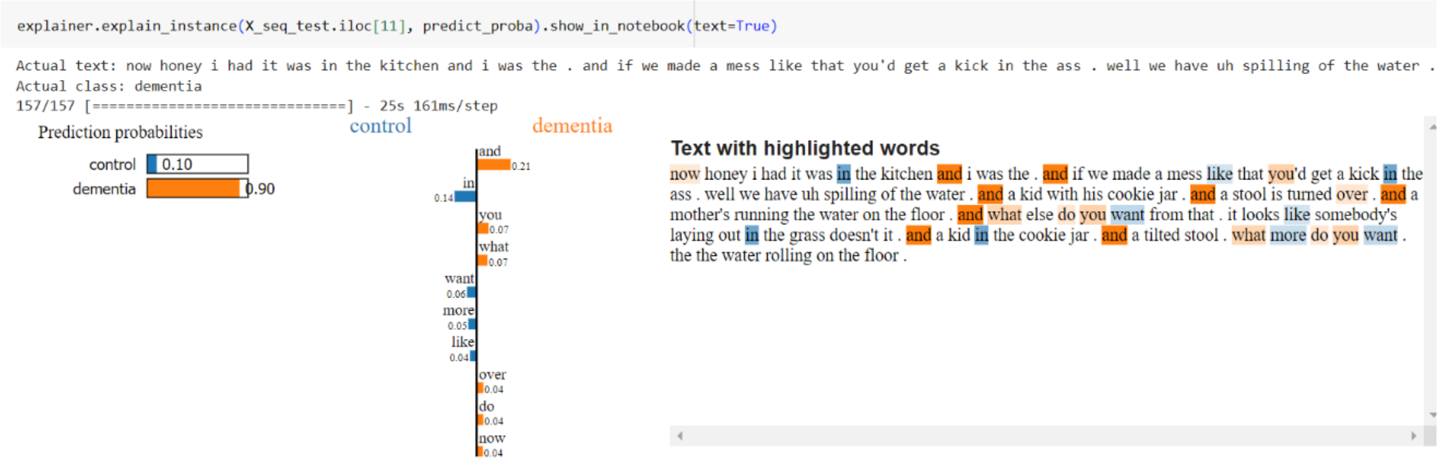


```
explainer.explain_instance(X_seq_test.iloc[50], predict_proba).show_in_notebook(text=True)
```

```
Actual class: control
157/157 [==============================] - 45s 289ms/step
```

Text with highlighted words

okay uh the boy is up in the cupboard uh getting cookies and . little girl's reaching up her hand . she has her finger up to her mouth . um mother's drying a dish at the sink . the water is running over onto the floor . um there's a cup and two cups and a dish on the counter . uh outside there's a um a path or some bushes uh grass . um let's see . cupboard doors open . and i said she's drying dishes . think that's about all .



```
explainer.explain_instance(X_seq_test.iloc[11], predict_proba).show_in_notebook(text=True)
```

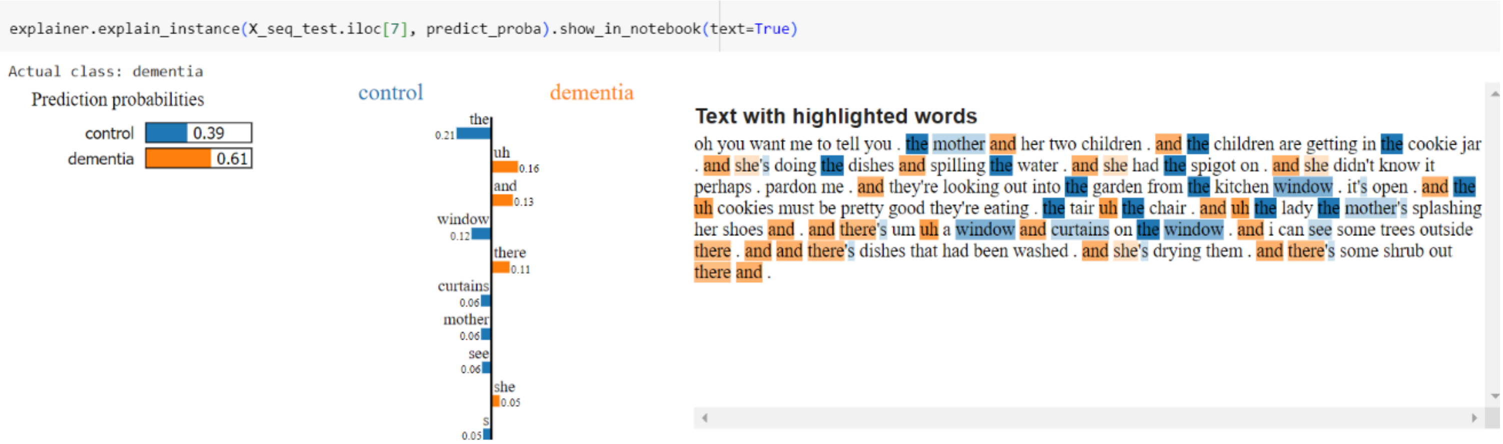
```
Actual text: now honey i had it was in the kitchen and i was the . and if we made a mess like that you'd get a kick in the ass . well we have uh spilling of the water .
Actual class: dementia
157/157 [==============================] - 25s 161ms/step
```

Text with highlighted words

now honey i had it was in the kitchen and i was the . and if we made a mess like that you'd get a kick in the ass . well we have uh spilling of the water . and a kid with his cookie jar . and a stool is turned over . and a mother's running the water on the floor . and what else do you want from that . it looks like somebody's laying out in the grass doesn't it . and a kid in the cookie jar . and a tilted stool . what more do you want . the the water rolling on the floor .

Transcripts (7) and (11) of the Doc2Vec LSTM model demonstrated a 90% prediction probability for dementia, correctly classified. These transcripts were marked by the frequent use of the word 'and', indicating fragmented and incomplete sentence constructions typical in AD speech patterns. Instance (11) displayed repetitive phrases, such as 'what else do you want from that' and 'what more do you want', which align with known repetitive speech behaviors in dementia. Transcript (50) predicted with an 84% probability as the control class, featuring coherent tokens like 'that', 'dish', 'counter', 'open', and 'apostrophe s', suggesting structured and coherent speech.

LIME for Doc2Vec CNN-LSTM

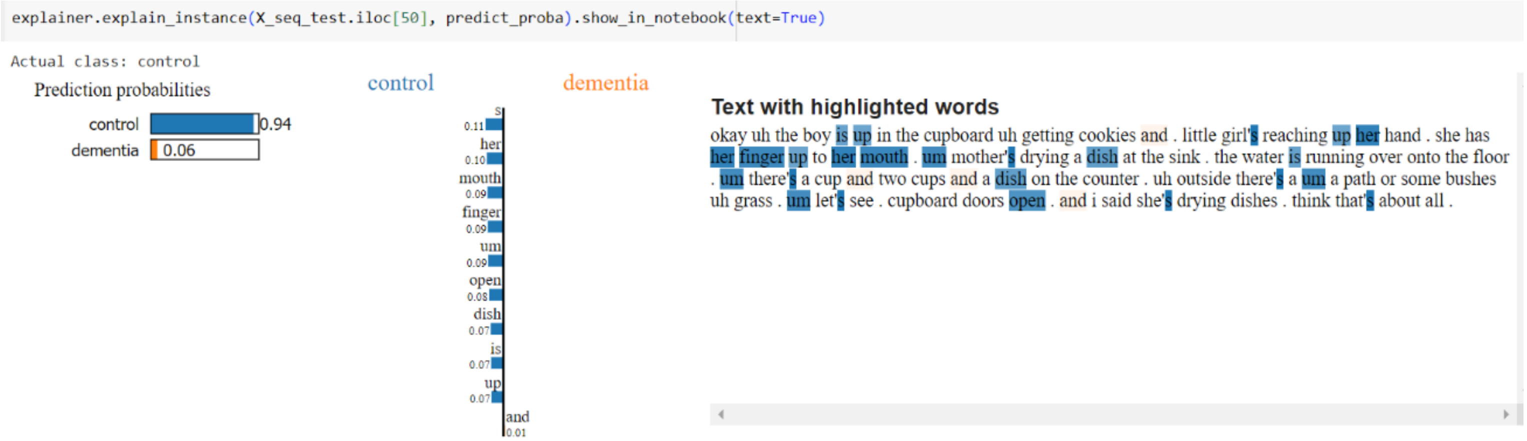


```
explainer.explain_instance(X_seq_test.iloc[7], predict_proba).show_in_notebook(text=True)
```

```
Actual class: dementia
```

Text with highlighted words

oh you want me to tell you . the mother and her two children . and the children are getting in the cookie jar . and she's doing the dishes and spilling the water . and she had the spigot on . and she didn't know it perhaps . pardon me . and they're looking out into the garden from the kitchen window . it's open . and the uh cookies must be pretty good they're eating . the tair uh the chair . and uh the lady the mother's splashing her shoes and . and there's um uh a window and curtains on the window . and i can see some trees outside there . and and there's dishes that had been washed . and she's drying them . and there's some shrub out there and .



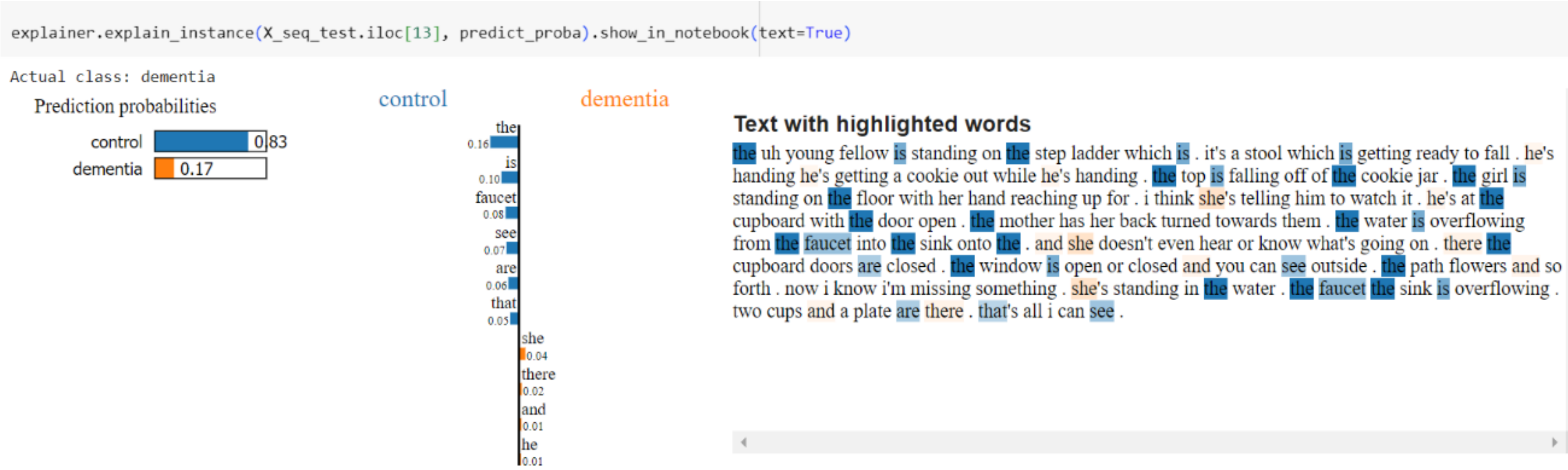
```
explainer.explain_instance(X_seq_test.iloc[50], predict_proba).show_in_notebook(text=True)
```

```
Actual class: control
```

Text with highlighted words

okay uh the boy is up in the cupboard uh getting cookies and . little girl's reaching up her hand . she has her finger up to her mouth . um mother's drying a dish at the sink . the water is running over onto the floor . um there's a cup and two cups and a dish on the counter . uh outside there's a um a path or some bushes uh grass . um let's see . cupboard doors open . and i said she's drying dishes . think that's about all .

BERT CNN-LSTM

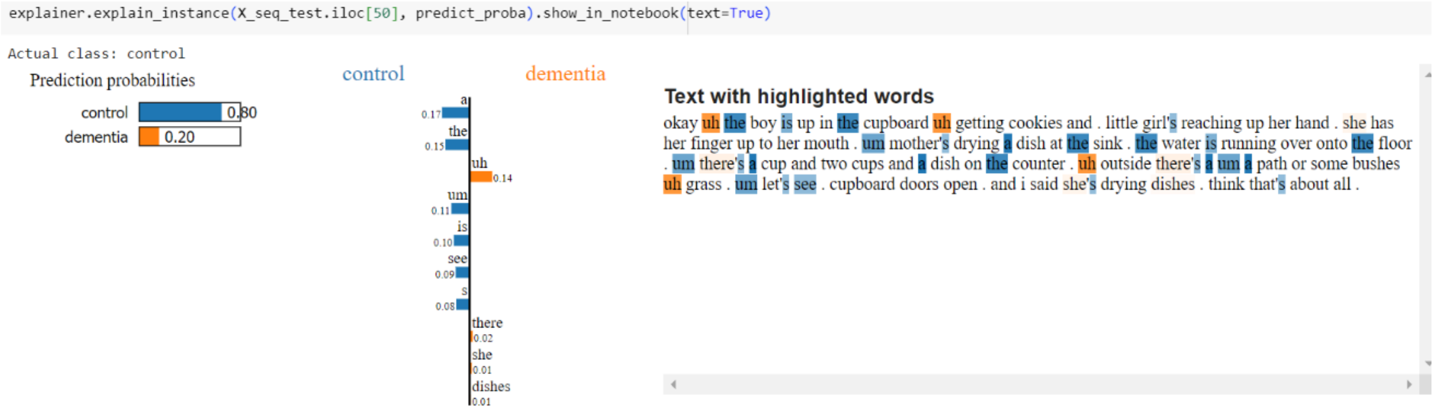


```
explainer.explain_instance(X_seq_test.iloc[13], predict_proba).show_in_notebook(text=True)
```

```
Actual class: dementia
```

Text with highlighted words

the uh young fellow is standing on the step ladder which is . it's a stool which is getting ready to fall . he's handing he's getting a cookie out while he's handing . the top is falling off of the cookie jar . the girl is standing on the floor with her hand reaching up for . i think she's telling him to watch it . he's at the cupboard with the door open . the mother has her back turned towards them . the water is overflowing from the faucet into the sink onto the . and she doesn't even hear or know what's going on . there the cupboard doors are closed . the window is open or closed and you can see outside . the path flowers and so forth . now i know i'm missing something . she's standing in the water . the faucet the sink is overflowing . two cups and a plate are there . that's all i can see .

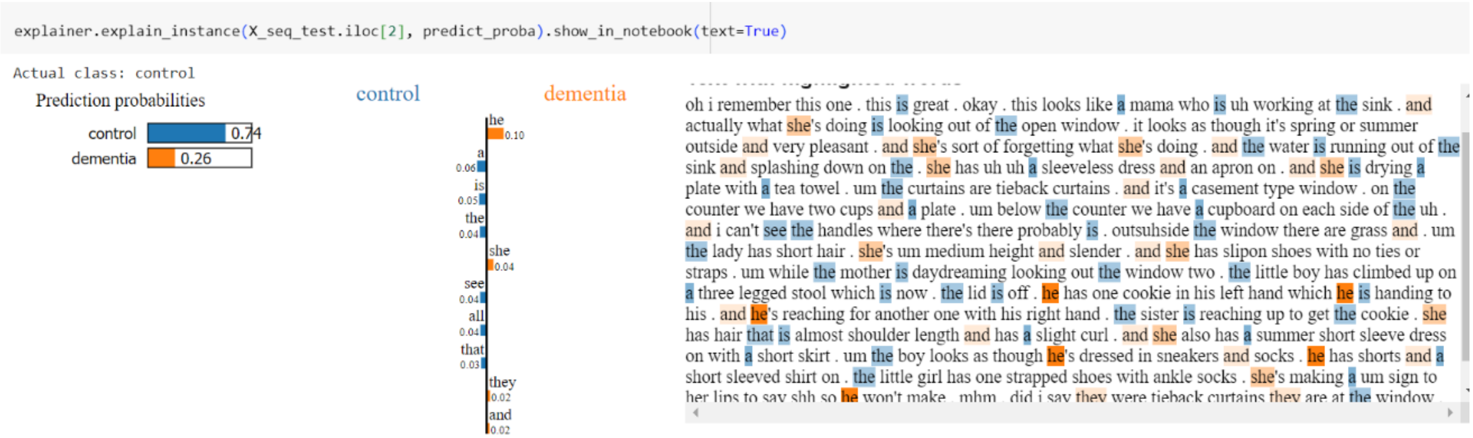


```
explainer.explain_instance(X_seq_test.iloc[50], predict_proba).show_in_notebook(text=True)
```

```
Actual class: control
```

Text with highlighted words

okay uh the boy is up in the cupboard uh getting cookies and . little girl's reaching up her hand . she has her finger up to her mouth . um mother's drying a dish at the sink . the water is running over onto the floor . um there's a cup and two cups and a dish on the counter . uh outside there's a um a path or some bushes uh grass . um let's see . cupboard doors open . and i said she's drying dishes . think that's about all .



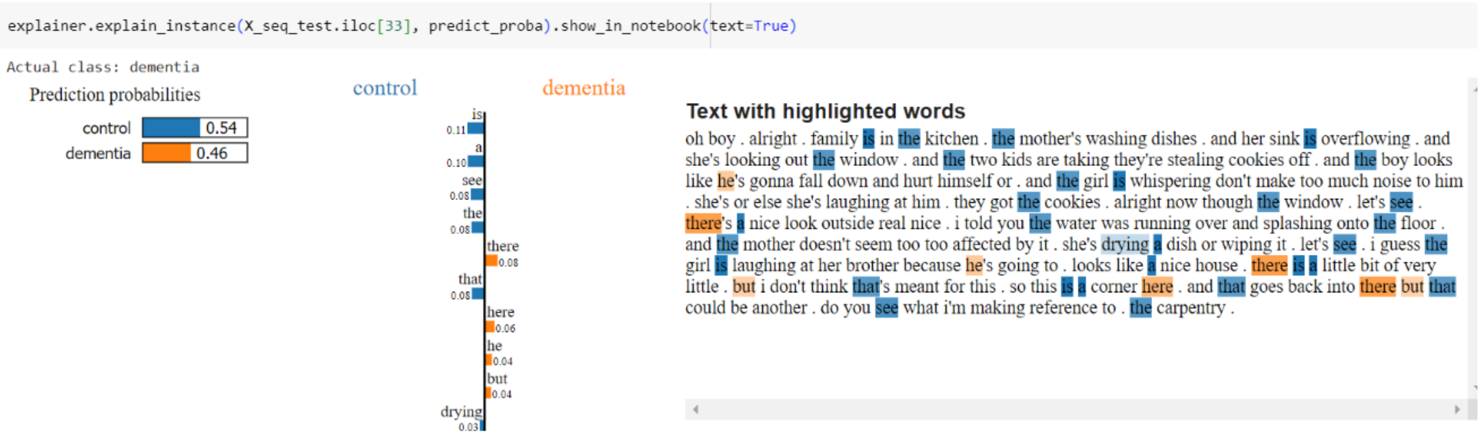
```
explainer.explain_instance(X_seq_test.iloc[2], predict_proba).show_in_notebook(text=True)
```

```
Actual class: control
```

oh i remember this one . this is great . okay . this looks like a mama who is uh working at the sink . and actually what she's doing is looking out of the open window . it looks as though it's spring or summer outside and very pleasant . and she's sort of forgetting what she's doing . and the water is running out of the sink and splashing down on the . she has uh uh a sleeveless dress and an apron on . and she is drying a plate with a tea towel . um the curtains are tieback curtains . and it's a casement type window . on the counter we have two cups and a plate . um below the counter we have a cupboard on each side of the uh . and i can't see the handles where there's there probably is . outsuhside the window there are grass and . um the lady has short hair . she's um medium height and slender . and she has slipon shoes with no ties or straps . um while the mother is daydreaming looking out the window two . the little boy has climbed up on a three legged stool which is now . the lid is off . he has one cookie in his left hand which he is handing to his . and he's reaching for another one with his right hand . the sister is reaching up to get the cookie . she has hair that is almost shoulder length and has a slight curl . and she also has a summer short sleeve dress on with a short skirt . um the boy looks as though he's dressed in sneakers and socks . he has shorts and a short sleeved shirt on . the little girl has one strapped shoes with ankle socks . she's making a um sign to her lips to say shh so he won't make . mhm . did i say they were tieback curtains they are at the window .

Transcript (50) had an 80% probability of being part of the control class, which is notably lower than the prediction probability of it's doc2vec model counter parts, with predominant tokens 'a' and 'the' influencing the explainer. Transcript (2), which was also lower than the Doc2Vec models, had a 74% probability of control, influenced strongly by 'a', 'is', 'the', demonstrating clear and logical speech patterns. Predictions explained for the BERT CNN-LSTM model also sheds light on instances of misclassification, where LIME predicted a control classification based on the presence of words typically found in control transcripts, but the true class was AD. Conversely, terms that often appeared in dementia-classified transcripts, such as 'uh', 'and', 'there', and 'he', were pivotal in instances where dementia was correctly identified or misclassified, indicating their strong associative impact on model predictions. The term 'he' showed up as a strong indicator for the dementia class potentially for the focus on the sentences containing that term described the boy sneaking into the cookie jar. Instance (7) predicted the correct class dementia with weaker probability of 61%, 'the' was the most heavily weighted word for the control class and for dementia 'uh', 'and' and 'there' had the strongest affect.

Transcript (13) an example of a prediction probability being misclassified. The actual class was dementia and LIME predicted the probability of control at 83% because this transcript contained lots of words that have commonly been found in control transcripts, such as ‘the’, ‘is’, and ‘faucet’. Another example of an incorrect prediction probability is instance (33). The true class was dementia, but LIME gave a 54% probability of the transcript being control and 46% of probability of the transcript being dementia. The transcript itself contained ‘is’, ‘a’, ‘see’, for control and ‘there’, here’, and ‘he’ for dementia.



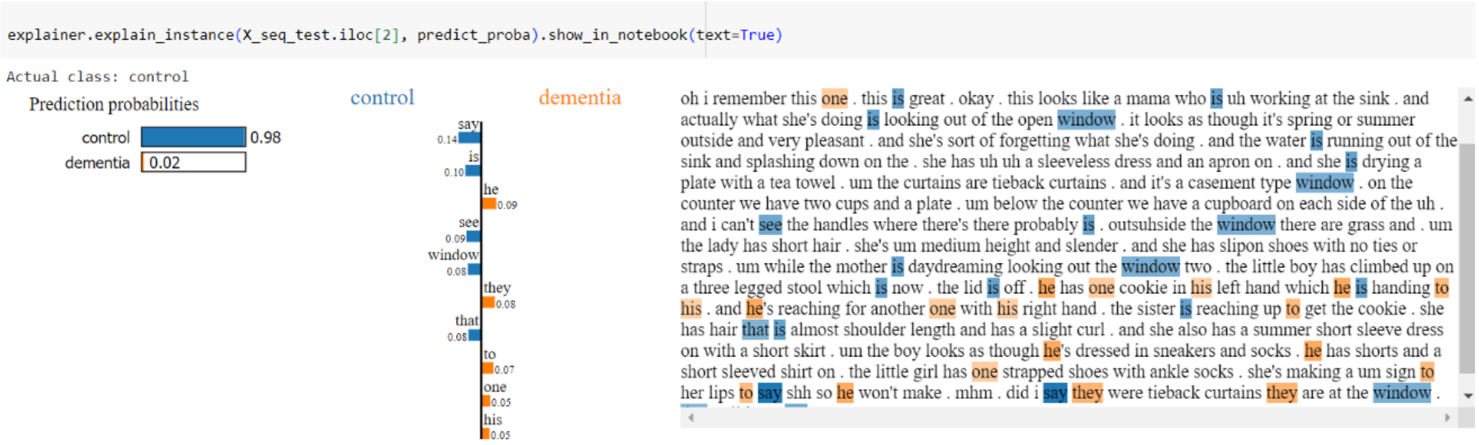
```
explainer.explain_instance(X_seq_test.iloc[33], predict_proba).show_in_notebook(text=True)
```

```
Actual class: dementia
```

Text with highlighted words

oh boy . alright . family is in the kitchen . the mother's washing dishes . and her sink is overflowing . and she's looking out the window . and the two kids are taking they're stealing cookies off . and the boy looks like he's gonna fall down and hurt himself or . and the girl is whispering don't make too much noise to him . she's or else she's laughing at him . they got the cookies . alright now though the window . let's see . there's a nice look outside real nice . i told you the water was running over and splashing onto the floor . and the mother doesn't seem too too affected by it . she's drying a dish or wiping it . let's see . i guess the girl is laughing at her brother because he's going to . looks like a nice house . there is a little bit of very little . but i don't think that's meant for this . so this is a corner here . and that goes back into there but that could be another . do you see what i'm making reference to . the carpentry .

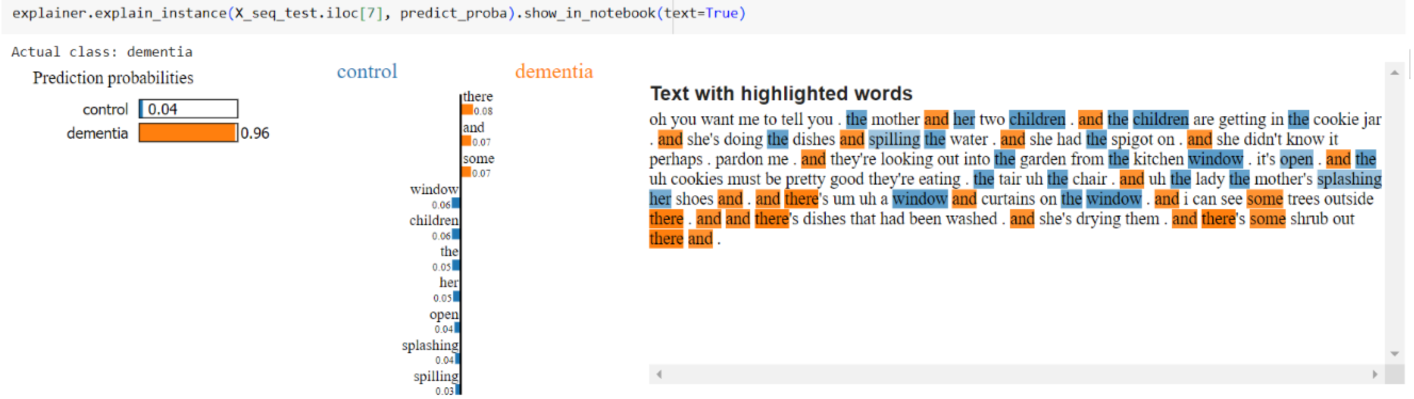
Doc2Vec Bi-LSTM



```
explainer.explain_instance(X_seq_test.iloc[2], predict_proba).show_in_notebook(text=True)
```

```
Actual class: control
```

oh i remember this one . this is great . okay . this looks like a mama who is uh working at the sink . and actually what she's doing is looking out of the open window . it looks as though it's spring or summer outside and very pleasant . and she's sort of forgetting what she's doing . and the water is running out of the sink and splashing down on the . she has uh uh a sleeveless dress and an apron on . and she is drying a plate with a tea towel . um the curtains are tieback curtains . and it's a casement type window . on the counter we have two cups and a plate . um below the counter we have a cupboard on each side of the uh . and i can't see the handles where there's there probably is . outsuhside the window there are grass and . um the lady has short hair . she's um medium height and slender . and she has slipon shoes with no ties or straps . um while the mother is daydreaming looking out the window two . the little boy has climbed up on a three legged stool which is now . the lid is off . he has one cookie in his left hand which he is handing to his . and he's reaching for another one with his right hand . the sister is reaching up to get the cookie . she has hair that is almost shoulder length and has a slight curl . and she also has a summer short sleeve dress on with a short skirt . um the boy looks as though he's dressed in sneakers and socks . he has shorts and a short sleeved shirt on . the little girl has one strapped shoes with ankle socks . she's making a um sign to her lips to say shh so he won't make . mhm . did i say they were tieback curtains they are at the window .

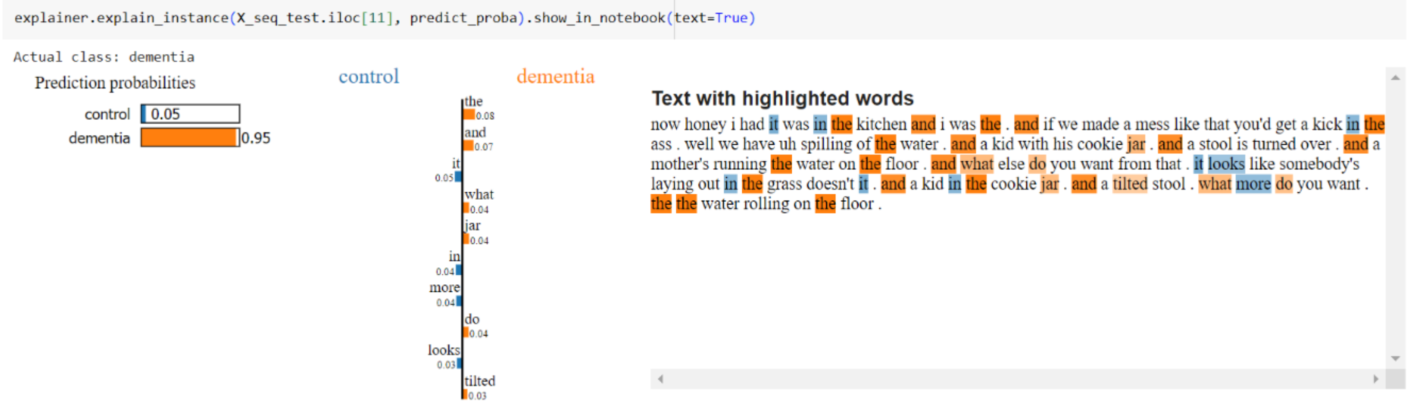


```
explainer.explain_instance(X_seq_test.iloc[7], predict_proba).show_in_notebook(text=True)
```

```
Actual class: dementia
```

Text with highlighted words

oh you want me to tell you . the mother and her two children . and the children are getting in the cookie jar . and she's doing the dishes and spilling the water . and she had the spigot on . and she didn't know it perhaps . pardon me . and they're looking out into the garden from the kitchen window . it's open . and the uh cookies must be pretty good they're eating . the tair uh the chair . and uh the lady the mother's splashing her shoes and . and there's um uh a window and curtains on the window . and i can see some trees outside there . and and there's dishes that had been washed . and she's drying them . and there's some shrub out there and .



```
explainer.explain_instance(X_seq_test.iloc[11], predict_proba).show_in_notebook(text=True)
```

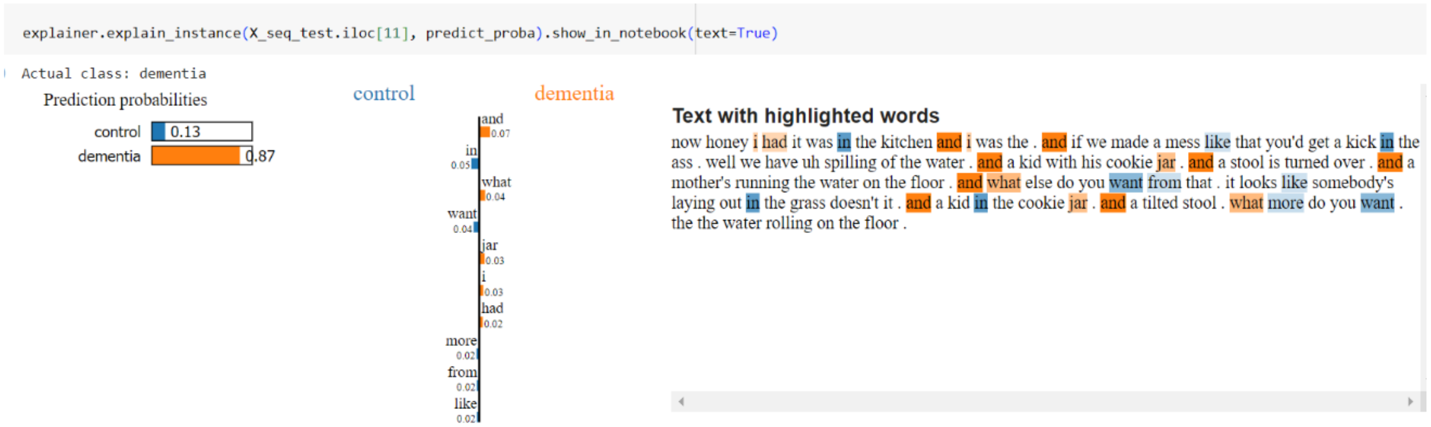
```
Actual class: dementia
```

Text with highlighted words

now honey i had it was in the kitchen and i was the . and if we made a mess like that you'd get a kick in the ass . well we have uh spilling of the water . and a kid with his cookie jar . and a stool is turned over . and a mother's running the water on the floor . and what else do you want from that . it looks like somebody's laying out in the grass doesn't it . and a kid in the cookie jar . and a tilted stool . what more do you want . the the water rolling on the floor .

Transcript (50), similar to the LSTM model, LIME predicted as part of the control class with an 84% probability. It contained overlapping significant tokens with the LSTM model including 'open', 'dish', 'that', and 'her'. In transcript (7), the AD class was predicted with a 96% probability, with critical tokens, such as 'there', 'and', and 'some', which are indicative of AD due to their repetitive use in the transcript. We see repetitive words like ‘and' and 'there’ more than once and in multiple dementia transcripts. Instance (11), a dementia transcript with the prediction probability of AD as 95%, repeats 'a kid in in the cookie jar', as well as, the word ‘the’ and ‘and’. Transcript (2) predicted the probability of control, the correct class, at a 98% probability with words like 'say', 'is', and 'window' being the most important.

Doc2Vec CNN-LSTM

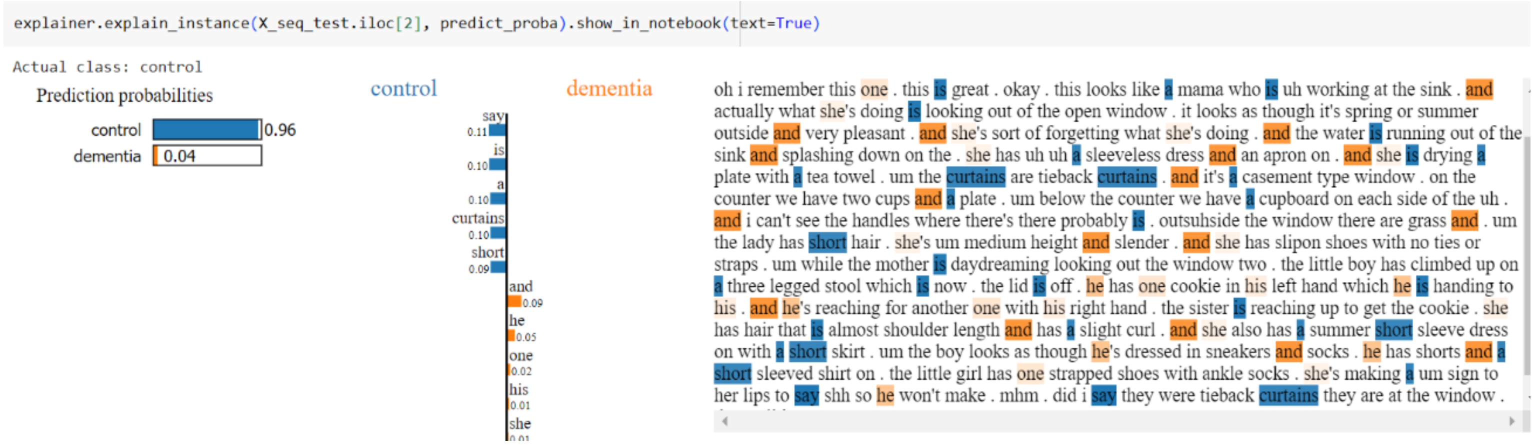


```
explainer.explain_instance(X_seq_test.iloc[11], predict_proba).show_in_notebook(text=True)
```

```
Actual class: dementia
```

Text with highlighted words

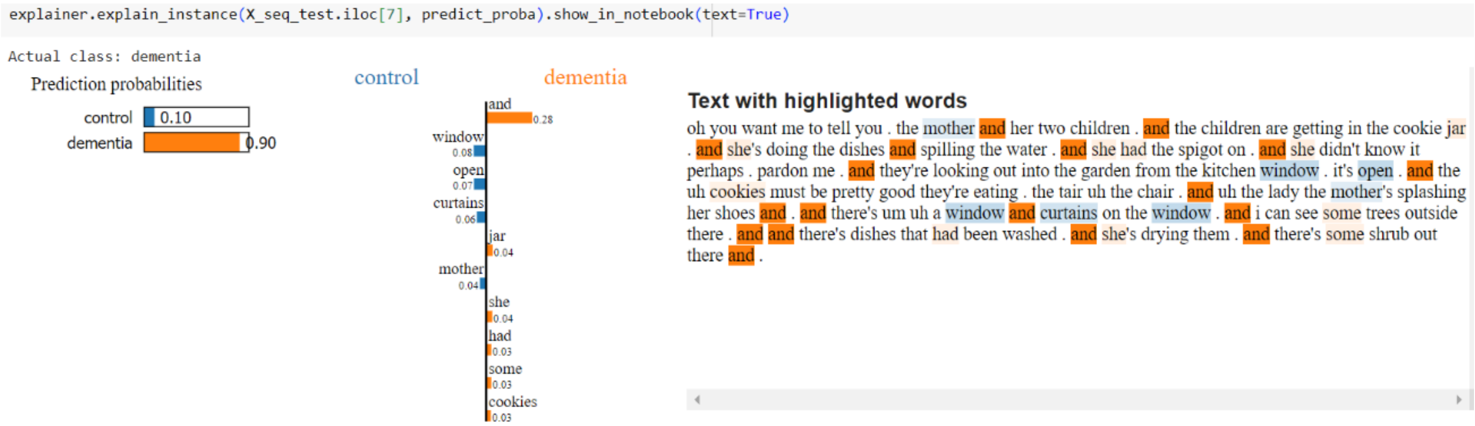
now honey i had it was in the kitchen and i was the . and if we made a mess like that you'd get a kick in the ass . well we have uh spilling of the water . and a kid with his cookie jar . and a stool is turned over . and a mother's running the water on the floor . and what else do you want from that . it looks like somebody's laying out in the grass doesn't it . and a kid in the cookie jar . and a tilted stool . what more do you want . the the water rolling on the floor .



```
explainer.explain_instance(X_seq_test.iloc[2], predict_proba).show_in_notebook(text=True)
```

```
Actual class: control
```

oh i remember this one . this is great . okay . this looks like a mama who is uh working at the sink . and actually what she's doing is looking out of the open window . it looks as though it's spring or summer outside and very pleasant . and she's sort of forgetting what she's doing . and the water is running out of the sink and splashing down on the . she has uh uh a sleeveless dress and an apron on . and she is drying a plate with a tea towel . um the curtains are tieback curtains . and it's a casement type window . on the counter we have two cups and a plate . um below the counter we have a cupboard on each side of the uh . and i can't see the handles where there's there probably is . outsuhside the window there are grass and . um the lady has short hair . she's um medium height and slender . and she has slipon shoes with no ties or straps . um while the mother is daydreaming looking out the window two . the little boy has climbed up on a three legged stool which is now . the lid is off . he has one cookie in his left hand which he is handing to his . and he's reaching for another one with his right hand . the sister is reaching up to get the cookie . she has hair that is almost shoulder length and has a slight curl . and she also has a summer short sleeve dress on with a short skirt . um the boy looks as though he's dressed in sneakers and socks . he has shorts and a short sleeved shirt on . the little girl has one strapped shoes with ankle socks . she's making a um sign to her lips to say shh so he won't make . mhm . did i say they were tieback curtains they are at the window .



```
explainer.explain_instance(X_seq_test.iloc[7], predict_proba).show_in_notebook(text=True)
```

```
Actual class: dementia
```

Text with highlighted words

oh you want me to tell you . the mother and her two children . and the children are getting in the cookie jar . and she's doing the dishes and spilling the water . and she had the spigot on . and she didn't know it perhaps . pardon me . and they're looking out into the garden from the kitchen window . it's open . and the uh cookies must be pretty good they're eating . the tair uh the chair . and uh the lady the mother's splashing her shoes and . and there's um uh a window and curtains on the window . and i can see some trees outside there . and and there's dishes that had been washed . and she's drying them . and there's some shrub out there and .

Transcript (2) predicted a 96% probability of control (with a true class of control). Key tokens included 'say', 'is', 'a', and 'curtains', denoting clear and structured speech. Transcript (11) displayed an 87% probability of dementia, with 'and', 'what', and 'jar' significantly influencing the prediction, reflecting typical narrative disruptions seen in dementia. Whereas, transcript (7) highlighted 'and' and 'jar' as strong indicators of AD, with a 90% probability of correct dementia classification.