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

Alzheimer's Disease Detection from Spontaneous Speech through Combining Linguistic Complexity and (Dis)Fluency Features with Pretrained Language Models

Yu Qiao[1], *Xuefeng Yin*[1], *Daniel Wiechmann*[2], *Elma Kerz*[1]

[1]RWTH Aachen University, Germany
[2]University of Amsterdam, Netherlands

yu.qiao@rwth-aachen.de, xuefeng.yin@rwth-aachen.de, d.wiechmann@uva.nl, elma.kerz@ifaar.rwth-aachen.de

Abstract

In this paper, we combined linguistic complexity and (dis)fluency features with pretrained language models for the task of Alzheimer's disease detection of the 2021 ADReSSo (Alzheimer's Dementia Recognition through Spontaneous Speech) challenge. An accuracy of 83.1% was achieved on the test set, which amounts to an improvement of 4.23% over the baseline model. Our best-performing model that integrated component models using a stacking ensemble technique performed equally well on cross-validation and test data, indicating that it is robust against overfitting.

Index Terms: Alzheimer's disease, disfluency, pretrained language models, automated Alzheimer's disease detection, linguistic complexity

1. Introduction

Alzheimer's disease (AD) is a gradual and progressive neurodegenerative disease caused by neuronal cell death [1]. The number of people diagnosed with AD is rapidly increasing[1]. The high prevalence of the disease and the high costs associated with traditional approaches to detection make research on automatic detection of AD critical [2]. A growing body of research has demonstrated that quantifiable indicators of cognitive decline associated with AD are detectable in spontaneous speech (see [3] for a recent review). These indicators encompass acoustic features, such as vocalisation features (i.e. speech-silence patterns) [4], paralinguistic features, such as fluency features [5] and speech pause distributions [6], as well as syntactic and lexical features extracted from speech transcripts [7].

This area of research has benefited from recent advances in natural language processing and machine learning, as well as an increasing number of interdisciplinary research collaborations. A prime example of this is the ADReSS(o) (Alzheimer's Dementia Recognition through Spontaneous Speech) Challenge, aimed at generating systematic evidence for the use of such indicators in automated AD detection systems and towards their clinical implementation. This challenge has made significant contributions to research on AD detection by enabling the research community to test their existing methods, develop novel approaches and to benchmark their AD detection systems on a shared dataset. The ADReSSo Challenge at INTERSPEECH 2021 [8] is geared towards automatic recognition of AD from spontaneous speech and involved three subtasks. Here in this paper, we focus on the AD classification subtask, for which research teams were asked to build a model to predict the label (AD or non-AD) for a short speech session. Participating teams could use the speech signal directly and extract acoustic features or automatically convert the speech to text (ASR) and extract linguistic features from this ASR-generated transcript.

1.1. Related work

In this section, we provide a concise review of research on automatic AD detection through speech, with particular attention to previous studies conducted as part of the 2020 ADReSS Challenge. The AD classification approaches in this challenge relied on a wide range of acoustic, paralinguistic, and linguistic features or their combination. Classification accuracy scores of the proposed models ranged between 68% and 89.6%. While some approaches either focused on acoustic or linguistic features, the best performing contributions in the 2020 challenge embraced a multi-modal approach combining several types of features (e.g. [9][10][11]). Furthermore, building on earlier work reporting on the effectiveness of the use of word embeddings in AD detection ([12][13]), several approaches successfully employed pretrained language models (e.g. [9][10][11]). Another important issue addressed in several studies concerned how to deal with variance in the predictive performance of pretrained models resulting from fine-tuning for downstream tasks with a small data set. In response to this issue, the authors of the best performing model [9] introduced an ensemble method to increase the robustness of their approach. In response to this issue, the best performing paper of the 2020 challenge [9] introduced an ensemble approach to increase the robustness of their models. Finally, it is important to note that some of the high-performing models in last year's challenge – including the best model described in [9] – used rich manual transcription that included pause and disfluency annotation. Such transcripts were not provided in the 2021 challenge, making it more demanding compared to last year's challenge.

1.2. Modeling approach

The modeling approach presented in this paper builds on key insights reported in the studies reviewed above and extends on these (1) by integrating linguistic indicators of linguistic complexity and sophistication, features of (dis)fluency and transformer-based pretrained language models and (2) by utilizing ensembling methods to combine the information from these feature groups and to reduce the variance in model predictions. Specifically, we perform experiments with classification based on three ensembling techniques: Ensembling by bagging via majority vote, ensembling by bagging using feature fusion, and ensembling by stacking.

[1] https://www.alz.org/alzheimers-dementia/facts-figures

http://dx.doi.org/10.21437/Interspeech.2021-1415

2. Data and analysis

2.1. Data

The Alzheimer's Disease Detection dataset provided by the organizers of the ADReSSo Challenge 2021 consists of speech recordings of picture descriptions from the Boston Diagnostic Aphasia Exam produced by 87 individuals with an AD diagnosis and 79 cognitively normal subjects (control group). The recordings were acoustically enhanced (noise reduction through spectral subtraction) and normalised. The data were also balanced with respect to age and gender. The organizers also provided segmentations of the recordings into vocalisation sequences with speaker identifiers. No transcripts were provided.

2.2. Speech Recognition

We used AppTek's Automatic Speech Recognition technology via a cloud API service[2] for automatically transcribing the audio files. The transcripts were converted from XML into raw text formats with full stops being added at the end of each utterance based on the provided segmentations. These files served as the input for the automated text analysis (see Section 2.4).

2.3. (Dis)fluency

To model the speakers' articulatory (in particular (dis)fluency-related) characteristics, we derived several features from the ASR system that fall into four classes. (1) *Silent pauses* - The ASR output contained the start- and end-times as well as confidence scored for each recognized word. Durations of pauses were calculated from forced alignment and binned by duration into short pauses ($< 2sec$) and long pauses ($> 2sec$). In addition, we calculated the total pause duration per sentence (in seconds). (2) *Speed of articulation* - We enriched the output of the ASR with syllable counts from the Carnegie Mellon University Pronouncing Dictionary[3]. Based on this information we assessed the mean syllable duration as well as syllables per minute for each utterance in the speech data. (3) *Filled pauses* - Next to the number and total duration of silent pauses, we derived frequency counts per sentence for two filled pause type, *uh* and *um*, that had been shown to discriminate between AD patients and controls in previous studies [9]. (4) *Pronunciation* - As the known symptoms of AD patients include mispronunciation [14], we calculated average word level confidence scores as a proxy of pronunciation quality, which have been employed for the speech pattern detection in the context of detection of Alzheimer's Disease [15]. All measures were calculated at utterance level. An overview of these measures with descriptive statistics for both groups is presented in Table 1.

2.4. Automated Text Analysis (ATA)

The speech transcripts were automatically analyzed using CoCoGen (short for: Complexity Contour Generator), a computational tool that implements a sliding window technique to calculate within-text distributions of scores for a given language feature (for current applications of the tool in the context of text classification, see [16, 17, 18]). In this paper, we employed a total of 293 features derived from interdisciplinary, integrated approaches to language [19] that fall into four categories: (1) measures of syntactic complexity, (2) measures of lexical richness, (3) register-based n-gram frequency measures, and (4) information-theoretic measures. In contrast to the standard approach implemented in other software for automated text analysis that relies on aggregate scores representing the average value of a feature in a text, the sliding-window approach employed in CoCoGen tracks the distribution of the feature scores within a text. A sliding window can be conceived of as a window of size ws, which is defined by the number of sentences it contains. The window is moved across a text sentence-by-sentence, computing one value per window for a given indicator. In the present study, the ws was set to 1. The series of measurements generated by CoCoGen captures the progression of language performance within a text for a given indicator and is referred here to as a 'complexity contour' (see Figure 1 for illustration). CoCoGen uses the Stanford CoreNLP suite [20] for performing tokenization, sentence splitting, part-of-speech tagging, lemmatization and syntactic parsing (Probabilistic Context Free Grammar Parser [21]).

Table 1: *Descriptive statistics of (dis)fluency measures*

	AD patients		Control	
(Dis)Fluency measure	M	SD	M	SD
Speed of articulation				
Mean syllable duration	0.28	0.05	0.26	0.03
Syllables per minute	205	45.7	224	35.7
Silent pauses				
Pause time per sentence (in sec)	0.92	0.89	0.63	0.49
N long pauses ($> 2sec$)	1.28	2.22	0.473	0.71
N short pauses ($< 2sec$)	13.2	9.28	15.4	11.7
Filled pauses				
N *uh*	0.29	0.88	0.24	0.54
N *um*	0.07	0.37	0.31	0.74
Pronunciation				
Mean ASR confidence	0.83	0.09	0.86	0.08

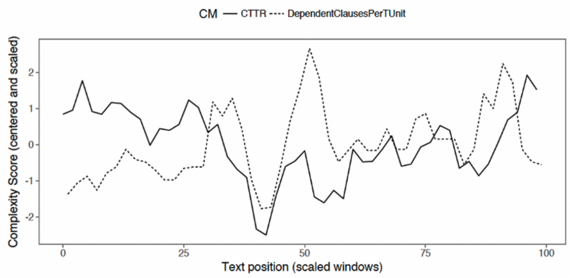

Figure 1: *Schematic representation of 'complexity contours' for two out of 293 complexity measures (CM) investigated: CTTR (Corrected Type Token Ratio) and Dependent Clauses per TUnit). Centering/scaling was applied here only for purposes of illustration.*

2.5. Pretrained Language Models

Since their inception, transformer-based pretrained language models such as BERT [22] and ERNIE [23] have achieved state-of-the-art performance in various classification tasks.The results of previous research demonstrate that the language characteristics of AD too can be captured by pretrained language models fine-tuned to the task of AD classification (see above). In this paper, pretrained BERT and ERNIE models were fine-tuned for the AD classification task and combined with classifiers trained on complexity and (dis)fluency features (see Section 3). Each of the 161 speakers in the training data is considered as a data point. The input of the model consists of all the text sequences of each speaker obtained by the ASR system, and the output is the class of the corresponding speaker, 0 for Control and 1 for AD.

[2] https://www.apptek.com/

[3] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

3. Experimental Setup

In this section we describe the component models used in our approach and how they were combined. To assess the performance of each model, 5-fold cross validation was used.

3.1. CNN Complexity + (Dis)Fluency Models

In order to make optimal use of the complexity and (dis)fluency features, which are sequential in nature, we built convolutional neural network (CNN) models. Originally proposed in computer vision, CNNs have been successfully adapted to various NLP tasks [24] and sentence classification tasks [25][26][27]. The CNN model has the advantage over models that rely on aggregated features, e.g. mean feature values, in that it is capable of capturing patterns in a feature sequence. We followed the approach proposed by [26], but replaced the word embedding with the concatenation of complexity and (dis)fluency features. Due to the small size of the dataset, we set the size of filters to be $2 \times d$, $3 \times d$, $4 \times d$ where d is the input feature dimension. Eight filters were used for each of the three filter types.

3.2. Fine-tuned BERT and ERNIE Models

The Huggingface Transformers library [28] was adopted for fine-tuning pretrained language models. Bert-for-Sequence-Classification was used and initialized with 'bert-base-uncased' and 'nghuyong/ernie-2.0-en' as our pretrained BERT and ERNIE model, respectively. In both cases, the base model was used rather than the large one, as preliminary experiments revealed no reliable differences in terms of classification accuracy between the two models on our dataset. Both models consists of 12 Transformer layers with hidden size 768 and 12 attention heads. The following hyperparameters were used for fine-tuning: the learning rate was set to 2×10^{-5} with 50 warmup steps and l_2 regularization set to 0.1. The maximum sequence length for both models was set to 256. For both models, default tokenizers were used.

3.3. Use of Ensembling Methods

Previous research on predicting AD using pretrained language models has demonstrated that their predictions based on fine-tuning for downstream tasks with a small dataset tend to be brittle and subject to high variance. To reduce this variance, we used an adapted version of the ensembling approach proposed in [9]: Each of the models described above was trained 50 times ($N = 50$). During the prediction phase, each model instance independently generated a prediction. The final classification decision was then determined by hard-voting, i.e. each model contributed its class prediction as a vote and the class that receives the majority of the votes was returned by the ensemble model. Besides using ensemble methods so as to reduce the variance in the prediction of a model, we also employed them to integrate information from different models. To this end, we performed experiments with two types of ensemble based methods, which are referred to here as *ensembling by bagging* and *ensembling by stacking*. Bagging involves fitting several independent models and pooling their predictions in order to obtain a model with a lower variance, while stacking involves combining the models by training a meta-model to output a prediction based on the different models predictions (see below). In each of the combined models, we used the same hyperparameter settings as stated above.

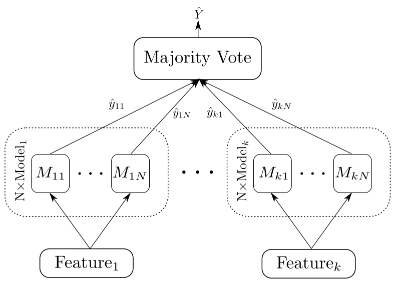

Figure 2: *Structure diagram of Model A. During training, we train each of the k models N times. During inference, jth instance of model i gives prediction $\hat{y}_{ij}$ independently. The final output of the ensembled model $\hat{Y}$ is the label, which the majority of the $k \times N$ model instances agree upon.*

3.3.1. Model A: Ensembling by bagging via majority vote

Ensembling by bagging via majority vote has been shown to be a simple yet effective method to increase the performance of classification models [29][30]. The first classification model (Model A) employed majority voting among 50 CNNs that used complexity and (dis)fluency features and 50 ERNIE models (see Figure 2). That is, as specified above, in this approach, each model was first trained/fine tuned 50 times, meaning that the final classification was based on 100 model instances. The classification in the Model A approach was then determined by counting the votes for each class (AD and controls (CN)) and choosing the more frequent class as the predicted one.

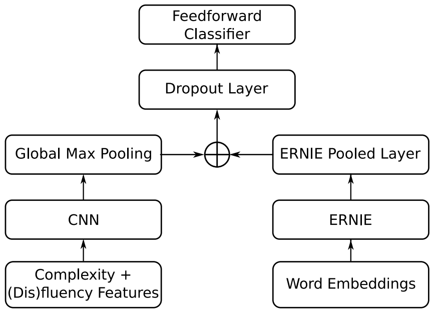

Figure 3: *Structure diagram of Model B.*

3.3.2. Model B: Ensembling by bagging using feature fusion

The second model (Model B) combined a CNN and a ERNIE model, which has previously been shown to perform better than either model alone [31]. Following the approach of [31], we built a model in which complexity and (dis)fluency information was first concatenated at the feature-level and subsequently fed into a CNN (see Figure 3). The hidden vector coming from CNN is then concatenated with the pooled output vector for the [CLS][4] token of Ernie model. The concatenated vector will serve as the input of a feed forward classifier on top of CNN and Ernie. To train this model, we first fine-tune ERNIE model. Then we freeze the parameters of the ERNIE model and jointly train the CNN model and feedforward classifier.

3.3.3. Model C: Ensembling by stacking

The final model, Model C, used in our experiments employed a stacking approach to ensemble all models [32], which has been

[4] [CLS], stands for classification, is a special token added in front of every input samples of BERT/ERNIE model to represent sample-level classification [22].

Table 2: *Mean accuracy (with standard deviations), precision, recall and F1 scores over a 5 fold cross-validation*

		Precision		Recall		F1	
Model	Acc	CN	AD	CN	AD	CN	AD
CNN Comp	M (SD)	M (SD)	M (SD)	M (SD)	M (SD)	M (SD)	M (SD)
CNN[Comp+DisFl]	0.80 (0.06)	0.79 (0.06)	0.81 (0.08)	0.78 (0.07)	0.83 (0.06)	0.78 (0.05)	0.82 (0.07)
Bert-Base	0.79 (0.06)	0.77 (0.09)	0.84 (0.08)	0.81 (0.11)	0.78 (0.12)	0.78 (0.06)	0.80 (0.07)
Ernie-Base	0.80 (0.04)	0.80 (0.08)	0.81 (0.04)	0.77 (0.07)	0.83 (0.09)	0.78 (0.04)	0.82 (0.05)
Model A: CNN[Comp+DisFl]+[Ernie] (*sep mod, bagging*)	0.76 (0.07)	0.61 (0.13)	0.88 (0.05)	0.79 (0.08)	0.74 (0.08)	0.68 (0.10)	0.80 (0.06)
Model B: CNN[Comp+DisFl]+[Ernie] (*fusion, bagging*)	0.83 (0.06)	0.75 (0.11)	0.89 (0.04)	0.83 (0.09)	0.82 (0.06)	0.78 (0.09)	0.85 (0.04)
Model C: LR[Comp]+LR[DisFl]+[Ernie]+[Bert] (*stacking*)	**0.83** (0.07)	0.82 (0.10)	0.85 (0.09)	0.83 (0.10)	0.84 (0.09)	0.82 (0.08)	0.84 (0.07)

Table 3: *Performance of the three ensemble models on test set*

		Precision		Recall		F1	
Model	Acc	CN	AD	CN	AD	CN	AD
Model A: CNN[Comp+DisFl]+Ernie(*sep mod, bagging*)	0.79	0.77	0.81	0.83	0.74	0.80	0.78
Model B: CNN[Comp+DisFl]+Ernie (*fusion, bagging*)	0.75	0.73	0.77	0.81	0.69	0.76	0.72
Model C: LR[Comp]+LR[DisFl]+Ernie+Bert (*stacking*)	**0.83**	0.82	0.85	0.86	0.80	0.84	0.82

shown to effectively increase the accuracy of the ensembled individual models. Specifically, we employed model stacking to combine two logistic regression models (LR) using complexity and (dis)fluency features respectively, and the two pretrained language models, i.e. BERT and ERNIE. The training procedure consists of two stages (see Figure 4). First, in stage one, each of the four models is trained/fine-tuned independently using 5-fold cross-validation (CV). For each sample in the test fold, we obtain one prediction vector from each of the four models (Models 1 to 4). These predictions vectors are then concatenated and constitute the input data in a subsequent stage (stage 2). The final predictions of Model C are derived from another logistic regression model trained on the concatenated prediction vectors from stage 1. To perform inference on the test set, we take the predictions from all model instances trained in stage 1 and average them by model, which will served as input of stage 2 after concatenation. All hyperparameters for the training/fine-tuning of each of the ensembled models were selected as above.

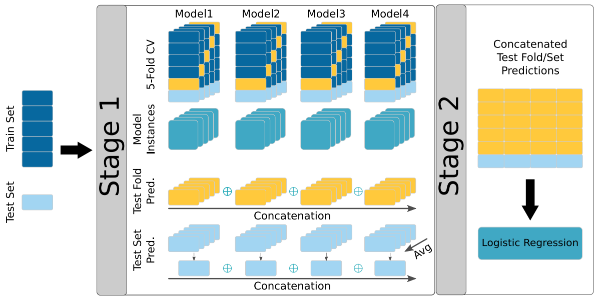

Figure 4: *Schematic representation of ensembling by stacking.*

4. Evaluation

In this section, we present our results on the AD detection task. The evaluation metrics for detection (accuracy, precision, recall, and F1 score) on the cross-validation (CV) set are presented in Table 2. The results on the evaluation set are shown in Table 3. As indicated by boldface numbers, the best performing model in both cross-validation (mean accuracy = 83.16%) and testing (accuracy = 83.10%) was Model C, i.e. the model that combined complexity and (dis)fluency features with both pretrained language models using stacking. Model B, which combined a CNN trained on utterance-level complexity and (dis)fluency features with the best performing fine-tuned pretrained language model (ERNIE) using late fusion and ensembling by bagging, fell close behind reaching 82.7% accuracy in CV. Model A, which combined the same features using majority voting with separate classifiers, performed below the accuracy levels of its component models, reaching 75.69% accuracy in CV. On the test set, the accuracy score of 83.1% of the best performing model, Model C, constitutes an improvement by 4.23% over the baseline model, which was based on fusion of linguistic and acoustic features [8]. Surprisingly, the relative performances of Model A and Model B were reversed on the test set, with Model A matching the performance of the baseline exactly (accuracy = 78.87%) and Model B falling just short of that (accuracy = 74.65%). The considerable discrepancies between the CV and test set classification accuracy for these models suggest that they suffer from overfitting. In contrast, Model C, which employed the stacking technique, performed equally well on CV and test data, indicating that it is robust against overfitting.

5. Discussion and Conclusion

The work presented here combined linguistic complexity and (dis)fluency features with pretrained language models for the task of Alzheimer's disease detection. An accuracy of 83.1% was achieved on the test set, which amounts to an improvement of 4.23% over the baseline model, which was based on fusion of linguistic and acoustic features. Our best performing model combined component models using a stacking ensemble technique. A key finding of this study is that incorporating information on linguistic complexity and (dis)fluency improved the performance of fine-tuned pretrained language models in AD classification by 3%, suggesting that different component models encode complementary information regarding the characteristic language patterns of AD. Another important aspect of our results is that the ensemble model trained on 'complexity contours', i.e. utterance-level measurements of human-interpretable complexity and fluency features, was able to match the performance of both fine-tuned pretrained BERT-like language models: Using 5-fold cross-validation with ensembling of 50 models in each fold, we obtained robust performance scores (≈ 80%) for both types of models. This finding has important implications in light of increasing calls for moving away from black-box models towards white-box (interpretable) models for critical industries such as healthcare, finances and news industry [33, 34].

6. References

[1] M. P. Mattson, "Pathways towards and away from alzheimer's disease," *Nature*, vol. 430, no. 7000, pp. 631–639, 2004.

[2] J. Zeisel, K. Bennett, R. Fleming *et al.*, "World alzheimer report 2020: Design, dignity, dementia: Dementia-related design and the built environment," 2020.

[3] S. de la Fuente Garcia, C. Ritchie, and S. Luz, "Artificial intelligence, speech, and language processing approaches to monitoring alzheimer's disease: A systematic review," *Journal of Alzheimer's Disease*, no. Preprint, pp. 1–27, 2020.

[4] S. Luz, "Longitudinal monitoring and detection of alzheimer's type dementia from spontaneous speech data," in *2017 IEEE 30th International Symposium on Computer-Based Medical Systems (CBMS)*. IEEE, 2017, pp. 45–46.

[5] E. L. Campbell, R. Y. Mesía, L. Docío-Fernández, and C. García-Mateo, "Paralinguistic and linguistic fluency features for alzheimer's disease detection," *Computer Speech & Language*, vol. 68, p. 101198, 2021.

[6] P. Pastoriza-Dominguez, I. G. Torre, F. Dieguez-Vide, I. Gomez-Ruiz, S. Gelado, J. Bello-Lopez, A. Avila-Rivera, J. Matias-Guiu, V. Pytel, and A. Hernandez-Fernandez, "Speech pause distribution as an early marker for alzheimer's disease," *medRxiv*, pp. 2020–12, 2021.

[7] R. S. Bucks, S. Singh, J. M. Cuerden, and G. K. Wilcock, "Analysis of spontaneous, conversational speech in dementia of alzheimer type: Evaluation of an objective technique for analysing lexical performance," *Aphasiology*, vol. 14, no. 1, pp. 71–91, 2000.

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

[9] J. Yuan, Y. Bian, X. Cai, J. Huang, Z. Ye, and K. Church, "Disfluencies and fine-tuning pre-trained language models for detection of alzheimer's disease," *Proc. Interspeech 2020*, pp. 2162–2166, 2020.

[10] M. S. S. Syed, Z. S. Syed, M. Lech, and E. Pirogova, "Automated screening for alzheimer's dementia through spontaneous speech," *INTERSPEECH (to appear)*, pp. 1–5, 2020.

[11] A. Balagopalan, B. Eyre, F. Rudzicz, and J. Novikova, "To bert or not to bert: Comparing speech and language-based approaches for alzheimer's disease detection," *arXiv preprint arXiv:2008.01551*, 2020.

[12] J. S. Guerrero-Cristancho, J. C. Vásquez-Correa, and J. R. Orozco-Arroyave, "Word-embeddings and grammar features to detect language disorders in alzheimer's disease patients," *TecnoLógicas*, vol. 23, no. 47, pp. 63–75, 2020.

[13] B. Mirheidari, D. Blackburn, T. Walker, A. Venneri, M. Reuber, and H. Christensen, "Detecting signs of dementia using word vector representations." in *INTERSPEECH*, 2018, pp. 1893–1897.

[14] J. B. Orange, R. B. Lubinski, and D. J. Higginbotham, "Conversational repair by individuals with dementia of the alzheimer's type," *Journal of Speech, Language, and Hearing Research*, vol. 39, no. 4, pp. 881–895, 1996.

[15] Y. Pan, B. Mirheidari, M. Reuber, A. Venneri, D. Blackburn, and H. Christensen, "Improving detection of alzheimer's disease using automatic speech recognition to identify high-quality segments for more robust feature extraction," *Proc. Interspeech 2020*, pp. 4961–4965, 2020.

[16] E. Kerz, Y. Qiao, D. Wiechmann, and M. Ströbel, "Becoming linguistically mature: Modeling english and german children's writing development across school grades," in *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, 2020, pp. 65–74.

[17] Y. Qiao, D. Wiechmann, and E. Kerz, "A language-based approach to fake news detection through interpretable features and brnn," in *Proceedings of the 3rd International Workshop on Rumours and Deception in Social Media (RDSM)*, 2020, pp. 14–31.

[18] M. Ströbel, E. Kerz, and D. Wiechmann, "The relationship between first and second language writing: Investigating the effects of first language complexity on second language complexity in advanced stages of learning," *Language Learning*, vol. 70, no. 3, pp. 732–767, 2020.

[19] M. H. Christiansen and N. Chater, "Towards an integrated science of language," *Nature Human Behaviour*, vol. 1, no. 8, Jul. 2017.

[20] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. McClosky, "The stanford corenlp natural language processing toolkit," in *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, 2014, pp. 55–60.

[21] D. Klein and C. D. Manning, "Accurate unlexicalized parsing," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2003, pp. 423–430.

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

[23] Y. Sun, S. Wang, Y. Li, S. Feng, H. Tian, H. Wu, and H. Wang, "Ernie 2.0: A continual pre-training framework for language understanding," *arXiv preprint arXiv:1907.12412*, 2019.

[24] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of machine learning research*, vol. 12, no. ARTICLE, pp. 2493–2537, 2011.

[25] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," *arXiv preprint arXiv:1404.2188*, 2014.

[26] Y. Kim, "Convolutional neural networks for sentence classification," 2014.

[27] M. Ma, L. Huang, B. Xiang, and B. Zhou, "Dependency-based convolutional neural networks for sentence embedding," *arXiv preprint arXiv:1507.01839*, 2015.

[28] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45. [Online]. Available: https://www.aclweb.org/anthology/2020.emnlp-demos.6

[29] C. Costello, R. Lin, V. Mruthyunjaya, B. Bolla, and C. Jankowski, "Multi-layer ensembling techniques for multilingual intent classification," 2018.

[30] N. C. Oza and K. Tumer, "Classifier ensembles: Select real-world applications," *Information fusion*, vol. 9, no. 1, pp. 4–20, 2008.

[31] I. Alghanmi, L. Espinosa-Anke, and S. Schockaert, "Combining bert with static word embeddings for categorizing social media," 2020.

[32] D. H. Wolpert, "Stacked generalization," *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.

[33] C. Rudin, "Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead," *Nature Machine Intelligence*, vol. 1, no. 5, pp. 206–215, 2019.

[34] O. Loyola-Gonzalez, "Black-box vs. white-box: Understanding their advantages and weaknesses from a practical point of view," *IEEE Access*, vol. 7, pp. 154 096–154 113, 2019.