Fantastic Features and Where to Find Them: Detecting Cognitive Impairment with a Subsequence Classification Guided Approach

Benjamin Eyre[1], Aparna Balagopalan[1], Jekaterina Novikova[1]

[1]Winterlight Labs, Toronto, Canada

1. Introduction

Despite the impact of word and sentence embeddings on NLP, feature engineering remains an important practice in several domains where interpretability is important, such as healthcare.

- Using embeddings as input can lead to issues with interpretability [1], while feature engineering approaches lend themselves to being more easily interpreted.
- *Is there a way to supplement the current suites of engineered features for predicting cognitive impairment while reducing the amount of time and resources spent on excess feature engineering?*

Cognitive Impairment Detection:

- There is a wealth of literature that details different successful methods for detecting cognitive impairment that use engineered features [2].
- Cognitively impaired (CI) individuals and healthy (HC) individuals have been shown to display different pausing patterns when speaking [3].
- *Can we extract distinguishing, pause related transcript-level features while minimizing the noise added from unrelated factors?*

Solution: Use the efficacy of sequential machine learning models for finding patterns in sequences to determine which regions in a pause-centred sequence contain the most predictive information for detecting CI, and use this to avoid engineering excess features.

Our major contributions are:

- A method of classifying speech using only a token of interest and a small context around it: *subsequence classification*.
- A novel *feature engineering approach* guided by subsequence classification.
- Validating this approach by showing that it aids in achieving transcript-level classification results comparable to the state of the art on a standard data set of CI speech.

4. Proposed Feature Engineering Approach

1. **Construct Subsequences:** For each input transcript, subsequences of varying length centered around a token of interest, such as a pause, must be extracted. After extracting token-level features for each token in these subsequences, they should be grouped into subsets based on maximum length.
2. **Subsequence Classification:** A sequential ML model must be cross validated on each of the subsequence data subsets from the previous step in a subsequence classification experiment.
3. **Construct Transcript-Level Features:** Construct transcript-level aggregations of the token level features used in subsequence classification. These should only be extracted for the distances that produced the greatest cross validated accuracy during subsequence classification.

Goal: Use the cross validated accuracy from subsequence classification to determine where distinguishing information can most easily be extracted. Then, extract transcript-level features exclusively from these information rich regions in the transcript.

7. Conclusions

- We describe a novel method for text classification - subsequence classification - where text is modelled as a token interest with surrounding tokens of context.
- We demonstrate how subsequence classification can be used to engineer features that extract distinguishing information while minimizing added noise, and consequently match SOTA performance on a standard data set of CI speech.

2. Pause Sequence Modelling

- To conduct subsequence classification, we extract subsequences of varying length from each transcript from Dementiabank, a publicly available data set of spontaneous picture descriptions from subjects both with and without cognitive impairment.
- For each transcript, we extract each of the pauses, along with the tokens that are one, two, or three tokens away (DB-C1, DB-C2, DB-C3) from the pause when possible. A visualization of this is presented below. Our final data set, DB-Utt, includes every utterance from the Dementiabank transcripts that contained a pause. We refer to the Dist 1, Dist 2, and Dist 3 positions as D1, D2, and D3 respectively.

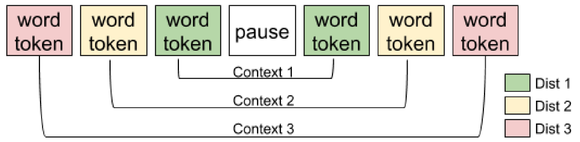

Figure: Visualization of the difference between contexts and distances in a pause-focussed sequence.

5. Results

Subsequence Classification: We present the average cross validated subsequence classification accuracy across 4 random seeds for each of the best performing hyperparameter configurations across each of the 4 subsequence data subsets.

Data Subset	D-C1	D-C2	D-C3	D-Utt
Accuracy	59.6±2.6	**60.7±2.5**	59.8±0.9	60.3±1.0

Table: Subsequence classification performance. Accuracy is averaged across four random seeds.

The superior performance on DB-C2 leads us to believe that using features from the two tokens preceding and succeeding a pause could enhance transcript-level classification performance.

Transcript Classification: To validate this feature engineering method, we create five transcript-level feature sets: features aggregated from tokens at the D1 position in reference to a pause (*F-D1*), features aggregated from the D2 position (*F-D2*), features aggregated from the D3 position (*F-D3*), the combination of F-D1, F-D2, and F-D3 (*F-C3*), and the combination of F-D1 and F-D2 (*F-C2*). We extend the *Original* feature with each of these newly engineered feature sets. We hypothesize that the greatest performance will be attained using F-D1, F-D2, or F-C2.
Presented is the cross-validated transcript classification accuracy achieved using the *Original* feature set, along with each newly engineered feature set. Results presented are for the most effective hyperparameter configuration for that feature set.

Feature Set	Model	Acc	Prec	Sens	Spec
Original	Ens	74.77±0.6*	82.08±0.4*	73.1±0.5*	79.74±0.4*
Original w/ feat.sel.	Ens	75.18±1.4	83.37±1.2	72.21±1.3*	81.67±1.6
Original + F-D1	NN	74.41±1.9*	78.59±1.1*	**77.15±3.3**	72.63±1.9*
Original + F-D2	Ens	**77.09±1.0**	**84.40±0.8**	75.21±1.0	82.32±0.9
Original + F-D3	Ens	76.05±0.8	84.02±0.9	71.92±0.7*	**83.33±1.3**
Original + F-C2	NN	75.14±1.4	79.85±1.5*	76.93±0.9*	74.02±2.5*
Original + F-C3	Ens	74.82±1.2*	83.68±1.2	70.62±1.9*	82.63±1.4

Table: Transcript classification performance for each feature set's best performing classification model, averaged across four random seeds. Bold indicates best performance, and * indicates significance ($p < 0.05$) when compared to the model using F-D2 features.

We observe that the highest transcript classification accuracy, which is 2.3% higher than baseline, is achieved by an ensemble model that used the *Original* + F-D2 feature set, as hypothesized. We also note that this model was able to achieve an accuracy of 78.36% using one of the four random seeds, the same as the single-seed SOTA accuracy of 78% [2].

3. Feature Extraction and Classification

For transcript classification: We extract a set of over 500 linguistic and acoustic features from each transcript in Dementiabank, such as part of speech counts and average word length. This *Original* feature set forms our baseline set of features for transcript classification.

To evaluate our feature engineering approach, we classify transcripts as CI or HC in 10 fold cross validation. We use the *Original* feature set as a baseline, and extend it with each of the feature sets produced by our feature engineering approach. 5 different models are tested using grid-search for each feature set: an SVM, a gradient boosting ensemble, a 2-layer neural network (NN), a random forest, and an ensemble of the previous four models (Ens)

Data Subset	HC	CI	Total
DB (transcripts)	229 (42%)	321 (58%)	550
DB-C1	317 (33%)	645 (67%)	962
DB-C2	511 (35%)	963 (65%)	1,474
DB-C3	529 (35%)	980 (65%)	1,509
DB-Utt	755 (42%)	1,059 (58%)	1,814

Table: Overview of the number of samples (subsequences or transcripts) in different subsets of DB.

For subsequence classification: A small suite of lexical features is extracted from each of the tokens in each of the subsequence based data subsets. Features include length of word, sentiment measures, age of acquisition, etc.

We perform 5 fold cross validation on each sequence based data subset. Sequences are classified as CI or HC using a GRU-based architecture with attention, with hyperparameters tuned for each data subset.

6. Discussion

To investigate how transcript classification and subsequence classification are connected, we conduct a statistical analysis on the token-level and transcript-level features.

Distance	Token-Level	Transcript-Level
D1	18	12
D2	21	12
D3	7	6

Table: Number of features that are significantly different between classes according to two sided t-tests for each distance.

Two sided t-tests between features extracted from tokens found at D1, D2, and D3 from different classes show similar patterns for features that are significantly different between classes for both the token and transcript-level. Larger concentrations of distinguishing features are found at D1 and D2 than at D3. This could explain the effectiveness of features from the D2 position in both tasks.

This does not explain why the *Original* + *F-D3* feature set is more effective than the *Original* + *F-D1* feature set in transcript classification. However, we note that trend of features from D3 leading to better performance than features than D1, but worse performance than features at D2, is consistent between both subsequence and transcript classification experiments.This may indicate that subsequence classification can provide valuable insight into potential transcript classification performance for different features.

References

[1] Kindermans, Pieter-Jan, et al. "The (un) reliability of saliency methods." Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Springer, Cham, 2019. 267-280.

[2] Hernández-Domínguez, Laura, *et al.* "Computer-based evaluation of Alzheimer's disease and mild cognitive impairment patients during a picture description task." Alzheimer's Dementia: Diagnosis, Assessment Disease Monitoring 10 (2018): 260-268.

[3] Pistono, Aurelie, *et al.* "Pauses during autobiographical discourse reflect episodic memory processes in early Alzheimer's disease." Journal of Alzheimer's disease 50.3 (2016): 687-698.