Designing Explainable Speech-Based Machine Learning for the Estimation of Job Interview Outcomes: A Comprehensive Approach

Alexander Paradise[1], Abhijeet Ganji[2], and Sri Vamsi Andavarapu[3]

[1]Email: alpa2994@colorado.edu
[2]Email: abga1338@colorado.edu
[3]Email: sran9587@colorado.edu

Abstract

Contents

1 Introduction

The selection process of a candidate after a job-interview involves a lot of factors. The interviewer accounts for various qualities the candidate displays throughout the interview. These can broadly be classified into three areas: Domain specific knowledge for gauging the technical rigor, interpersonal skills necessary to navigate challenges in workplace, and finally the confidence they give off during the interview process and the impression left on the hiring team. We are interested in the last factor and while often viewed as intangible or subjective, this impression is significantly shaped by how candidates speak during the interview. We aim to investigate how various speech elements—specifically syntactic, semantic, and prosodic features—contribute to the perceived confidence of candidates and potentially influence interview outcomes.

Syntactic features include the structure and complexity of sentences, the use of active vs. passive voice, and clarity in verbal expression. More structured and fluent speech is usually considered more competent.

Semantic features pertain to the meaning and content of speech such as the choice of diction, the relevance of examples, the use of appropriate analogies, and clarity of thought conveyed. Candidates who speak with precise, meaningful content are likely to be seen as more prepared and confident.

Prosodic features involve nuances like rhythm, intonation, pitch, speech rate, and pauses in speech. For example, speaking with a steady pace, appropriate intonation, and controlled pitch can signal calmness and confidence, whereas excessive filler words, uptalk, or erratic pacing may hint at being unsure.

2 Preprocessing and Feature Generation

2.1 Data Compilation

The first step of the project was to assess, extract, clean and organize the data. The full interview data set consisted of 138 separate interviews across 69 participants, with each participant having two interviews. The transcripts were recorded in full, and included the speaker for each statement. For each transcript there were two aggregate scores, one for the overall interview performance, and one for the overall excitement. Finally there was a third data set with prosodic features for each interview question.

To better analyze the data, we extracted all the transcripts, and created a dictionary keyed on participants to store both the first and second interviews. Along with the full transcript, we also stored the transcript cleaned of speaker tags, in order to facilitate unbiased sentiment analysis. We also separately stored each speakers individual transcript for analysis. Finally, the scores and prosodic features were stored along with the scores for each separate interview. This allowed us to move toward feature analysis, now that the data has been cleaned.

2.2 VADER Feature Generation

For our feature analysis, we decided on one easily interpretable method, and one less interpretable one. For the former, we decided to use scores created by using the VADER sentiment analysis model Hutto and Gilbert 2014.

At its most basic VADER uses a lexicon of words individually scored by multiple people to analyze the sentiment of any body of text. It then uses various heuristics to modify the scores, such as checking for negation. The analyzer then returns the proportion of positive, negative and neutral sentiment, as well as an overall compound score.

While feature extraction based on sentiment analyze often takes the form of keying onto specific words that seem most informative, we decided it may be interesting to look at the VADER scores in aggregate. Toward this end we generated the sentiment scores for each sentence for each

separate transcript we saved: we generated a set for the interviewee's transcript, and set of the interviewer's transcript This was done at the sentence level as VADER is better optimized for shorter strings of text, instead of at the document length. We then computed summary statistics for each transcript, including the mean, standard deviation, and coefficient of variation. The coefficient of variation defined as the ratio of the population mean and the standard deviation, was included in hopes of identifying how varied the sentence scores were on average. The hypothesis was that perhaps more extreme and contrasting scores from sentence to sentence would help the model identify excitement, or lack thereof. The standard deviation and coefficient were generated from the overall composite scores towards this end.

Along with the breakdown of positive, negative, neutral and composite scores for each speaker, we also wanted to ensure we included the full transcript. While we hoped that breaking down each interview into different speakers would be informative, (such as a negative interviewer tone leading to a bad interview score, or a neutral interviewee leading to a low excitement score), it seemed very important to also include the average positive, negative, neutral and composite scores for each transcript. This left us with a total of 16 features: six for each speaker, and four for the transcript as a whole.

2.3 BERT Embedding

Along with the VADER analysis we also created a low dimensional embedding with the BERT model Devlin *et al.* 2019. We tokenized the data using the BERT tokenizer, and then generated the embedding using mean pooling. BERT took our high dimensional transcripts and embedded them in a length 768 vector.

Since 768 dimensions is a bit to many to visualize, we preformed Principle Component Analysis to project these vectors down into two dimensions. The hope was that in the principle component space we may be able to see some clustering by score, or other differentiations that could prove informative.

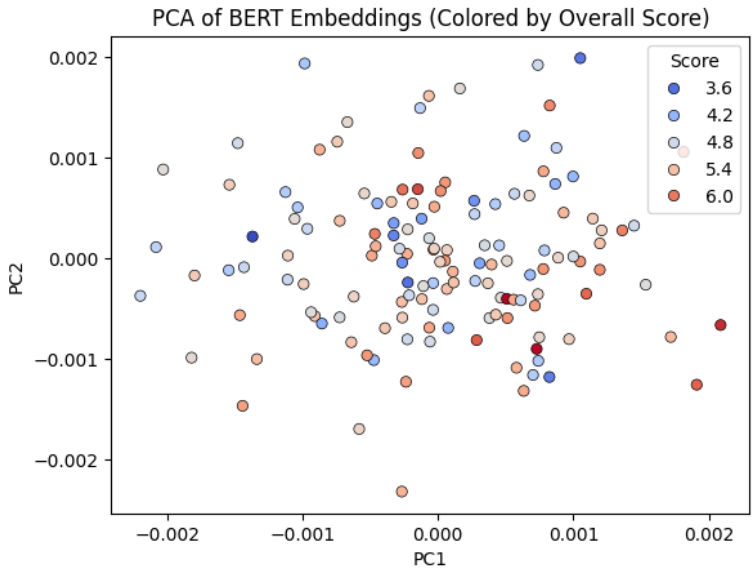


Figure 1. 768 dimensional BERT embedding projected onto the first two principle components for overall score

As it is somewhat clear from the images, our hopes proved unfounded. The first two principle components only accounted for about 11% and 8% of the variation respectively, and to the eye no clustering by score was visible. This however still presents some potential for follow up; perhaps

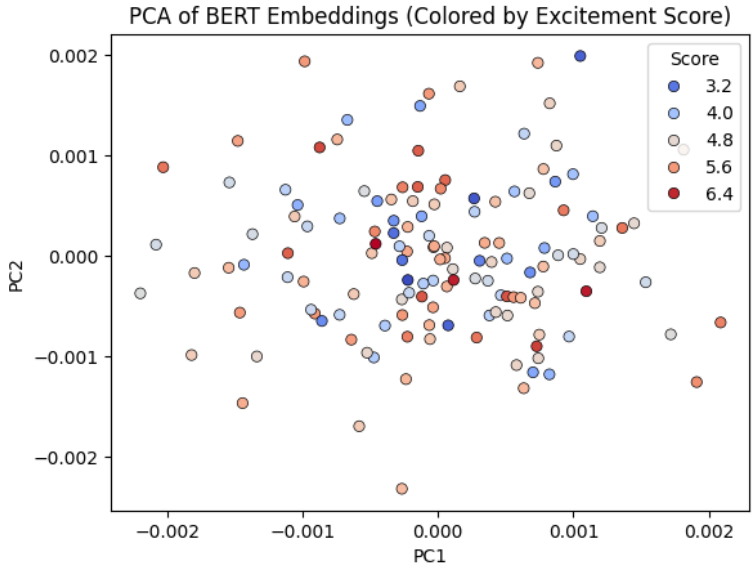


Figure 2. 768 dimensional BERT embedding projected onto the first two principle components for overall score

clustering does occur in higher dimensional space, which could be elucidated by k-means clustering or other unsupervised techniques.

3 Feature Selection

3.1 Methods

Now that our 16 sentiment features from VADER have been created, we now aimed to figure out which of them was most important. We ended up using three different metrics to study each features relative information gathering potential: Mutual Information, a binned Mutual Information and the Spearman correlation coefficient.

Mutual information can be thought of as how much information one gains about a random variable X given knowledge of a random variable Y Franz and Schölkopf 2006. A mutual information of zero implies that the variables are independent. The formula for the mutual information between two variables is given as:

$$I(X;Y) = E[P_{XY}] \log(\frac{P_{XY}}{P_X P_Y})$$

Where

$$E[P_{XY}]$$

represents the expected value of the joint distribution of X and Y. We had hoped that the mutual information would let us know which features gave us the most information about the scores. Due to the small variation in the scores, we also binned each section of scores uniformly, in hopes that this discretized version may handle the small variations better.

Finally, since mutual information does not provide a direction, and only a magnitude of association, we decided to examine the Spearman correlation coefficient. The spearman coefficient works very similarly to the Pearson correlation, but instead is based on rank, instead of the mean of variables Laerd Statistics 2023. Its formula is given as:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

This allows Spearmen to capture not only linear relationships, but all monotonic relationships between two variables. Since we don't know, or even have a very good guess, what relationships will be present between the VADER scores, it seemed the pragmatic choice for a directional metric.

3.2 Feature Importance

With our metrics established, we then computed the various correlation coefficients for each of the 16 features we previously determined.

Feature	Spearman (Overall)	MI (Overall)	Binned MI (Overall)
trans_neut_mean	0.282	0.041	0.024
interviewee_neut_mean	0.176	0.000	0.012
interviewee_comp_mean	0.140	0.000	0.019
interviewer_neut_mean	0.081	0.000	0.018
interviewer__comp_std	0.079	0.020	0.061
interviewer_comp_mean	0.059	0.035	0.098
trans_comp_mean	0.007	0.000	0.008
interviewer_comp_coefvar	-0.038	0.000	0.075
interviewer_pos_mean	-0.080	0.000	0.032
interviewee__comp_std	-0.087	0.006	0.000
interviewee_neg_mean	-0.098	0.038	0.090
interviewer_neg_mean	-0.115	0.065	0.073
interviewee_pos_mean	-0.133	0.000	0.013
interviewee_comp_coefvar	-0.139	0.000	0.012
trans_neg_mean	-0.145	0.084	0.060
trans_pos_mean	-0.236	0.154	0.127

Table 1. Correlation and Mutual Information Scores for Overall Score Prediction Features

Feature	Spearman (Excitement)	MI (Excitement)	Binned MI (Excitement)
trans_comp_mean	0.241	0.107	0.000
interviewee_comp_mean	0.160	0.000	0.006
interviewer_neut_mean	0.108	0.028	0.000
trans_pos_mean	0.025	0.145	0.079
interviewee_pos_mean	0.020	0.000	0.000
interviewee_neut_mean	0.013	0.000	0.000
interviewer_comp_mean	0.011	0.044	0.000
trans_neut_mean	0.003	0.061	0.016
interviewer_comp_coefvar	-0.031	0.044	0.025
interviewee_neg_mean	-0.057	0.011	0.102
trans_neg_mean	-0.067	0.018	0.000
interviewee__comp_std	-0.077	0.076	0.032
interviewer__comp_std	-0.079	0.047	0.098
interviewer_pos_mean	-0.099	0.000	0.011
interviewer_neg_mean	-0.148	0.036	0.024
interviewee_comp_coefvar	-0.159	0.055	0.038

Table 2. Correlation and Mutual Information Scores for Excitement Prediction Features

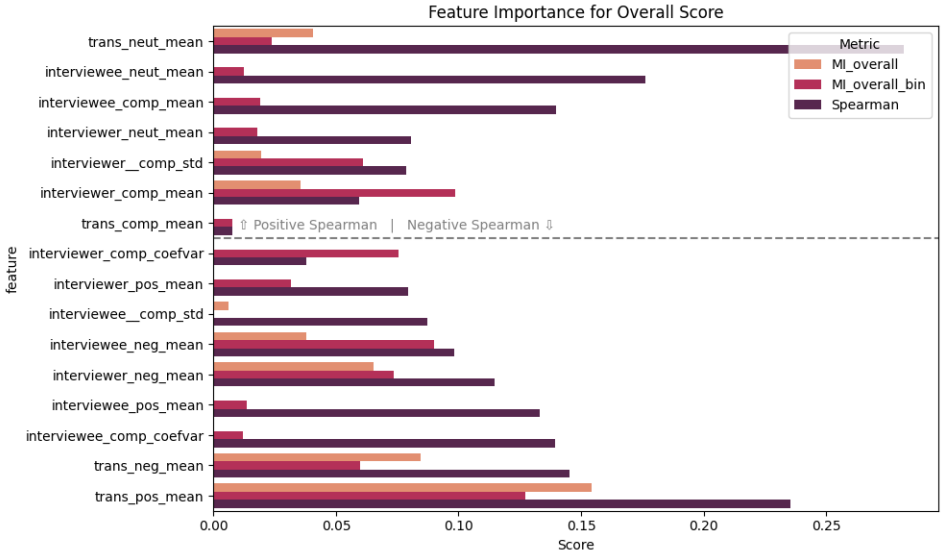


Figure 3. Feature importance based on different metrics for overall score ranked by the value of the Spearman correlation

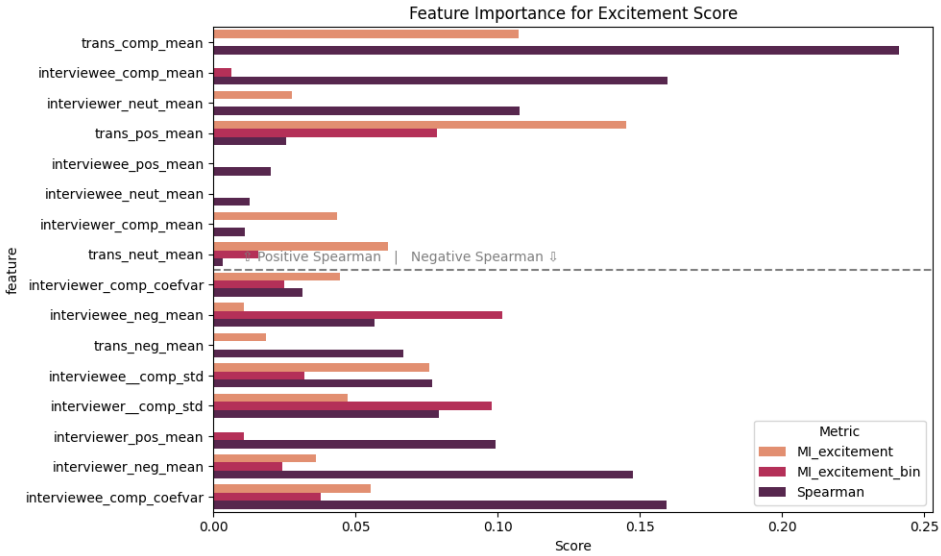


Figure 4. Feature importance based on different metrics for excitement score ranked by the value of the Spearman correlation

Interestingly enough, while the MI and Spearman agree generally, they are not a perfect match in what they consider most important. There appears to especially be a discrepancy in the overall scores, which may imply some non-linear features lurking below the surface. However given the directionality, we chose to use Spearman for our general feature selection.

Another thing of note is that by far the most positively correlated metric was the proportion of neutral words, and the most negatively correlated were the proportions of both positive and negative words. This implies that less emotional, less dynamic interviews were consistently rated higher. This implies that candidates that were more straight to the point, and weren't perceived as negative, or perhaps disingenuously positive, had better scores overall: a potential insight into best practices for interviews going forward.

4 ML Models

4.1 Approach

We used the semantic features extracted from Vader to compare Absolute Relative Error and Pearson's correlation r for Overall Score metric and Excitement Score metric, across different number of top-k selected features using Spearman's correlation. We used XGBoost as the tree-based model with hyperparameters set as no_estimators = 300, max_depth = 6, learning_rate = 0.05, and a Multilayered Perceptron(FNN) as the deep learning model with 2 hidden layers consisting of 64 neurons and 32 neurons respectively, and the relu activation function. To see how good our model performs compared to a blind guess, we calculate the pearson's correlation and relative error and compare them with the baseline which uses a constant function as a prediction for all the test samples. We used the sample mean of the train samples for obtaining the baseline.

4.2 Findings

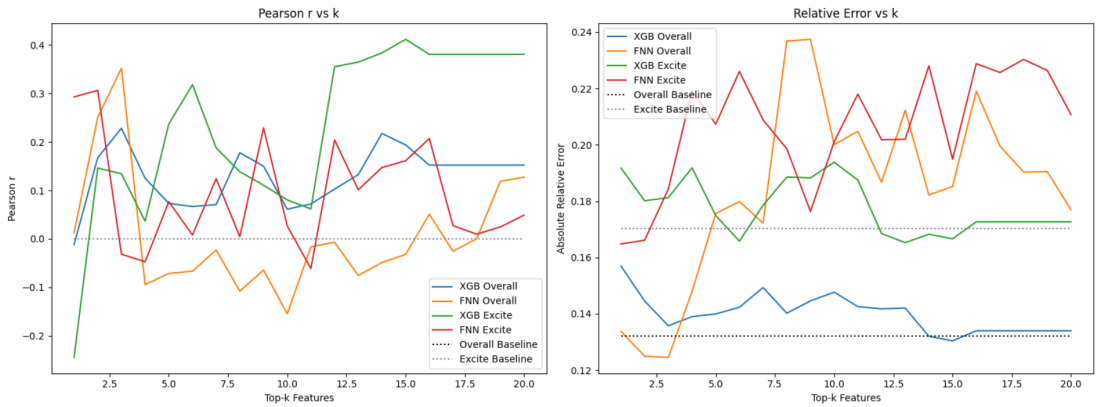


Figure 5. XBG and FNN Performance on Overall Score and Excitement score using semantic features from Vader, as predictors

On average, XGB for Excitement shows the highest correlation whereas FNN for Overall shows the lowerst. At around 15 features the relative error for XGB predictions for both overall and excitement are below the respective baseline and go above it as soon as features are added. FNN predictions are always above the respective baselines.

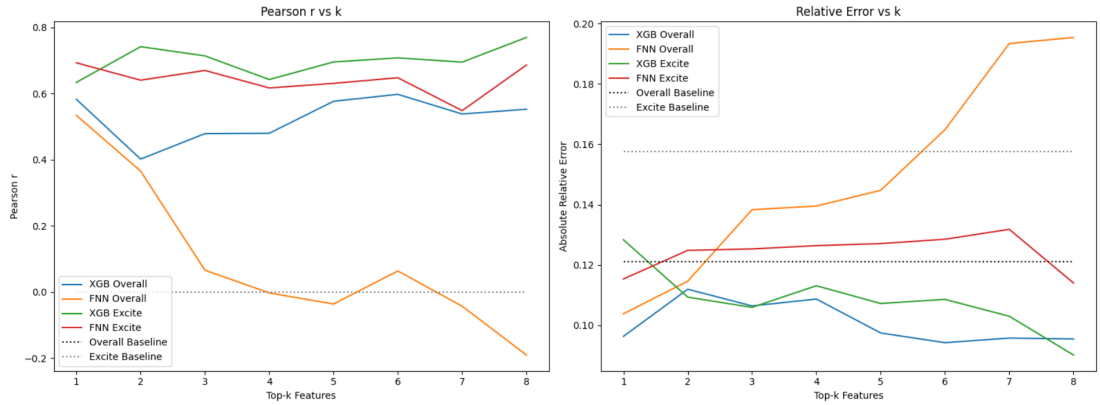


Figure 6. XBG and FNN Performance on Overall Score and Excitement score using prosodic features, as predictors

We performed the same procedure using Prosodic features. On average, the correlation is in the following descending order: XGB for Excite, FNN Excite, XGB Overall and FNN Overall. Regarding the

relative error, the performace of XGB is rather good, almost staying below the baseline throughout the feature span. FNN for excite is always below the baseline but FNN for overall is only below the baseline for the top 2 features but goes above from 3 and keeps on increasing with successive addition of feature.

5 Multimodal Comparision

5.1 Approach

We compared performance of FNN and XGB for the two metrics across three different feature sets; only semantic features, only prosodic features, combined set of semantic and prosodic features.

5.2 Findings

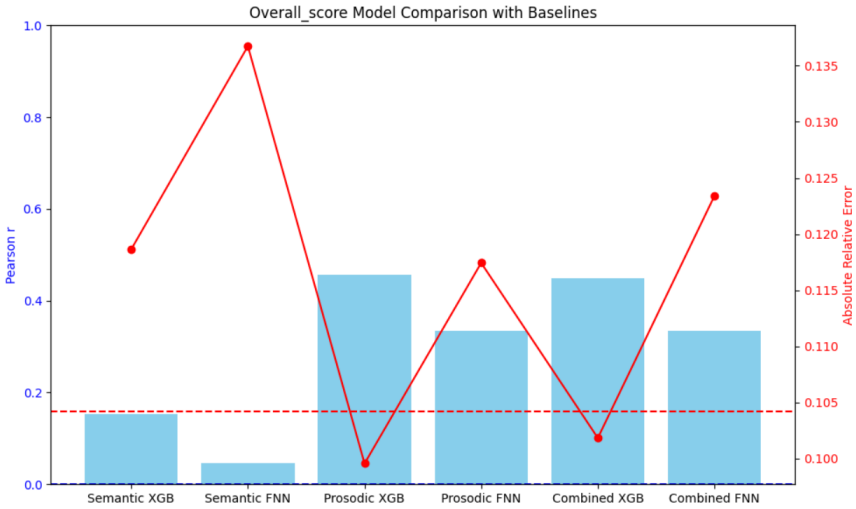


Figure 7. Multi-modal modeling comparing XGB and FNN across different set of features for Overall Score

For predicting Overall score, the best performer is XGB using only the prosodic set followed XGB using the combined set of features, both of which are below the baseline for relative error. All the three combinations of feature sets used for FNN are above the baseline

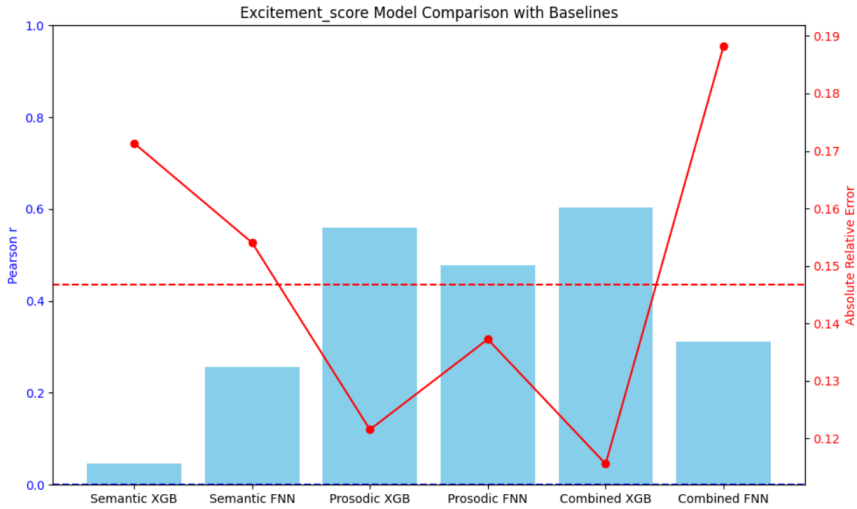


Figure 8. Multi-modal modeling comparing XGB and FNN across different set of features for Excitement Score

Coming to Excitement score, the three best models falling below the baseline are XGB using the combined feature set followed by XGB using the prosodic set alone and lastly FNN using just the prosodic set. The results of our findings state XGB as a better predictor compared to FNN and the prosodic features are much better features when it comes to predicting job-interview performance metrics.

6 Explainable AI

6.1 Approach

To interpret the predictions made by the models developed in Part (c), we used two powerful explainability techniques: SHAP for the tree-based XGBoost model, and LIME for the deep learning Feedforward Neural Network (FNN).

6.2 SHAP

We chose SHAP because it provides both global and local explanations by attributing the output prediction to each feature's contribution. In our case, we used it to interpret the overall interview performance scores predicted by the XGBoost model.

The x-axis indicates the model output value; moving right increases the value and left decreases it. The color shows the actual feature value, from blue (low) to red (high).

The SHAP summary plot revealed that features such as intensityQuant, trans_pos_mean and pitchUvsVRatio had the highest positive impact on performance prediction. Features like interviewer_neg_mean pulled the score lower, escpecially when values were high - indicating that when the interviewer was more negative, it impacted the outcome negatively.

Figure 9. SHAP for XGB: Overall Score

6.3 LIME

We used LIME to generate local explanations for individual predictions made by the FNN model predicting interviewee excitement. It helped us understand which features most influenced a single excitement prediction.

The LIME summary plot revealed that intensitySD had a strong positive contribution, suggesting consistent vocal energy led to higher excitement predictions. intensityMean, avgBand2, and PercentBreaks were negative contributors, indicating that flat tone and frequent pauses may signal lower excitement.

The predicted excitement score is 5.58, which lies near the higher end of the scale. On the right the orange bars show the positive contributions, and on the left, blue bars show negative contributions to the predicted value.

Despite having the lower value, intensitySD strongly increased the prediction - possibly suggesting that a consistent energy level in speech is associated with high excitement. But on the other hand, low intensitymean and high avgBand2 negatively impacted the score - possibly indicating flat tone.

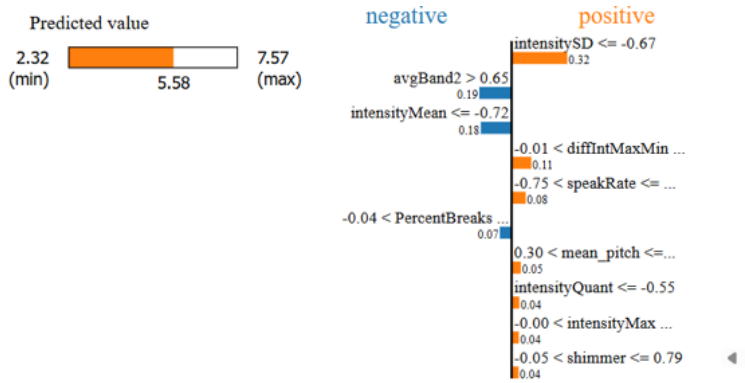


Figure 10. LIME for FNN: Excitement Score

This table shows the actual values of prosodic features that LIME used to explain the FNN's predicted excitement scores. We can see that the interviewee had very low average and standard deviation in intensity, along with a high number of breaks in speech. These suggest a low-energy delivery, which matches the model's output. LIME used these actual values to determine how each feature contributed to the prediction.

Feature	Value
intensitySD	-1.05
avgBand2	0.66
intensityMean	-0.97
diffIntMaxMin	0.14
speakRate	-0.68
PercentBreaks	0.51
mean_pitch	0.41
intensityQuant	-1.02
intensityMax	0.15
shimmer	0.47

Figure 11. Feature Value Table

7 Large Language Models

As a final method of evaluation, we decided to use the relatively new power of Large Language Models, or LLMs, to predict interview scores. With the meteoric rise of text generators like ChatGPT or Gemini in the last few years, the potential applications of LLMs have expanded drastically. With the rapid pace of AI advancement, it seemed likely that LLMs could predict interview scores with accuracy rivaling that of deep learning based models.

We first attempted to implement some of the transformer based models from the Hugging Face library, specifically, GPT-2, LlaMa, and Llama2. Our model input included both instructions for output as well as some examples. We used a few-shot approach, feeding it two or three training examples, in order for it to get an idea of what the scores should look like for differing transcripts. We then appended the full transcript of the interview we wished to evaluate.

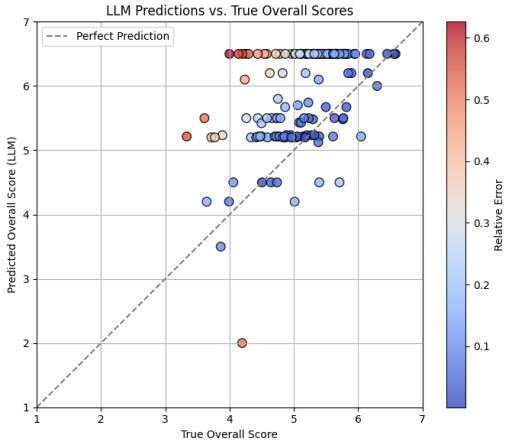
While powerful models in the right context, these methods proved entirely incapable of predicting interview scores with any modicum of consistency. Our outputs were plagued by constant hallucinations, with the models attempting to generate new transcripts, or sometimes outputting seemingly random text. When it did manage to output a score, it typically just repeated the score of the training example back to us.

We tried modifying out prompts, our examples, our instructions, and even shortening our transcripts to no avail. No matter what we tried, we could not get a consistent output for any one transcript, and more often then not, the model didn't return a numerical score at all. In general, however, this is not too surprising. The LLMs we have available through Hugging Face are much, much smaller and less powerful than SOTA. As such, it seems likely we were simply overwhelming it with the amount of text we wished to process, confusing the model, and leading to hallucinations. While with some fine tuning and more careful model selection it may be possible to get good results with Hugging Face, we decided to instead throw more computational power at the problem, in hopes of a superior result.

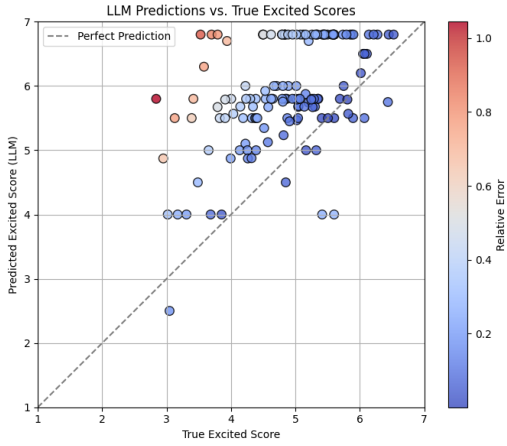
OpenAI allows for users to directly access the API for several of its models through python, through use of an API key. While this is not free, and as such could be prohibitive for large tasks, our small dataset seemed perfectly suited for it. In fact, running the model on our entire set cost about two cents per attempt. We made use of GPT-40-mini OpenAI 2024, and preformed the same type of prompting. We tested both two and three-shot methods for comparison, and requested output as (Overall Score, Excitement Score, Reason). See below the sample output, generated for participant one, interview one.

"- Predicted Overall Score: 5.487 - Predicted Excitement Score: 5.672 - Explanation: The candidate demonstrates a solid understanding of their field and has relevant experience, particularly through leadership roles in fundraising and community service. They articulate their interests well and show initiative in overcoming challenges, such as addressing team dynamics and seeking help when needed. However, there are some areas where their responses could be more polished and confident, particularly regarding their weaknesses and technical preparedness. Overall, they present a good balance of enthusiasm and capability, which contributes to a strong performance score"

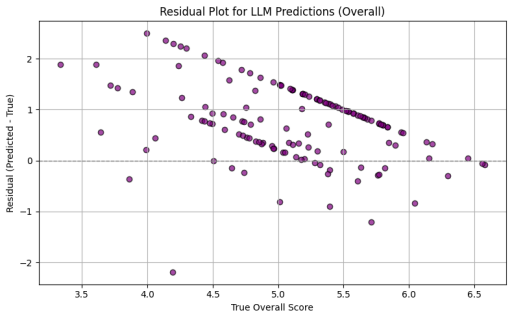
We then, one by one, fed each example into it, and obtained the following results.



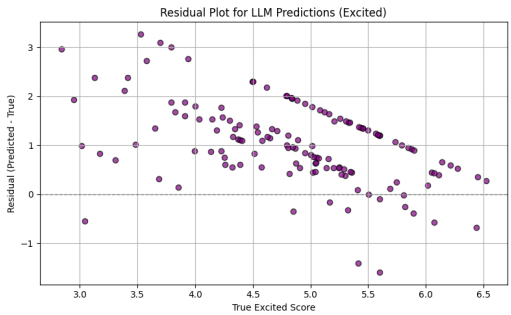
(a) 2-shot: LLM vs. True Overall Score. Pearson r:.451



(b) 2-shot: LLM vs. True Excitement Score. Pearson r:.454

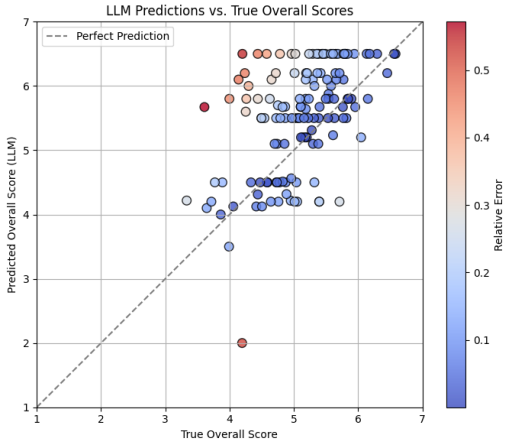


(c) 2-shot: LLM vs. True Overall Score Residuals. MAE: 0.886

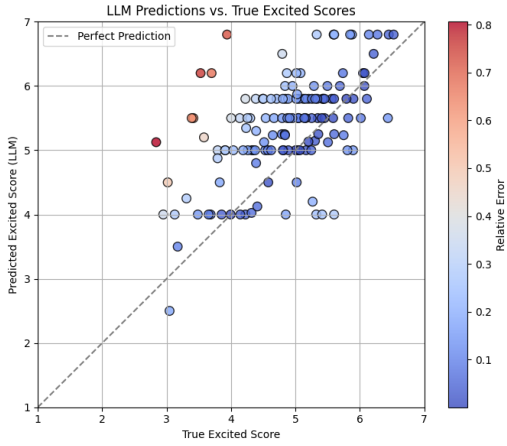


(d) 2-shot: LLM vs. True Excitement Score Residuals. MAE: 1.107

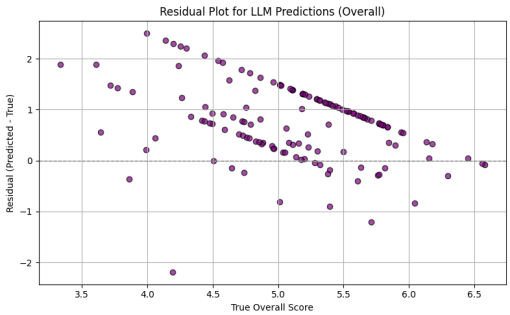
Figure 12. 2-shot LLM predictions versus true scores with residuals.



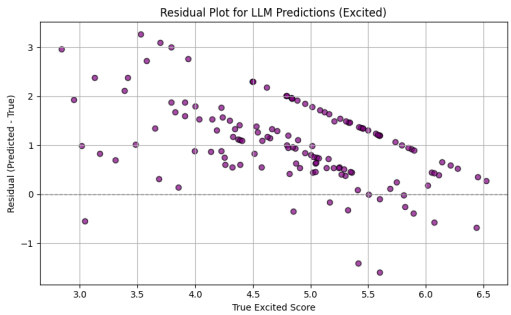
(a) 3-shot: LLM vs. True Overall Score. Pearson r:.509



(b) 3-shot: LLM vs. True Excitement Score. Pearson r:.559



(c) 3-shot: LLM vs. True Overall Score Residuals. MAE: 0.683



(d) 3-shot: LLM vs. True Excitement Score Residuals. MAE: .728

Figure 13. 3-shot LLM predictions versus true scores with residuals.

As you can see, the model did modestly well at predicting interview scores, especially in the three-shot case. As the plots suggest, it does a decent job at predicting most scores, though it seems to consistently err on the side of overestimating the score. It should be noted that the plots above represent the best results obtained after iterating through prompt engineering. It's worth mentioning that while the exact wording of our prompts did not seem to matter much, the examples we chose, especially in the three-shot case, did. In general, we saw best results when providing interview examples from across the spectrum: one poor, one mediocre, and one strong. While it did not outperform the trained models discussed above, it did outperform the all NN's who only had access to sentiment data. For their small training data and ease of use, LLMs appear to be a viable option for this kind of task.

8 Future Work

- It would be quite interesting to look at how overlapping identities (e.g., race and gender) influence the perception of speech features during interviews.
- We could also analyze how AI(if used) for assessing interviews can perpetuate or mitigate existing biases built from speech patterns
- Study how interviewers perceive and process speech using neuro-cognitive methods (e.g., fMRI, EEG) or eye-tracking.
- Experimenting with structured coaching on speech and non-verbal communication for candidates to see how far they can go when it comes to improving chances of selection, without altering identity markers like accent.

References

Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova. 2019. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL),* 4171–4186. Association for Computational Linguistics. https://aclanthology.org/N19-1423/.

Franz, M., and B. Schölkopf. 2006. *Mutual information.* http://www.scholarpedia.org/article/Mutual_information. Accessed April 2025, 12.

Hutto, C., and E. Gilbert. 2014. “VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text.” In *Proceedings of the Eighth International Conference on Weblogs and Social Media (ICWSM).* https://ojs.aaai.org/index.php/ICWSM/article/view/14550.

Laerd Statistics. 2023. *Spearman’s Rank-Order Correlation - A Guide with Examples.* Accessed April 2025. https://statistics.laerd.com/statistical-guides/spearmans-rank-order-correlation-statistical-guide.php.

OpenAI. 2024. *GPT-4o and GPT-4o-mini Technical Overview.* https://openai.com/index/gpt-4o. Accessed April 2025.