Automated Job Interview Analysis

Emre Uğur*, Elifsena Öz†, İpek Öztaş‡
*Bilkent University, Ankara, Turkey
emreu02@gmail.com
†Bilkent University, Ankara, Turkey
‡Bilkent University, Ankara, Turkey
*Main Author

***Abstract*—This paper presents our work for the online recruitment platform JobTalk, which can automatically assess the interviews. We propose an automated framework for analyzing non-verbal behaviors and hiring decisions using OTS (Over The Shelf) tools. Additionally, we describe a semi-automated method to analyze the content quality of the interviews using an LLM (Large Language Model). We trained our models with the MIT Interview Dataset, which consists of 138 video interviews, each scored on friendliness, recommended hiring, colleague, engagement, etc. We trained an SVR (Support Vector Regression) and selected the best-performing models. We present a multimodal framework for predicting these qualities by combining prosodic, lexical, and facial features. We used OTS tools to obtain these features, such as Whisper for transcription, python-fer for facial analysis, and Praat for prosodic analysis.**

***Index Terms*—Non-verbal behavior analysis, job interviews, automated framework, large language model, MIT Interview Dataset, support vector regression, multimodal framework, Whisper, python-fer, Praat**

I. Introduction

Recruitment is essential for company success since it impacts talent quality. Rising candidate numbers and diverse qualifications pose challenges in recruitment. Online methods streamline the process, yet evaluating data remains complex. HR may lack technical expertise, necessitating an unbiased approach for effective hiring. This paper proposes the research we conducted for the online recruitment platform JobTalk. JobTalk ensures a fair evaluation process using machine learning algorithms, eliminating human bias related to appearance, ethnicity, or background. Furthermore, JobTalk speeds up the recruitment process, requiring less action from the human resources department. By providing an automated report on the candidate, JobTalk improves the overall recruitment performance.

II. Related Work

Previous work has been done to characterize the speaking proficiency of a speaker through audio feature extraction and machine learning techniques. A. Preciado and R. Brena [9] introduce a multi-class classification approach to the speaking proficiency problem by taking the mel-cepstral coefficients of a five-second recording and training a classifier. S. Sabahi [11] describes a similar approach where low-level audio features are calculated using the open-source software Praat [12], and a regression model is trained on these features to estimate the CEFR level. These methods do not consider if the language is spoken at all. One can imitate a good speaker by mumbling, resembling that language, and get a good score. Therefore, there has been a growing interest in utilizing an automated speech recognition (ASR) system in speaking proficiency classification. K. Zechner et al. [1][2][3] describe an automated system for scoring IELTS test responses. Their initial work [1][2] computes several features from the output of their custom ASR, such as the speaking rate, the length of continuously spoken word chunks, and the ASR confidence score. In the domain of non-verbal behavior recognition, several automated systems are proposed. Ranganath et al. [13] analyze several conversational styles, such as awkwardness, assertiveness, flirtatiousness, and friendliness, with only prosodic and linguistic features. They based their analysis on the SpeedDate corpus, where dates rated each other based on these aspects. Kapoor et al. [14] analyze facial and postural features to evaluate a learner's interest in the subject. Nguyen et al. [15] propose an automated process to predict the hiring decision based on an interview by analyzing prosodic and facial features. I. Naim et al. [5] extend the work of Nguyen et al. by including lexical features and analyzing several behavioral traits (e.g., friendliness, excitement, engagement). We extend the work of Naim et al. by including prosodic features related to sentence-forming characteristics from the timestamped transcription. Additionally, we propose a semi-automated content analysis workflow using a large language model.

III. Proposed Workflow

To build an automated workflow for automatically analyzing and predicting scores for a job interview recording. We separated the problem into two parts: content analysis and performance analysis. The content analysis part is responsible for analyzing the interview transcript to verify whether the candidate matches the profile given by the company. The performance analysis part concerns how the interviewee delivered the interview, such as the emotion and engagement portrayed, the speaking ability and characteristics, the word choice to provide the content, and so on. Fig 1 Illustrates the high-level flow.

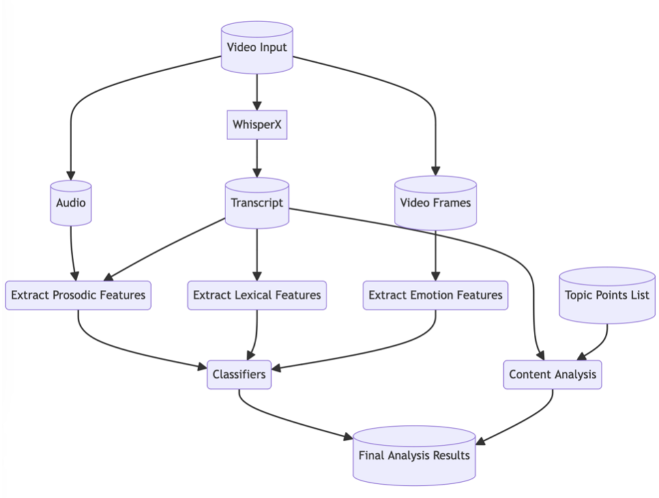

Fig. 1. Analysis Workflow

IV. Performance Analysis

For performance analysis, we used the MIT job interview dataset[6] to train multiple models that capture several aspects of the interview. N. Iftekhar et al. propose a multimodal machine learning pipeline to analyze the interviews.[5]

Features extracted for model training can be separated into three classes. The first class of features is concerned with how the speech is delivered. These prosodical features capture the rhyme, stress, intonation, speaking rate, and many other aspects of the speech. We used the software Pratt[12] to capture these prosodic features. Praat is a powerful open-source software that can analyze various low-level prosodical features related to frequency content, jitters and shimmers, and speech energy. We went beyond that and analyzed several other mid-level features, such as the silent pauses and filled pauses(uhhms, umms) in the speech, the rate of the speech, and the word chunk lengths. The next class of features is computed on the candidate's facial features. We extracted several emotions from each video frame. Finally, the last feature class is related to the word choice and distribution of words in the speech.

A. *Prosody Analysis*

We computed the following sets of low-level features using the Praat software.

TABLE I
Low-level Prosodic Features

Feature	Description
f1_mean	Mean frequency of F1
f2_mean	Mean frequency of F2
f3_mean	Mean frequency of F3
f4_mean	Mean frequency of F4
f1_med	Median frequency of F1
f2_med	Median frequency of F2
f3_med	Median frequency of F3
f4_med	Median frequency of F4
dur	Total duration
meanF0	Mean fundamental freq
stdevF0	Std dev of fundamental freq
hnr	Harmonics-to-noise ratio
loc_jitter	Local jitter measure
loc_abs_jitter	Absolute local jitter
rap_jitter	Relative avg perturbation jitter
ppq5_jitter	Five-point period perturbation jitter
ddp_jitter	Diff of periods jitter
loc_shimmer	Local shimmer measure
locdb_shimmer	Shimmer in dB
apq3_shimmer	Three-point ampl perturbation shimmer
apq5_shimmer	Five-point ampl perturbation shimmer
apq11_shimmer	Eleven-point ampl perturbation shimmer
dda_shimmer	Diff of amplitudes shimmer
mean_spec_energy	Mean spectral energy
intensity_mean	Mean intensity
intensity_min	Minimum intensity
intensity_max	Maximum intensity
intensity_range	Intensity range
intensity_sd	Std dev of intensity

We wanted to go beyond these low-level features since using filler words, pauses, and speech rate greatly captures a speaker's speaking ability and characteristics.

Speech Rater[1][2][3] utilizes a custom-made ASR to make word-level timestamped transcriptions on which they calculate the features. A critical aspect of their custom ASR is capturing important structural events such as filled pauses and disfluencies in the speech. This is a significant shortcoming of many off-the-shelf ASR tools since they are internally designed to filter out such sounds. Therefore, a VAD(Voice Activity Detection) is employed to detect filled pauses in the speech, as described in[4].

Our pipeline starts with a word-level timestamped transcription. We decided to use Whisper[7] since it is extremely powerful and open-source. However, Whisper can only provide segment-level timestamps. Segments are similar structures to sentences. To tackle the problem of word-level timestamps, we used a variation of whisper named whisper-x[8], which does forced alignment with the transcription and the audio file. Using whisper-x, we obtained a word-level timestamped transcription. From this transcription, we gathered the features in Table 2.

TABLE II
LEXICAL FEATURES

Feature	Description
uniq_words	Number of unique words
avg_chunk_len	Avg number of words per chunk
mean_dev_chunks	Avg deviation in words per chunk
artic_rate	Words per minute
dur_sil_per_word	Avg duration of silences per word
mean_sil_dur	Avg duration of all silences
mean_long_pauses	Avg duration of long pauses
freq_long_pauses	Freq of long pauses relative to total words
types_uttsegdur	Ratio of unique words to total duration
mean_filled_pauses	Avg length of filled pauses
freq_filled_pauses	Freq of filled pauses
asr_score	ASR analysis score

We defined a word chunk as a sequence of words with a gap between them under a fixed threshold duration. Identifying filled pauses was another major problem we had to tackle since Whisper treats those filled pauses as silences and filters them so that they are not in the final transcript. We utilized the method described by G. Zhu et al. [4]. The algorithm first identifies the regions where the ASR is not triggered. These regions are either silences or filled pauses without a spoken word. We then run a VAD over these identified regions to identify which parts contain a voice and which are silent. The algorithm described in [4] further passes the voiced regions to a classifier model to detect if the part is a filled pause like umm, hmm, or another type of voiced part such as laughter or breaths. We implemented the classification by thresholding the length of the voiced parts. Shorter regions tend to have short bursts of breaths, and more significant regions tend to be filled with pauses.

B. Lexical Analysis

In this class of features, we calculate a histogram of word classes from the transcript. We used the LIWC2007[16] dictionary and included the most critical word classes identified in[5].

TABLE III
WORD HISTOGRAM

Description	Examples
I	I, I'm, I've, I'll, I'd
We	we, we'll, we're, us, our
They	they, they're, they'll, them
PosEmotion	hope, improve, kind, love
NegEmotion	bad, fool, hate, lose
Anxiety	Nervous, panic, shy
Anger	Agitate, confront, disgust
Sadness	Fail, grief, hurt
Cognitive	Cause, know, learn, notice
Inhibition	Prevent, stop, prohibit
Perceptual	Observe, experience, watch
Relativity	First, huge, new
Work	Project, study, thesis
Swear	Informal and swear words
Articles	A, an, the
Verbs	Common verbs
Adverbs	Common adverbs
Prepositions	Common prepositions
Conjunctions	Common conjunctions
Negations	No, none, don't
Quantifiers	All, few, ton
Numbers	first, second, hundred

C. Emotional Analysis

To capture the emotional state of the interviewee, we calculated seven features from the video.

TABLE IV
EMOTION-RELATED FEATURES

Feature	Description
angry	Level of anger
disgust	Level of disgust
fear	Level of fear
happy	Level of happiness
neutral	Neutral state
sad	Level of sadness
surprise	Level of surprise

We used a Python package called fer[17] to obtain these features for each frame. For a frame, fer gives a probability distribution over the emotion classes. After processing the video, we averaged the values along the time axis. That is, we took the average of all frames. Paper[5] describes a similar method where they have a binary classifier that can identify between smiling and non-smiling faces.

V. CONTENT ANALYSIS

This section describes a flexible content analysis flow using a LLM. We use OpenAI's GPT-4 API. Our flow starts by obtaining a set of topic points expected to be covered in the interviewee's answer. In a traditional approach, these topic points would be keywords where a search in the transcript is performed for the keyword. Using an LLM gives us flexibility in two ways. First, we can use a sentence or question as a topic point in addition to single words. This could be like "Interviewee gave examples of their weaknesses." and "Interviewee mentioned how they can work to improve their weaknesses." Second, if a topic point is given as a keyword, we can check if the context somehow infers that keyword without explicitly containing it. For example, for the keyword "Relational Database," if the transcript mentions SQL somehow, we can identify that the transcript includes information related to Relational databases. We ask the LLM to give 0 points if the topic point is not contained in the answer, one if the point is partially or weakly given, and 2 if the topic point is entirely contained in the answer. We then asked the LLM to provide a summary of the topic points. The summary then includes if the topic point is contained, how it is contained, or how it is partially mentioned. We then calculate a content score out of 1. This score is calculated as

$$\text{Content score} = \frac{\sum \text{topic scores}}{\text{num of topic points} \times 2} \tag{1}$$

VI. Regression Model Training & Results

After computing all the features, we concatenate them to a single vector. We then normalize each feature to mean 0 stdev 1. In the MIT dataset [6], each interview is scored under the 18 categories seen in Table 5. We trained a different SVR(Support Vector Regression) for each scoring category. AUC indicates how well the model can sort the inputs for a regression model. In the dataset, we had the following scores for interviews: Overall, RecommendHiring, Colleague, Engaged, Excited, EyeContact, Smiled, SpeakingRate, NoFillers, Friendly, Paused, EngagingTone, StructuredAnswers, Calm, NotStressed, Focused, Authentic, NotAwkward. We selected the models with high correlation and included them in our results and analysis pipeline.

We used twenty percent of the data to test our models. To remove any bias that can result in selecting a specific subset of the data for testing. We did 1000 different test splits and had the mean of the correlation and AUC scores.

TABLE V
Trained Model Statistics

Model	Corr	AUC
RecHire	0.6122	0.7727
Colleague	0.5484	0.7662
Engaged	0.6468	0.7894
Excited	0.7463	0.8716
Smiled	0.6950	0.8321
Friendly	0.7076	0.8144
Paused	0.5266	0.7246
EngTone	0.7355	0.8538
StrucAns	0.6191	0.7896

I. Naim et al. [5] report $r > 0.62$ with RecHire and $r > 0.7$ with engagement, excitement, and friendliness using their framework. Our findings using OTS tools match the results they present. The relatively high structured answers (StrucAns) correlation was also surprising as there aren't any direct features supporting that quality. We suspect the features related to word chunks in Table 2 provide a clue on structured answers as those features are related to speaking proficiency[1][2][3].

References

[1] K. Zechner and I. Bejar, "Towards Automatic Scoring of Non-Native Spontaneous Speech," in *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, R. C. Moore, J. Bilmes, J. Chu-Carroll, and M. Sanderson, Eds., New York City, USA, Jun. 2006, pp. 216–223. [Online]. Available: https://aclanthology.org/N06-1028

[2] K. Zechner, D. Higgins, X. Xi, and D. M. Williamson, "Automatic scoring of non-native spontaneous speech in tests of spoken English," *Speech Communication*, vol. 51, no. 10, pp. 883–895, 2009. [Online]. Available: https://doi.org/10.1016/j.specom.2009.04.009

[3] L. Chen, K. Zechner, S.-Y. Yoon, K. Evanini, X. Wang, A. Loukina, J. Tao, L. Davis, C. M. Lee, M. Ma, R. Mundkowsky, C. Lu, C. W. Leong, and B. Gyawali, "Automated Scoring of Nonnative Speech Using the SpeechRaterSM v. 5.0 Engine," *ETS Research Report Series*, 2018. [Online]. Available: https://doi.org/10.1002/ets2.12198

[4] G. Zhu, J.-P. Caceres, and J. Salamon, "Filler Word Detection and Classification: A Dataset and Benchmark," *arXiv preprint arXiv:2203.15135*, 2022.

[5] I. Naim, M. Tanveer, D. Gildea, and E. Hoque, "Automated Analysis and Prediction of Job Interview Performance," *IEEE Transactions on Affective Computing*, vol. PP, 2015. [Online]. Available: 10.1109/TAFFC.2016.2614299

[6] R. Abdul Baten *et al.*, "Automated Prediction of Job Interview Performances," *ROC HCI*. [Online]. Available: https://roc-hci.com/past-projects/automated-prediction-of-job-interview-performances/

[7] OpenAI, "Whisper," *GitHub repository*, GitHub, 2023. [Online]. Available: https://github.com/openai/whisper. [Accessed: 08-May-2024].

[8] M. Bain, "WhisperX: Automatic Speech Recognition with Word-level Timestamps (& Diarization)," *GitHub repository*, GitHub, 2023. [Online]. Available: https://github.com/m-bain/whisperX. [Accessed: 08-May-2024].

[9] P. H. and T. S., "AvaLinguo Dataset," *Kaggle*, 2023. [Online]. Available: https://www.kaggle.com/datasets/petalsonwind/avalinguo-dataset. [Accessed: 30-May-2024].

[10] P. Mishra, M. J. and P. D., "Speaker Fluency Level Classification using Deep Learning," *Papers with Code*, 2021. [Online]. Available: https://paperswithcode.com/paper/speaker-fluency-level-classification-using. [Accessed: 30-May-2024].

[11] S. K. Samiei, "Speechat," 2024. [Online]. Available: https://shahabks.github.io/Speechat/. [Accessed: 30-May-2024].

[12] P. Boersma and D. Weenink, "Praat: doing phonetics by computer," Version 6.3, University of Amsterdam, 2024. [Online]. Available: https://www.fon.hum.uva.nl/praat/. [Accessed: 30-May-2024].

[13] R. Ranganath, D. Jurafsky, and D. A. McFarland, "Detecting friendly, flirtatious, awkward, and assertive speech in speed-dates," *Comput. Speech Language*, vol. 27, no. 1, pp. 89–115, 2013.

[14] A. Kapoor and R. W. Picard, "Multimodal affect recognition in learning environments," in *Proc. 13th Annu. ACM Int. Conf. Multi- media*, 2005, pp. 677–682.

[15] L. Nguyen, D. Frauendorfer, M. Mast, and D. Gatica-Perez, "Hire me: Computational inference of hirability in employment interviews based on nonverbal behavior," *IEEE Trans. Multimedia*, vol. 16, no. 4, pp. 1018–1031, Jun. 2014.

[16] J. W. Pennebaker, R. L. Booth, and M. E. Francis, "The development and psychometric properties of LIWC2007," *ResearchGate*, 2007. [Online]. Available: https://www.researchgate.net/publication/228650445_The_Development_and_Psychometric_Properties_of_LIWC2007

[17] "FER: Facial Expression Recognition using PyTorch," PyPI. [Online]. Available: https://pypi.org/project/fer-pytorch/. [Accessed: May 30, 2024].