Speaking State Decoder with Transition Detection for Next Speaker Prediction

Shao-Hao Lu[1], *Yun-Shao Lin*[2], *Chi-Chun Lee* [3]

[1]Department of Electrical Engineering, National Tsing Hua University, Taiwan

{rosyhoward1223, astanley18074}@gmail.com, cclee@ee.nthu.edu.tw

Abstract

Next speaker prediction and turn change prediction are two important tasks in group interaction and human-agent interaction. In order to carry out a fluent conversation, we need to identify who is currently speaking, who is the next speaker and when the next speaker starts to speak. These questions are computationally designed as the task of *next speaker prediction*. Behaviors such as gaze direction, speaking prosody or gestures have been modeled to perform this task. In this work, we propose a decoder-based speaking state decoder (SSD) for next speaker prediction, which jointly considers current behavior features, past history of talking and speaking state transition detection model. Our decoder approach achieves next speaker prediction with UAR of 78.11%, which is 3.41% improvement over the champion model in MultiMediate challenge 2021.

Index Terms: next speaker prediction, transition detection, attention mechanism, decoder

1. Introduction

Face-to-face interaction is one of the most common and basic form of communication in daily life. In order to have a smooth and fluent participation in conversation, human must understand when the speakers are going to keep or yield their turns. This ability is known as *next speaker prediction*. Furthermore, smooth communication similar to face-to-face interaction is also desired in human-agent communication and remote human-to-human communication [1]. This has led to many research investigating ways to analyze group interaction or multiparty meetings. For example, Levinson et al. have investigated in the relationship between response time and psycho-linguistic norm [2]. Duncan et al. have analyzed the relationship between behavior cues from speakers and turn-taking tendency [3]. Lin et al. have investigated small group interaction behavior patterns with personality traits [4]. These works help us understand more about conversational interaction flow also enable computational research in making machines to become more user-friendly with human-like reliable next speaker prediction.

There are many computational frameworks developed over the years for turn-taking prediction and next speaker prediction. For example, Ishii et al. focus on extracting features about head movement, such as head movement amplitude (cm/sec) and head movement amount (cm) to identify the relationship between head gesture and next speaker [5]. Yang et al. propose a novel multimodal fusion algorithm with a gate parameter to learn the weight among different modalities [6]. Liang et al. focus on extracting linguistic information using multiple transformer blocks to model global verbal features to predict turn-taking [7]. In this work, we propose a speaking state decoder (SSD) and evaluate our model on the data used in the MultiMediate challenge 2021 [8]. The task of the MultiMediate challenge is to predict whether the target member will talk in 1 second by modeling the provided video clip as input. The champion model [9] in MultiMediate 2021 utilizes group focus and audio-video synchronization to perform the task. By working on this challenge corpus, we can have a fair comparison with others in the field.

Almost all of the participating methods in this challenge do not consider speaker's past speaking state history. That is, those methods do not explicitly modeling long range sequence of a member's speaking state tendency that naturally happens in conversation [10] [11]. A straightforward method would treat it as a sequence decoding task by using a speaking state decoder (similar to language model used in automatic speech recognition or speech emotion recognition [12]). However, direct state-counting maximum likelihood-based approaches often cause over-smooth prediction since those decoders tend to predict on majority class [13]. In this work, to eliminate such an issue, we propose to discriminatively learn a transition model that is dynamically integrated to the speaking state decoder.

In this work, we propose a speaking state decoder (SSD) with feature-based network for transition detection. SSD makes next speaker prediction after considering a participant's past speaking history. This decoder relies on three probabilities: base prediction, transition detection, and state assignment. We extract delicate frame-wise behavior features of gaze patterns and verbal behaviors to produce reliable base prediction. Furthermore, we design a transition model with two modified self-attention mechanisms to predict transition probability dynamically between previous and current sample. One focuses on *time-awareness*, this intends to make the transition model better aware on those behaviors that is either far from or close to the current time. Another one focuses on *behavior divergence*, this intends to make the model emphasize those regions where there exists a substantial divergence in the target speaker's behaviors. Specifically, our contribution in this work is listed below:

- Our proposed speaking state decoder (SSD) with transition model exceed champion model in MultiMediate 2021 challenge by 3.41%.
- Our transition model with time-awareness and behavior divergence attention mechanisms obtain relative model performance improvements by 22.33% in UAR when speaking state transition happens.

2. Methodology

2.1. Data corpus

In this paper, we conduct experiments on MultiMediate'21 data corpus. MultiMediate challenge [8] use MPIIGroupInteraction

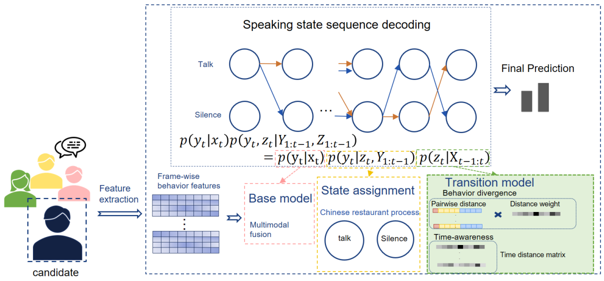

Figure 1: *Structure of speaking state decoder (SSD).*

corpus [14] with additional next speaker prediction annotation since the original MPIIGroupInteraction corpus is built for rapport detection that do not include relevant annotations for next speaker prediction. Each group interaction lasts 20 minutes, and the study manager chooses a discussion topic that is maximally controversial among the group participants (creating a highly-involved conversation). Each interaction recording is annotated with a strict protocol. It not only includes labels about the regular speaking utterances, the annotators are instructed to label back-channel (e.g., 'hmm' or 'right') and short affirmative or dissenting statement (e.g., 'yes' or 'no') as speaking. This carefully crafted challenge corpus has become the benchmark corpus for evaluating models of next speaker prediction.

When determining the next speaker task label, the corpus makes use of this speaking annotation. It finds the frames before a speaker state change occurs (i.e., silence to talk, talk to silence) and subtracts a random offset in the range [0, 1000] milliseconds as an anchor point to determine the last frame to be used as input. After the last frame of the input is determined, the next speaker prediction point is set at 1 second after that last frame. With this setup, the organizers ensure that the next speaker does not always start exactly 1 second after. Furthermore, to balance the dataset, they generate an equal number of samples with random anchor points where there is no speaker change occurs. In total, there is 30,184 samples in training set (16 group recordings) with 22,132 positive samples (talking) and 8,052 negative samples (silence); 16,144 samples in testing set (six group recordings) with 3,994 positive samples and 12,150 silence samples.

2.2. Task definition

With an observation window of 10 seconds for a speaker starting at time t, a model has to predict whether he/she speaks at time $t+1s$, i.e., one second after the end of an observation window. The next speaker prediction is a binary classification task (speaking versus not-speaking) for each participant's sample.

In this work, we propose to treat this as a state decoding task. That is, given a sample of video sequence from a target member $V = \{v_1, ..., v_t\}$, at time $t \in [1, T]$, our goal is to recognize y_t, i.e., whether the member will talk 1 second after v_t, depends on the sequence of previous $t-1$ predicted speaking state. This probabilistic sequence decoding can be formulated as equation 1,

$$p(Y, Z) = p(y_1|x_1) \prod_{t=2}^{T} p(y_t|x_t)p(y_t, z_t|Y_{1:t-1}, X_{t-1:t}) \quad (1)$$

where the probability of recognizing the t^{th} speaking state y_t can be factorized into equation 2.

$$\begin{aligned} p(y_t, x_t)p(y_t, z_t|Y_{1:t-1}, X_{t-1:t}) \\ = p(y_t|x_t)p(y_t|z_t, Y_{1:t-1})p(z_t|X_{t-1:t}) \end{aligned} \quad (2)$$

In equation 2, $p(y_t|x_t)$ indicates the base prediction, $p(y_t|z_t, Y_{1:t-1})$ indicates the probabilities of next speaker based on previous state history, and $p(z_t|X_{t-1:t})$ indicates probabilities of the transition (change) in speaking state. The overall model structure is shown in Figure 1.

2.3. Feature extraction network

Feature extraction network aims to extract behavior features from the video and audio input clips for our proposed SSD. This consists of two networks, TalkNet and GazeNet. TalkNet [15] is a state-of-the-art active speaker detection method which aims to learn long-term audio-video relationships for robust active speaker detection. With TalkNet, we are able to extract frame-wise active speaker confidence that captures whether the speaker in the video is talking or not. On the other hand, GazeNet is built to classify candidate's gaze directions with OpenFace [16] extracted features for gaze patterns detection. GazeNet is built with four fully-connected layer and output with four categories, where $Y_{gaze} \in$ {right, left, middle, nothing}.

2.4. Speaking state decoder (SSD)

2.4.1. Speaker state base prediction module

We design a network named BASE for next speaker prediction, which is built with self-attention module and fully-connected layers. We use self-attention to learn the frame-to-frame relationships from the extracted behavior features and concatenate these audio and video features. Then, we use a DNN network to obtain the next speaker prediction using the concatenated self-attended embeddings.

2.4.2. Speaking state assignment

We implement Chinese restaurant process [17] for next speaking state assignment. It is a clustering method to model the distribution of next speaking state. In this task, states can only jump between two states (speaking, silence) and the occurred state is assigned to corresponding speaking block. The probability assigned to current state y_t is determined as equation 3,

$$p(y_t = l|zt = 1, Y_{1:t-1}) \propto \begin{cases} N_{l,t-1}, & l \in Y_{1:t-1} \\ \alpha, & l \in new\ state \end{cases} \quad (3)$$

where $\alpha \in \mathbb{R}$, $N_{l,t-1}$ denotes the size of the speaking block for state l up to time $t-1$ and l indicates the states, that is, $talk, silence$. An speaking block is defined as a sequence that has the longest common-state consecutively by an individual speaker. The probability of switching back to previous appeared state is proportional to the number of continuous samples the member has spoken. When it comes to switching to a new state, the probability is proportion to a constant α, which is set to 1 in general. When $z_t = 0$, the speaking state remains unchanged as his/her previous speaking state. The joint distribution of Y given Z, α is equation 4,

$$p(Y|Z, \alpha) = \frac{\alpha^{K_T - 1} \prod_{l \in E} \Gamma(N_{l,T})}{\prod_{t=2}^{T} (\sum_{l \in E} N_{l,t-1} + \alpha)^{[z_t = 1]}} \quad (4)$$

where K_T indicates the number of unique states up to time t, E indicates the maximum states amounts, in this task $E = 2$.

2.4.3. *Transition detection module*

We design a transition detection model to generate dynamic state-transition probabilities between two consecutive samples. The module consists of two self-attention mechanisms: time-aware and behavior divergence.

$$P_{Transition} = P(z_t|X_{t-1:t}) \tag{5}$$

Time-aware self-attention

For time-aware self-attention, we aim to focus on the relationship between the time (temporal duration) difference and speaking state. First, we calculate the normalized frame-wise time difference between previous and current sample [18, 19]. Then, we multiply it with a trainable parameter a, where a learns the importance along temporal duration when the time difference increases/decreases. If a is positive after training, it indicates that the behaviors in recent time are more important. On the other hand, behaviors in distal time are more important if a is negative. The equation is shown in 6,

$$\begin{aligned} t_{i,j} &= P_i - C_j, \\ t'_{i,j} &= \tfrac{1}{\log(e+t_{i,j})}, \\ \alpha^t_{i,j} &= \text{Sigmoid}(a * t'_{i,j} + b) \end{aligned} \tag{6}$$

where P_i indicates ith time of previous sample, C_j indicates jth time of current sample, $e \approx 2.718$, $\alpha^t_{i,j}$ shows the time importance. The normalization aims to represent a global time relationship and bound the value into the range of [0,1]. When the time difference is larger, the values are smaller.

With equation 7, we calculate a time importance matrix $\alpha^T \in \mathbb{R}^{n*n}$ shows the time difference importance between previous and current sample. The matrix is shown in 7.

$$\alpha^T = \begin{bmatrix} \alpha^t_{1,1} & \alpha^t_{1,2} & \cdots & \alpha^t_{1,n} \\ \alpha^t_{2,1} & \alpha^t_{2,2} & . & \alpha^t_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha^t_{n,1} & \alpha^t_{n,2} & \cdots & \alpha^t_{n,n}, \end{bmatrix} \tag{7}$$

With the time importance α^T, we can combine it into self-attention algorithm [20], i.e., re-weighting the original self-attention with time importance attention, aiming to obtain V_{T_aware} with addition time importance information. $Q \in \mathbb{R}^{n \times d}$ obtained from current sample's embedding is transformed into $Q_t = \alpha^T \cdot Q$ which integrates the consideration of time importance. Equation 8 shows the combination of origin self-attention and time importance self-attention,

$$V_{T\_aware} = tanh(\frac{QQ^\top}{\sqrt{d_k}} + \frac{Q_tQ_t^\top}{\sqrt{d_k}})Q \tag{8}$$

where V_{T_aware} represents time-aware self-attention embedding, and d_k indicates the dimension of features.

Behavior divergence self-attention

For behavior divergence attention, we aim to indicate the importance of behavior differences. Similar to time-awareness mentioned above, we obtain the behavior distance matrix $D \in \mathbb{R}^{n \times n}$, by calculating $L1$norm between previous and current sample and multiply a weight $W \in \mathbb{R}^{f_{dim}}$,

$$\begin{aligned} D &= \textstyle\sum_{f=1}^{f_{dim}} |P_i^f - C_j^f| * w_f \\ \alpha^D &= Sigmoid(D) \end{aligned} \tag{9}$$

where f_{dim} indicates the dimension of behavior features, P_i^f indicates the fth behavior feature of ith data in previous sample, similar for C_i^f.

Having a behavior divergence importance α_d, we apply the same process as time-awareness to combine the importance into self-attention process. The equation is shown in 10, where V_d indicates behavior-divergence self-attention embedding.

$$V_{B\_div} = tanh(\frac{QQ^\top}{\sqrt{d_k}} + \frac{Q_dQ_d^\top}{\sqrt{d_k}})Q \tag{10}$$

After obtaining the embedding from time-awareness and behavior-divergence, we concatenate V_{T_aware}, V_{B_div}, V_{cp_cat} and input into fully-connected layers to predict transition probabilities, where V_{cp_cat} indicates the attended embedding obtained from concatenation of previous and current behavior features. With this proposed transition model, we are able to model the speaker state transition dynamically with local context that can be integrated to the speaker state decoder.

2.4.4. *Decoding*

Given a testing video sample sequence V, we treat U as a sequence and select state that maximizes posterior probability from the next speaker prediction module, transition shift modeling, and speaking assignment process.

$$\hat{Y} = \text{argmax } \log p(X, Y) \tag{11}$$

Instead of finding the best path exhaustively, we pick the best path using beam search where beam size $n = 5$.

$$\hat{y}_t, \hat{z}_t = \text{argmax } \log p(y_t, x_t) + \log p(y_t|z_t, \hat{Y}_{1:t-1}) + \log p(z_t) \tag{12}$$

3. Experiments and Results

3.1. Experiments setup

Our proposed models are implemented using the Pytorch library (version 1.10.2) with the Adam optimizer. The learning rate is set as $5e^{-4}$. Furthermore, we train our models with RTX 2080, and 4 hours is needed for training BASE. We use the training set for training with CV fold=5 and choose the model with lowest validation loss for evaluation in testing set. We evaluate the performance with unweighted average recall (UAR), accuracy, f1 score and precision score.

3.1.1. *Comparison of models*

We compare model performances from two perspectives: overall performance and performance under transition condition.

- Champion models [9]. MultiMediate 2021 champion models where the results are obtained from paper directly.
- Gate fusion [6]. The state-of-the-art multi-modal network for next speaker prediction. It uses trainable gate parameter to control the fusion weight between different modalities.
- BASE. Our next speaker base prediction model.
- SSD. Our proposed speaking state decoder approach.

3.1.2. *Comparison of transition detection model*

Comparing the performance of our designed transition model with other baseline.

- Multitask. BASE architecture training with next speaker prediction and transition detection.

	Method	Transition Probability	UAR (%)	Acc (%)	F1 score (%)	Precision (%)
Model architecture	Champion (TCN)[9]	-	74.70	-	-	-
	Champion (BLSTM)[9]	-	72.10	-	-	-
	Champion (syncNet)[9]	-	71.50	-	-	-
	Gate fusion [6]	-	75.00	71.75	68.64	68.91
	BASE	-	75.15	72.66	69.31	69.20
Sequence decoding	SSD+BASE	Fix transition bias	75.76	74.63	70.85	70.12
	SSD+BASE	Transition model result	**78.11**	77.92	73.94	72.67

Table 1: *Model performance comparison on next speaker prediction task.*

Method	model	UAR	Acc	F1 score	Precision
Multitask	base model structure	62.78	50.68	46.73	56.57
Pretrained embedding	svm	64.03	47.83	45.10	57.54
Base model concatenate	base model structure	64.18	53.47	48.81	57.18
Proposed transition model	nn	**67.58**	**61.66**	**54.63**	**58.86**

Table 2: *Performance comparison of transition detection task.*

		transition happened	same state continues	
Model	Transition bias	UAR (%)	UAR (%)	overall UAR
Gate fusion	-	40.20	81.63	75.00
BASE	-	40.92	81.78	75.15
SSD+BASE	Fix	40.88	82.68	75.76
	Proposed transition model	**63.21**	82.64	**78.11**

Table 3: *Model performance comparison of next speaker prediction under transition condition.*

- Pretrained embedding. Using the previous and current sample's pre-final layer from BASE and input it into SVM for transition detection.
- Base model concatenate. Using two BASE structures for previous and current input samples, concatenate last layer embedding into fully-connected layers for transition detection.

3.2. Results

Table 1 shows the performance in next speaker prediction task. We observe that our BASE have reached UAR 75.15%. Comparing between different model architectures, we outperform champion model and the state-of-the-art gate fusion by 0.75% and 0.45% in UAR respectively. By combining it with speaking state decoding (SSD), we are able to improve our performance to 78.11%, which exceeds the champion model by 3.41% and surpasses the latest gate fusion by 3.11% in UAR. This demonstrates our BASE has competitive performances. Furthermore, modeling time sequence speaker's speaking state is necessary for improving next speaker prediction.

3.3. Analysis of transition models

Table 2 shows the comparison of transition models. Our proposed model is able to detect transition with UAR 67.58%, which surpasses other baseline by 4.8%. This shows the importance of crafting designated algorithms for detecting transition, i.e., simply using the same architecture can not reflect subtle behavior difference between previous and current samples thus performing worse on transition detection.

Furthermore, we investigate into our time-awareness and behavior divergence network. In time-awareness, the trainable a which is mentioned in 2.4.3 results in a value of -0.03. This result shows that the behaviors in the distal time are more informative to determine whether the speaking state changes. On the other hand, Figure 2 shows the feature weight in behavior divergence. We observe that the more significant changes in talkative or group focus, the more likely a change in the speaking state of the target member would occur. That is, if previous and current samples show different behavior patterns, a speaking state change is likely to occur soon. When it comes to negative values (candidate's gaze behavior), it goes otherwise.

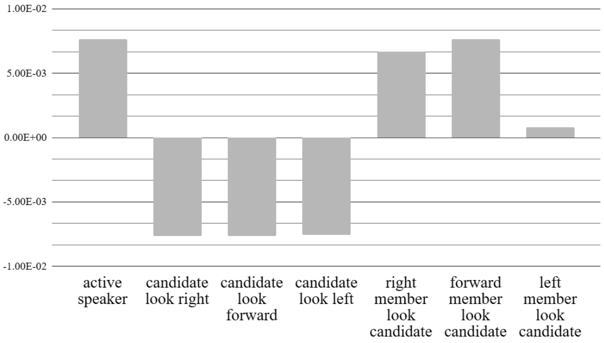

Figure 2: *Behavior divergence weight*

3.4. Model performance under transitions

Table 3 shows the performance in next speaker prediction task under transition condition. We find out that without the use of transition models. The performance drops severely when facing speaker state changes, i.e., -41.43% in gate fusion and -40.86% in BASE. Furthermore, the fix transition bias indicates using maximum likelihood to estimate transition probability for speaking states decoding. SSD with fix transition bias performs poorly when transitions happen. When the transition probability is static, it biases the decoder to predict majority labels and end up over-smooth the prediction. Our proposed SSD is able to maintain performance (63.21% UAR) under the transition condition by integrating dynamic transition probability into the decoding process.

4. Conclusions

In this work, we propose a speaker state decoder with transition detection using two novel attention mechanisms to perform speaker's speaking state sequence modeling for next speaker prediction. Our analysis shows that our efforts on delicate transition detection is able to handle the often poorly-detected speaking state transition. However, a limitation in this work is that our BASE can make consecutive false predictions which transition model is not able to correct through re-scoring. In our future work, we hope to decode group speaking states sequence jointly together instead of individual speaking state sequence inference done in this work; this enables the decoder to leverage every speaker's talking turn tendency and group dynamics as a whole (including components of each member's turn taking tendency, each member's interruption tendency or even member's tendency to start a new discussion). These directions is likely to further improve our SSD for next speaker prediction.

5. References

[1] T. Itoh, N. Kitaoka, and R. Nishimura, "Subjective experiments on influence of response timing in spoken dialogues," in *Tenth Annual Conference of the International Speech Communication Association*, 2009.

[2] S. C. Levinson and F. Torreira, "Timing in turn-taking and its implications for processing models of language," *Frontiers in Psychology*, vol. 6, 2015. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fpsyg.2015.00731

[3] S. Duncan, "Some signals and rules for taking speaking turns in conversations." *Journal of personality and social psychology*, vol. 23, no. 2, p. 283, 1972.

[4] Y.-S. Lin and C.-C. Lee, "Using interlocutor-modulated attention blstm to predict personality traits in small group interaction," in *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, 2018, pp. 163–169.

[5] R. Ishii, S. Kumano, and K. Otsuka, "Predicting next speaker based on head movement in multi-party meetings," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 2319–2323.

[6] J. Yang, P. Wang, Y. Zhu, M. Feng, M. Chen, and X. He, "Gated multimodal fusion with contrastive learning for turn-taking prediction in human-robot dialogue," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7747–7751.

[7] Y. Liang and Q. Zhou, "Detect turn-takings in subtitle streams with semantic recall transformer encoder," in *2020 International Conference on Asian Language Processing (IALP)*. IEEE, 2020, pp. 1–6.

[8] P. Müller, D. Schiller, D. Thomas, G. Zhang, M. Dietz, P. Gebhard, E. André, and A. Bulling, "Multimediate: Multi-modal group behaviour analysis for artificial mediation," in *Proc. ACM Multimedia (MM)*, 2021, pp. 4878–4882.

[9] C. Birmingham, K. Stefanov, and M. J. Mataric, "Group-level focus of visual attention for improved next speaker prediction," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 4838–4842.

[10] U. Malik, J. Saunier, K. Funakoshi, and A. Pauchet, "Who speaks next? turn change and next speaker prediction in multimodal multiparty interaction," in *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*, 2020, pp. 349–354.

[11] K. Jokinen, K. Harada, M. Nishida, and S. Yamamoto, "Turn-alignment using eye-gaze and speech in conversational interaction," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[12] C.-Y. Chen, Y.-S. Lin, and C.-C. Lee, "Emotion-shift aware crf for decoding emotion sequence in conversation," *Proc. Interspeech 2022*, pp. 1148–1152, 2022.

[13] Z. Zhou, R. Zhong, C. Yang, Y. Wang, X. Yang, and W. Shen, "A k-variate time series is worth k words: Evolution of the vanilla transformer architecture for long-term multivariate time series forecasting," *arXiv preprint arXiv:2212.02789*, 2022.

[14] P. Müller, M. X. Huang, and A. Bulling, "Detecting low rapport during natural interactions in small groups from non-verbal behaviour," in *23rd International Conference on Intelligent User Interfaces*, 2018, pp. 153–164.

[15] R. Tao, Z. Pan, R. K. Das, X. Qian, M. Z. Shou, and H. Li, "Is someone speaking? exploring long-term temporal features for audio-visual active speaker detection," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 3927–3935.

[16] T. Baltrušaitis, P. Robinson, and L.-P. Morency, "Openface: an open source facial behavior analysis toolkit," in *2016 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2016, pp. 1–10.

[17] D. M. Blei and P. I. Frazier, "Distance dependent chinese restaurant processes." *Journal of Machine Learning Research*, vol. 12, no. 8, 2011.

[18] T. Pham, T. Tran, D. Phung, and S. Venkatesh, "Deepcare: A deep dynamic memory model for predictive medicine," in *Pacific-Asia conference on knowledge discovery and data mining*. Springer, 2016, pp. 30–41.

[19] I. M. Baytas, C. Xiao, X. Zhang, F. Wang, A. K. Jain, and J. Zhou, "Patient subtyping via time-aware lstm networks," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2017, pp. 65–74.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.