INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Metadata-Aware End-to-End Keyword Spotting

Hongyi Liu, Apurva Abhyankar*, Yuriy Mishchenko, Thibaud Sénéchal, Gengshen Fu, Brian Kulis, Noah Stein, Anish Shah, Shiv Naga Prasad Vitaladevuni*

Amazon Alexa

{liuhn, apurvaa, yuriym, thibauds, gengshef, kulibria}@amazon.com,
{steinns, anishsh, shivnaga}@amazon.com

Abstract

As a crucial part of Alexa products, our on-device keyword spotting system detects the wakeword in conversation and initiates subsequent user-device interactions. Convolutional neural networks (CNNs) have been widely used to model the relationship between time and frequency in the audio spectrum. However, it is not obvious how to appropriately leverage the rich descriptive information from device state metadata (such as player state, device type, volume, etc) in a CNN architecture. In this paper, we propose to use metadata information as an additional input feature to improve the performance of a single CNN keyword-spotting model under different conditions. We design a new network architecture for metadata-aware end-to-end keyword spotting which learns to convert the categorical metadata to a fixed length embedding, and then uses the embedding to: 1) modulate convolutional feature maps via conditional batch normalization, and 2) contribute to the fully connected layer via feature concatenation. The experiment shows that the proposed architecture is able to learn the meta-specific characteristics from combined datasets, and the best candidate achieves an average relative false reject rate (FRR) improvement of 14.63% at the same false accept rate (FAR) compared with CNN that does not use device state metadata.

Index Terms: speech recognition, keyword spotting, metadata, convolutional neural network, feature embedding

1. Introduction

Keyword spotting refers to the task of locating a small vocabulary of words embedded in arbitrary conversation [1], and it has been widely adopted in various speech recognition applications. With the rapid expansion of Amazon Alexa products across the world, building high performance keyword spotting (wakeword detection) system has become one of the key factors behind the successful product launches and satisfied customers.

Conventional keyword spotting systems use the Gaussian-mixture-model based hidden Markov models (GMM-HMM)[2, 3, 4], where GMM is used for modeling observed acoustic features and HMM represents the dynamic sequential characteristics of keyword and background audio. With the prevalence of deep neural networks since the year of 2012 [5, 6], DNN starts to gradually replace GMM because of its strong representation learning power to deal with highly nonlinear manifolds in the feature space of acoustic data, and DNN-HMM hybrid system has become a common approach for large vocabulary continuous speech recognition (LVCSR) task.

Recent years have witnessed increasing popularity in adopting convolutional neural networks in automatic speech recognition tasks[7, 8, 9]. CNN [10] applies 3D convolutional kernels on audio spectrogram and models the local correlations in time and frequency domain. To model the sequential speech data, RNN, LSTM and GRU have also been proposed to exploit historical information and predict the labels of current frames during audio decoding [11, 12, 13].

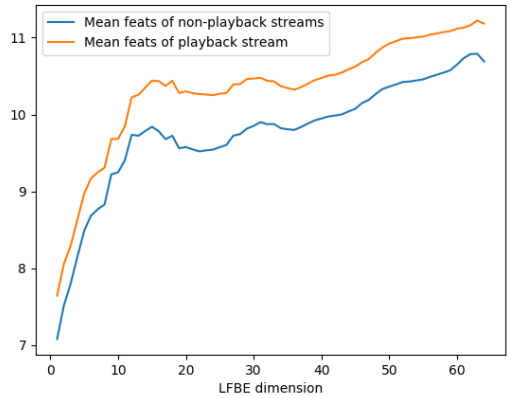


Figure 1: *The sample mean of LFBE features computed from 50k wakeword utterances with playback and non-playback metadata, separately.*

Multimodal learning involves relating information from multiple sources [14]. While single-modality models are able to generalize on various characteristic of training data, there are cases where multi-modality models are more favorable due to their adaptability to additional input feautres such as device states or settings. For example, we might want to use single model to process data with different device settings [15]. Figure 1 displays the discrepancy of LFBE features between utterances collected under normal condition and playback condition (where player is playing music or alarm clock) that contain wakeword. There exists noticeable mean shift for streams tagged as playback in LFBE feature space. This observation motivates us to explore the possibility of using metadata information to improve the performance of wakeword models.

We explore an approach to incorporate metadata based on batch normalization and its conditional extension. [16] proposes batch normalization (BN) which normalizes feature maps for each input mini-batch to reduce the internal covariant shift and accelerate DNN training. Conditional batch normalization (CBN) [17, 18] is later introduced to modulate the offset and scale of batch norm layer according to different types of inputs. The advantage of CBN is that it modulates the feature maps independently given different inputs, therefore improves network's generalization ability on heterogeneous data. Another commonly

* denotes equal contribution.

http://dx.doi.org/10.21437/Interspeech.2020-1262

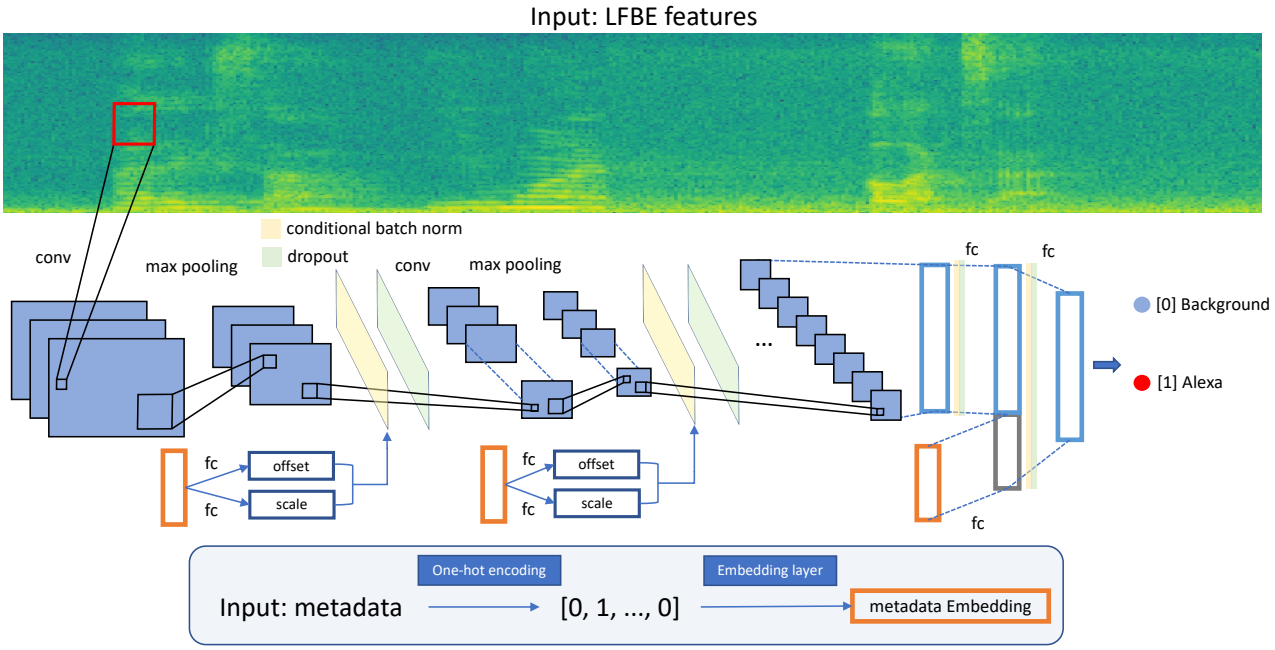


Figure 2: *Proposed CNN architecture. Playback status is used as additional input to generate metadata embedding vector, which can predict offset and scale for conditional batch normalization layer, as well as contribute to the audio feature via feature concatenation. Conv and fc denotes convolutional layer and fully connected layer, respectively.*

used approach to fuse multi-modal information is feature concatenation. The features of individual modality are concatenated together as a joint representation in mutlimodal space [19].

In this paper, we explore the benefits of utilizing metadata information as additional features for end-to-end keyword spotting task. We propose a new CNN architecture which learns to convert input metadata name to a fixed length embedding, and then uses the embedding to: 1) modulate convolutional feature maps via conditional batch normalization, and 2) contribute to fully connected layer via feature concatenation.

The rest of paper is organized as follows: In Section 2, we introduce the proposed network architecture. In Section 3, we describe our experiment settings and discuss the results. Section 4 concludes our paper.

2. Metadata-Aware Keyword Spotting

In this section we describe the proposed metadata aware end-to-end keyword spotting system.

2.1. Metadata Feature Embedding

In order to leverage metadata information to modulate the network, the metadata value v_i of example $\boldsymbol{x}_i$ is first converted to an embedding vector. To achieve this, a vocabulary V is built to contain all the unique metadata values. In our experiment:

$$V = \{\text{playback}, \text{non-playback}\} \tag{1}$$

where streams collected when player is playing music or alarm clock has *playback* in their metadata, and *non-playback* denotes inactive player state metadata.

Given a matadata value v_i, it is first encoded to an one-hot vector $\mathcal{I}_i \in \{0,1\}^{|V|}$ through a lookup table generated by V, and then transformed to metadata embedding $e_i \in \mathbb{R}^L$ by a learnable dense embedding layer:

$$e_i = f(\mathcal{I}_i) \tag{2}$$

where f is one-hidden layer MLP, and L is output embedding length.

2.2. Conditional Batch Normalization

Batch Normalization (BN) normalizes feature maps for each input mini-batch to reduce the internal covariant shift. It not only allows for higher learning rates which accelerates the training, but also makes the model less sensitive to parameter initialization [16]. BN can be described as:

$$BN(\boldsymbol{F}_{i,c,w,h}|\gamma_c,\beta_c) = \gamma_c \frac{\boldsymbol{F}_{i,c,w,h} - \mathbb{E}_{\mathcal{B}}[\boldsymbol{F}_{\cdot,c,\cdot,\cdot}]}{\sqrt{\text{Var}_{\mathcal{B}}[\boldsymbol{F}_{\cdot,c,\cdot,\cdot}] + \epsilon}} + \beta_c \tag{3}$$

where $\mathcal{B} = \{\boldsymbol{F}_{i,\cdot,\cdot,\cdot}\}_{i=1}^N$ is a mini-batch with N samples, and $\boldsymbol{F}_{i,c,w,h}$ corresponds to the c^{th} channel at location (w,h) of feature maps of i^{th} sample. γ_c and β_c are the scale and offset for adjusting the normalization. ϵ is a constant damping factor for numerical stability.

The idea of conditional batch normalization is to learn the γ_c and β_c of BN as a function of input embedding e_i.

$$\hat{\gamma_c} = f_1(e_i) \tag{4}$$
$$\hat{\beta_c} = f_2(e_i) \tag{5}$$

where $f_{1,2}$ can be arbitrary functions to control $\hat{\gamma_c}$ and $\hat{\beta_c}$. In our experiment, they are one-hidden layer MLPs. In this way, the CBN is able to modulate the feature maps independently given different inputs, therefore helps to improve network's generalization ability on heterogeneous data.

Following [18], we use e_i to predict the $\Delta\hat{\gamma}_c$ instead of $\hat{\gamma}_c$ to prevent the feature maps from being zeroed-out by initial $\hat{\gamma}_c$.

$$\hat{\gamma}_c = 1 + \Delta\hat{\gamma}_c = 1 + f_1(e_i) \tag{6}$$

2.3. Feature Concatenation

Feature concatenation is a common approach to fuse muti-modal information from different domains. In order to fuse the metadata information with audio features, the metadata embedding is fed into a fully connected layer and the output is concatenated to the flattened audio feature vector to generate the fused features h_{fuse}.

$$h_{\text{meta}} = MLP(e_i) \tag{7}$$

$$h_{\text{fuse}} = \begin{bmatrix} h_{\text{audio}} \\ h_{\text{meta}} \end{bmatrix} \tag{8}$$

2.4. Model Architecture

The architecture of our proposed model is shown in Figure 2. We add playback status as an additional feature into the CNN network. The playback status is first encoded to an one-hot vector, and then transformed to metadata embedding vector. We replaced the batch normalization layer with the conditional batch normalization, whose scale and offset are predicted by the metadata embedding via separate one-hidden layer MLPs. In addition, the metadata features are concatenated to the audio features to generate a joint input feature for the dense layers of the network. In order to achieve the end-to-end decoding excluding HMM, the segment-level posteriors are smoothed by a sliding window, and then compared with pre-defined threshold to detect the presence of wakeword. The structure of the CNN is shown in Table 1. The input features are LFBE and playback name. Each example used in training has LFBE feature size of 76×64.

Table 1: *CNN Network Architectures*

layer-number	1	2	3	4	5	6	7	8	9
filter-heights	9	7	4	3	3	3	1	1	1
filter-widths	5	3	3	3	3	3	1	1	1
stride-heights	1	3	1	1	1	1	1	1	1
stride-widths	1	1	1	1	1	1	1	1	1
pooling-heights	2	1	1	1	1	1	1	1	1
pooling-widths	3	2	1	1	1	1	1	1	1
num-filters	96	128	128	160	160	500	500	500	2

3. Experiments

3.1. Experimental Setup

The objective of the experiments is to compare and quantify performance change in our keyword spotting model when trained and evaluated with additional playback metadata using CBN and feature concatenation. Our model architecture is defined in Table 1. Our baseline model (without conditional batch normalization layers and feature concatenation) is trained on all data without metadata labels.

3.1.1. Keyword Datasets with playback metadata

The training and test datasets consist of anonymized Alexa data, including stream audio, wakeword-related annotation, and playerback metadata. No customer identity or stream-level transcription information are used. The audio streams in the datasets contain playback metadata which partitions the dataset in two set of streams, playback and non-playback. Playback streams are keyword initiated streams when the device speaker is ON (device is playing audio via its speaker). Accordingly, non-playback streams are keyword initiated streams when the device speaker is OFF. The statistics of the datasets are shown in Table 2. To conduct a fair comparison between the baseline and candidate models, same set of audio streams are used to train baseline and candidate models.

Table 2: *Dataset with metadata statistics*

Condition	**# Training Streams**	**# Test Streams**
playback streams	9,163,611	596,762
non-playback streams	25,534,833	1,659,542
Total	34.6 Million	2.26 Million

3.1.2. Models

In the experiments, four types of models are trained:

- Default CNN, no metadata input (vanilla-cnn; baseline).
- CNN with conditional batch normalization only (cbn).
- CNN with feature concatenation only (concat).
- CNN with both conditional batch normalization and feature concatenation (cbn-concat).

To explore the benefits of incorporating playback embedding as additional features into the CNN, different concatenation layer number $K \in \{6, 7, 8\}$ (fully connected layer) are used in the experiment. The length of playback feature h_{meta} for concatenation is set to two which is the same as metadata embedding. All the experiments are sharing the same training specs described in Section 3.1.3.

3.1.3. Training Specs

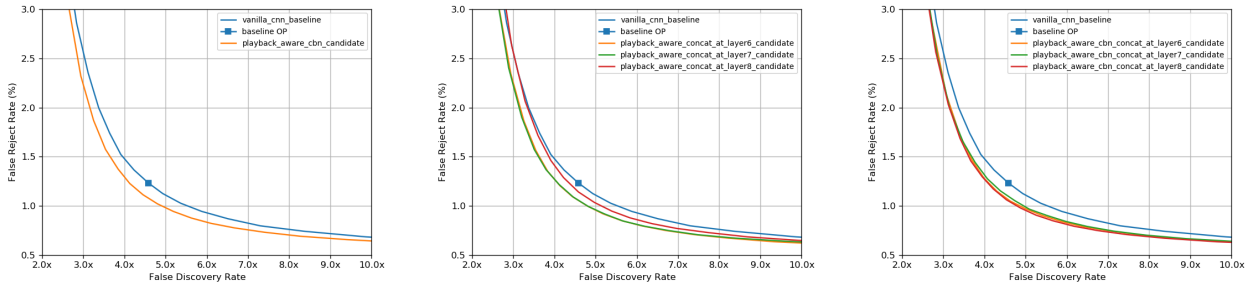
The experiments are implemented on AWS using TensorFlow v1.10.0 [1] under DLAMI 14.0 [2]. The p3.8xlarge instance which has 4 Tesla V100 GPUs is used to train different types of CNNs. We use batch size of 500 samples for each GPU (2,000 in total). The learning rate is set to 0.001, and dropout rate is 0.3. All the models are trained with 100,000 steps. This hyper-parameter set works well for the baseline model. We use cross-entropy as the loss function, and the gradients are averaged from all GPUs before applying weights update. When training is completed, an exponential moving average with decay equals to 0.99 is performed to export final models.

3.1.4. Evaluation

We evaluate baseline and candidate models on both playback and non-playback test datasets in Table 2. In Table 3, we report the false reject rate (FRR) of candidate models by fixing the false discovery rate (FDR) to quantify the improvement of candidate models against the baseline. The fixed FDR is the FDR of baseline model at a reasonably chosen operating point, and the real numbers are obfuscated per Alexa EULA. We also mark the operating points on DET curves in Figure 3 and 4.

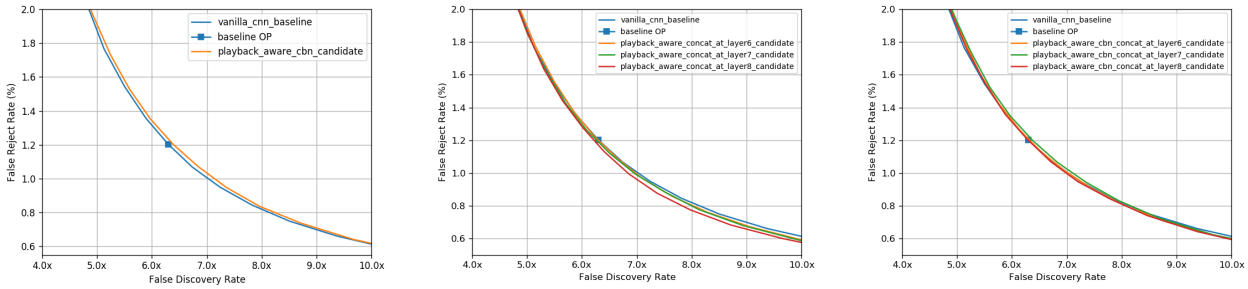
[1] https://www.tensorflow.org/

[2] https://aws.amazon.com/releasenotes/deep-learning-ami-amazon-linux-version-14-0/



(a) *models with CBN* (b) *models with FC* (c) *models with CBN and FC*

Figure 3: *DET curves comparison of candidate models on **playback** dataset. The FDR axis is scaled due to the sensitivity of these information. Best viewed in color.*



(a) *models with CBN* (b) *models with FC* (c) *models with CBN and FC*

Figure 4: *DET curves comparison of candidate models on **non-playback** dataset. The FDR axis is scaled due to the sensitivity of these information. Best viewed in color.*

3.2. Results

Baseline and candidate model comparisons are shown in Figure 3 and 4. Figure 3 suggests conditional batch normalization (CBN) and feature concatenation (FC) independently (3a, 3b) and combined (3c) have a positive impact on the performance of keyword spotting in playback condition. The best candidates from cbn, concat and cbn-concat model achieve 12.2%, 13.82% and 14.63% relative FRR improvement respectively on playback dataset. The candidate models show minimal or no improvement over the non-playback condition. We conjecture that the baseline model performance to begin with is better on non-playback conditions given the substantially higher non-playback stream count in training dataset, this is what leads to a minimal/no improvement of candidate models on non-playback conditions.

4. Conclusion

In this paper, we present a new CNN architecture with metadata embedding for metadata-aware end-to-end keyword spotting. The proposed model utilizes device playback metadata embedding for conditional batch normalization and feature concatenation, and demonstrates relative FRR improvement of 14.63% improvements over the vanilla CNN model on playback datasets. Future work includes adding more types of metadata such as volume, device type, etc as input feature.

Table 3: *False reject rate (FRR) comparison of baseline and candidate models at fixed false discovery rate on playback and non-playback data sets. K represent concatenation layer number. The best scores on each dataset are marked in bold. Lower FRR corresponds to better model performance.*

Model	K	FRR on playback	FRR on non-playback
vanilla-cnn		1.23	1.20
cbn		1.08	1.24
concat	6	1.06	1.20
	7	1.06	1.19
	8	1.14	**1.17**
cbn-concat	6	1.06	1.21
	7	1.10	1.23
	8	**1.05**	1.20

5. Acknowledgements

We would like to thank Varun Nagaraja, Joe Tighe, Nikko Strom and Manoj Sindhwani for insightful discussions and feedback.

6. References

[1] J. R. Rohlicek, W. Russell, S. Roukos, and H. Gish, "Continuous hidden markov modeling for speaker-independent word spotting," in *International Conference on Acoustics, Speech, and Signal Processing,*, May 1989, pp. 627–630 vol.1.

[2] L. R. Rabiner, "A tutorial on hidden markov models and selected ap-

plications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, Feb 1989.

[3] B.-H. Juang, S. Levinson, and M. Sondhi, "Maximum likelihood estimation for multivariate mixture observations of markov chains (corresp.)," *IEEE Transactions on Information Theory*, vol. 32, no. 2, pp. 307–309, 1986.

[4] L. Deng, P. Kenny, M. Lennig, V. Gupta, F. Seitz, and P. Mermelstein, "Phonemic hidden markov models with continuous mixture output densities for large vocabulary word recognition," *IEEE Transactions on Signal Processing*, vol. 39, no. 7, pp. 1677–1681, July 1991.

[5] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury *et al.*, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal processing magazine*, vol. 29, 2012.

[6] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on audio, speech, and language processing*, vol. 20, no. 1, pp. 30–42, 2012.

[7] T. N. Sainath, A.-r. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for lvcsr," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 8614–8618.

[8] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on audio, speech, and language processing*, vol. 22, no. 10, pp. 1533–1545, 2014.

[9] T. Sainath and C. Parada, "Convolutional neural networks for small-footprint keyword spotting," 2015.

[10] Y. LeCun, Y. Bengio *et al.*, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, p. 1995, 1995.

[11] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 6645–6649.

[12] A. Graves, N. Jaitly, and A.-r. Mohamed, "Hybrid speech recognition with deep bidirectional lstm," in *2013 IEEE workshop on automatic speech recognition and understanding*. IEEE, 2013, pp. 273–278.

[13] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[14] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ser. ICML'11. Madison, WI, USA: Omnipress, 2011, p. 689–696.

[15] J. M. Facil, B. Ummenhofer, H. Zhou, L. Montesano, T. Brox, and J. Civera, "Cam-convs: camera-aware multi-scale convolutions for single-view depth," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 11 826–11 835.

[16] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[17] H. De Vries, F. Strub, J. Mary, H. Larochelle, O. Pietquin, and A. C. Courville, "Modulating early visual processing by language," in *Advances in Neural Information Processing Systems*, 2017, pp. 6594–6604.

[18] E. Perez, H. De Vries, F. Strub, V. Dumoulin, and A. Courville, "Learning visual reasoning without strong priors," *arXiv preprint arXiv:1707.03017*, 2017.

[19] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, "Multimodal machine learning: A survey and taxonomy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 2, pp. 423–443, 2019.