Contextual Acoustic Barge-In Classification for Spoken Dialog Systems

Dhanush Bekal, Sundararajan Srinivasan, Sravan Bodapati, Srikanth Ronanki, Katrin Kirchhoff

AWS AI Labs

{dkannang, sundarsr, sravanb, ronanks, katrinki}@amazon.com

Abstract

In this work, we define barge-in verification as a supervised learning task where audio-only information is used to classify user spoken dialogue into true and false barge-ins. Following the success of pre-trained models, we use low-level speech representations from a self-supervised representation learning model for our downstream classification task. Further, we propose a novel technique to infuse lexical information directly into speech representations to improve the domain-specific language information implicitly learned during pre-training. Experiments conducted on spoken dialog data show that our proposed model trained to validate barge-in entirely from speech representations is faster by 38% relative and achieves 4.5% relative F1 score improvement over a baseline LSTM model that uses both audio and Automatic Speech Recognition (ASR) 1-best hypotheses. On top of this, our best proposed model with lexically infused representations along with contextual features provides a further relative improvement of 5.7% in the F1 score but only 22% faster than the baseline.

Index Terms: Spoken dialog systems, barge-in, speech representations.

1. Introduction

Voice based chat-bots have become a ubiquitous part of commercial goal oriented dialogue systems [1, 2, 3]. These dialog systems in general have an additional barge-in feature [4] which allows customers to interrupt the ongoing dialogue at any point of time to reach the goal as quickly as possible. However, barge-in systems [4, 5] are quite sensitive to background speech and non chat-bot directed customer speech (e.g., whispering or talking to others) resulting in a high number of false positives. Reducing these false positives helps improve customer experience by avoiding unnecessary processes in the downstream dialog system. In our work, we define true barge-in as those speech utterances with interruptions directed at the chat-bot and any other detected speech is classified as a false barge-in (see Table 1). This is akin to device-directed speech detection applied to close talk chat-bot dialogue systems with an added task of barge-in (or interruption) detection. However, our task is different from device directed speech detection in the literature [6, 7, 8] as we do not deal with far field audio or wake-word recognition.

Traditionally, barge-in is handled in three stages: detection, verification, and recovery [4]. Detection is the identification of speech input from the user and the problems here include accurately detecting foreground voice activity in the presence of background speech, noise, music, and other audio events. Verification means determining whether the utterance is actually meant to be a barge-in, or something else (e.g., non chat-bot directed speech) and generally requires information about the lexical content of the utterance. Some systems [9] use confidence scores from incremental ASR results (partial decoding hypotheses) for this purpose, others also make use of NLU results, or use special language model arcs. It has been shown that model-based verification improves barge-in detection over pure voice activity detection (VAD) based approaches [5]. Including user-specific information such as typical barge-in rates and average ASR accuracy for a given user have also been shown to help during the verification stage [10].

Table 1: *Examples of True and False Barge-in in a dialogue*

No barge-In	**Bot** : Would you like to book a car ? **User** : Yes **Bot** : Would you like a Sedan, SUV, Hatchback ? **User** : Sedan
True Barge-In	**Bot** : Would you like to book... **User** : Yes **Bot** : Would you like a Sedan... **User** : Sedan
False Barge-In	**Bot** : Would you like a Sedan... **User** : Can you get me a coffee?

One potential drawback of ASR/NLU model-driven verification is latency and in general, human-computer interaction (HCI) is significantly impacted by delayed responses from a cascaded spoken dialogue system. Also, NLU models often employ data-efficient techniques to make them robust to ASR errors. Hence, end-to-end (E2E) spoken language understanding (SLU) solutions have recently been proposed to address cascading errors as well as to decrease latency [11, 12, 13]. Similarly, our work focuses on using audio and other dialogue information available prior to an ASR system for barge-in verification. Although E2E solutions are elegant and straightforward, they are often not considered for practical use due to a gap in performance over a cascaded system. In this work, we show that self-supervised representation learning models like Hidden-Unit BERT (HuBERT) [14] that implicitly encode linguistic information can be utilized for the barge-in classification task without compromising on performance. Experimental results show that our best proposed model to validate barge-ins directly from the audio is not only significantly better (10.4% relative F1 improvement: 73.6 to 81.3) but also 22% faster than a baseline system that uses both audio and ASR hypothesis. The main contributions of our work are summarized as follows:

- Pre-trained speech representations derived directly from the audio can be utilized for barge-in classification with improved accuracy and latency over a baseline model that uses lexical information from ASR hypothesis.
- We introduce a novel technique to infuse lexical information directly into HuBERT-based speech representations and show that it helps improve the performance of our downstream end-to-end SLU classification task.
- We show that contextual features such as bot prompt and dialogue context play a prominent role in identifying true barge-in interruptions.

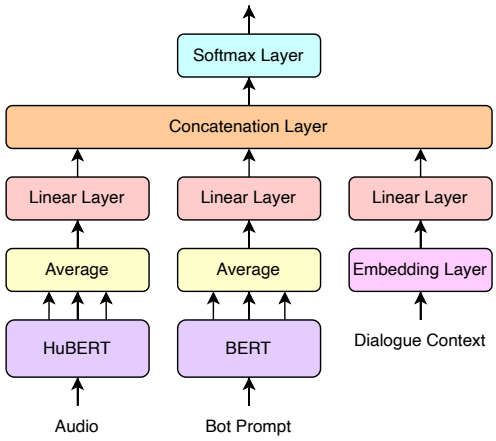

Figure 1: *Schematic diagram of contextual barge-in model*

2. Proposed Approach

Figure 1 illustrates the contextual model we use for the barge-in classification task. The model encodes three types of input features: (i) audio encoder using a pretrained HuBERT model, (ii) bot prompt encoder using a pretrained BERT model, and (iii) dialogue context encoder using a trainable embedding layer. All three encoded representations are joined by a concatenation layer and then passed through a softmax layer for classification.

2.1. Audio Encoder for Speech Representation

Self-supervised learning (SSL) has shown prominent results for speech processing, especially on phoneme classification and ASR [15, 16, 17]. Recently, [18] proposed the SUPERB benchmark [1] to evaluate SSL models across different tasks and as per the results, HuBERT [14] and WavLM [19] (both use masked prediction loss) enjoy the best generalization ability in the overall evaluation. HuBERT utilizes an offline clustering step to generate noisy target labels for a BERT-like pre-training [20]. The loss function used for HuBERT pre-training is a combination of unmasked and masked prediction loss. The cross entropy loss computed only over unmasked time steps is similar to acoustic modeling and the loss computed only over masked time steps where the model has to predict the targets corresponding to unseen frames from context, is analogous to language modeling. This way the HuBERT model learns both the acoustic representation of unmasked segments and the long-range temporal structure of the speech data [21]. Prior literature shows that linguistic information helps in tasks like device directed speech detection, and therefore we leverage these implicit language representations for our barge-in classification task.

Let X denote a speech utterance $X = \{x_1, x_2, ...x_T\}$ of T frames. The HuBERT-based audio encoder transforms the speech sequence X into a sequence of hidden representations $H_x = \{h_{x_1}, h_{x_2}, ...h_{x_M}\}$. We then compute the mean of the ensemble and pass it through a linear layer to obtain a single speech representation r_x, which is defined as:

$$r_x = W^x(\frac{1}{T}\sum_{i=1}^{T} h_{x_i}) \quad (1)$$

where W^x denotes the weight matrix of the linear layer. For audio-only experiments presented in section 5, this speech representation is directly passed through a softmax layer.

[1] https://superbbenchmark.org/leaderboard

2.2. Bot Prompt and Dialog Context Encoder

Spoken dialog systems typically span multiple turns of back and forth between a user and a bot. These interactions mostly adhere to a dialogue structure that can be determined beforehand. *Bot prompt* and *dialogue context* are two such contextual dialogue inputs available prior to processing user utterance for barge-in.

Bot Prompt Representation: Bot prompts are the questions asked by a chat-bot to user in spoken dialog systems. Examples of a single turn bot prompt and user response are given in Table 1. The intuition behind using bot prompt information is that they are correlated to the user response in true barge-in cases i.e., users interrupt with responses that are expected by the chat-bot. For example, when the chat-bot is asking a question, there is a higher chance of interruption in certain dialogue states if the user has knowledge of the interaction. In order to extract prompt representations, we tokenize bot prompts into sub-words and encode them using a pretrained BERT model [20].

Let P denote a bot prompt utterance $P = \{p_1, p_2, ...p_N\}$ of N tokens. The BERT-based bot prompt encoder transforms the prompt sequence P into a sequence of hidden representations $H_p = \{h_{p_1}, h_{p_2}, ...h_{p_N}\}$. The ensembeled average is then passed through a linear layer whose weights are defined by W^p to obtain a single prompt representation r_p, which is defined as:

$$r_p = W^p(\frac{1}{N}\sum_{i=1}^{N} h_{p_i}) \quad (2)$$

Dialogue Context Representation: We create dialogue context labels associated with each bot prompt. These dialogue context labels indicate whether the chat-bot expects an intent, and a slot in the user response to the bot prompt. We use 3 intents and 7 slots resulting in a total of 10 unique dialogue context labels. To represent these dialogue contexts (r_d), we embed them using a trainable embedding layer followed by a linear projection layer and is defined as:

$$r_d = W^d \bar{E} D \quad (3)$$

where $\bar{E} \in \mathbb{R}^{dxm}$ is the dialogue context embedding matrix. m and d are the dialogue embedding dimensionality and the number of unique dialog context labels respectively.

2.3. Combined Model

As shown in figure 1, all three representations are concatenated and then passed through a feed forward layer with *tanh* activation followed by a softmax layer to get the final output ($\hat{y}$):

$$r_c = tanh(W^c(r_x \oplus r_p \oplus r_d) + b^c) \quad (4)$$

$$\hat{y} = softmax(W^y r_c + b^y) \quad (5)$$

where W^c, W^y, b^c, b^y denote weight matrices and bias of the concatenation and classification layer respectively. The model is trained using a cross entropy loss function.

3. Language Infusion into Speech Representation

For improving the semantic information implicitly learned during HuBERT pre-training, we propose a novel approach to infuse lexical information directly into speech representations. Figure 2 shows the overview of our language infusion approach. For each training utterance, the corresponding time-aligned word transcript is obtained either using forced alignment of human labels or using an ASR system that returns

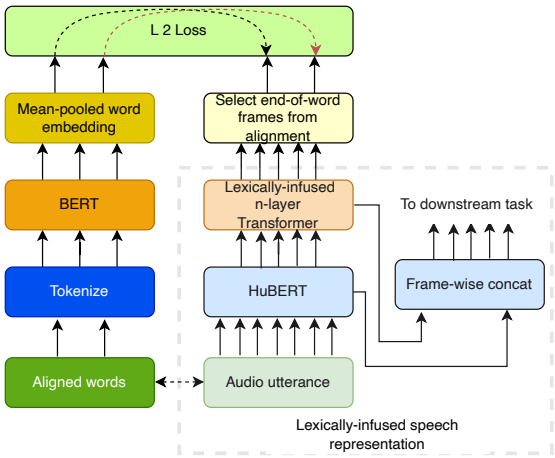

Figure 2: *Training to inject lexical information into speech representation*

words with time-stamps. The words are tokenized to sub-words and run through a pre-trained BERT model. Each word w is then represented by the mean of the BERT embeddings corresponding to the sub-word tokens belonging to the word. On the acoustic side, the speech representation is first extracted from a pre-trained HuBERT-base model. The frame-level features are then passed through a multi-layer transformer [22] (this is the core language-infusion sub-network) to predict the word-level average BERT embeddings. Only frames corresponding to word end times tw (obtained from force-alignment) are used to minimize the L2 loss between the frame-level predictions and the corresponding word-level embeddings. When using explicit language layers, we do not freeze the HuBERT weights during training. In our experiments, we show that we can directly infuse language information into HuBERT weights without using language layers. Let b_e represent the BERT embeddings for tokens in the word, then the word embedding $embed_w$ is defined as:

$$embed_w = mean(b_e) \tag{6}$$

The loss to train the language-infused network is:

$$L_{utt} = \Sigma_{words}((p(f_{tw}) - embed_w)^2) \tag{7}$$

where f_{tw} is the closest frame corresponding to the word ending, p is the output of the network.

When using language layers, the representations of the final language-infused transformer layer and HuBERT's final layer are concatenated and used in place of the HuBERT representations of the downstream task. When no language layers are used, we use the language infused HuBERT representations as inputs for barge-in classification. This technique has similarities to SpeechBERT [23], however SpeechBERT still requires ASR at the inference stage. Our technique is closer to SPLAT [24], but SPLAT only matched the CLS token embedding of BERT with the first frame output of speech representation, which we believe may not lead to as strong an association between the two modalities as in our model. We plan to compare the efficacy of SPLAT and our approach in the future.

4. Experiments

4.1. Data

Since spoken dialogue systems can be utilized in sensitive public environments, we did not consider home device directed speech data sets for experimentation. Keeping the barge-in task in mind, we have collected 12000 single turn chat-bot prompts, and user response pairs in various environments such as home, car, malls, and streets. The chat-bot prompt consists of multiple simulated questions and users respond to those questions through a simulated false or true barge-in. Users are asked to simulate a true interruption by cutting off the chat-bot with chat-bot directed utterances and false interruptions by speaking random utterances. To create realistic dialog simulation, we provided users with a large list of false interruptions and uncorrelated utterances to choose from and asked them to mimic non chat-bot directed speaking style. For each of these single turn pairs, we also annotate an associated dialogue context. Out of the total 12000 data pairs collected, we made three splits: train (9000), validation (1000) and test (2000). All three splits have balanced true and false barge-in classes and the audio data is upsampled to 16kHz (for experimentation with pretrained models) with each audio spanning an average of 2.4 seconds [25].

4.2. Model Configurations

For baseline systems, we trained a VAD model using TDNN architecture [26] and a multi-layer LSTM model for barge-in classification. For speech representation, HuBERT [14] offers three different configurations with varying number of parameters, transformer encoder blocks and embedding size: HuBERT Base (95M, 12 layers, 768), Large (317M, 24 layers, 1024), and X-Large (964M, 48 layers, 1280). Keeping latency in mind, we have chosen the HuBERT Base model which was pretrained on 16kHz Librispeech standard 960hr training data [27] for experimentation. The pre-trained model used for chat-bot prompt representation is a 12-layer *BERT-base-uncased* model with 768 dimensional output from Huggingface [28]. For dialogue context representation, we used a trainable embedding layer with 64 dimensional output. We project all three representations to 128 dimensional vectors using separately trained linear layers. For training barge-in models, Stochastic Gradient Descent (SGD) with a learning rate of 5e-4 and a dropout of 0.2 was applied. The *Lexically-Infused HuBERT* (LI-HuBERT) model used the same Librispeech pre-trained Hubert-Base model and the BERT model was frozen during pre-training. The transformer layers of the core language-infusion sub-network are similar to that of transformer layers in the HuBERT-Base model. We experimented with and without additional language layers where we infuse language information directly into the HuBERT-Base model. For LI-HuBERT pre-training, we used an in-house conversational dataset consisting of 90k utterances, with each utterance averaging 5s and the word time-stamps for the same are generated from a Kaldi-based hybrid ASR model [29]. For training, we used ADAM optimizer with a learning rate of 2e-4, and gradient norm was clipped to 5. The model was trained for 800k steps, with a batch size of 16.

To measure latency, we used an AWS EC2 M5 instance with 64GB memory. We utilized the Torch library to load the models on CPU that were previously trained and saved on GPU. We evaluated each model on the aforementioned test split by setting the batch size to 1 and the latency numbers shown in the results section are measured in milliseconds (ms).

5. Results and Discussion

The performance of each system is measured by metrics such as average recall and F1 along with latency. First, we present results for a baseline barge-in system with audio (filter-bank fea-

Table 2: *Performance of VAD and baseline Barge-in system with audio (filterbank features) and ASR hypotheses as inputs*

Model	Input	Recall	F1	Latency
TDNN (VAD)	Audio	51.6	65.4	-
LSTM	Audio	60.7	61.3	20
	ASR Hypothesis	72.1	72.5	219
	+ Audio	73.0	73.6	226

tures) and ASR hypotheses as inputs in Table 2. We observe that using a VAD alone doesn't solve the problem as it can not disambiguate the cases of speech not intended for the chat-bot further motivating barge-in classification task. Also, we observe that using ASR hypotheses along with audio provides significant improvement over a pure audio based approach showing that lexical information plays an important role in barge-in classification. However, the latency is increased by 10-fold from 20ms to 226ms as the computation includes the time for ASR decoding and therefore we do not consider ASR hypotheses for rest of the experiments in this paper.

Table 3 provides the results for the barge-in classification task with both frozen and fine-tuned HuBERT representations. As expected, the model trained with HuBERT frozen representations did not perform that well as it was pre-trained only on Librispeech data and is quite diverse from spoken dialogue data used for this task. The fine-tuned HuBERT model using only audio as input outperformed the baseline system that uses lexical information from ASR hypotheses in terms of both F1 (4.5% relative improvement: 73.6 to 76.9) and latency (reduced from 226ms to 140ms).

Table 3: *Performance of HuBERT based barge-in model augmented with contextual features*

Model	Input	Recall	F1	Latency
HuBERT (Frozen)	Audio	60.6	62.3	140
	+ Bot Prompt	63.5	64.0	172
	+ Dialogue Ctx	65.0	64.0	143
	+ Bot Prompt	64.6	65.9	176
HuBERT (Fine-Tuned)	Audio	75.7	76.9	140
	+ Bot Prompt	76.1	76.9	172
	+ Dialogue Ctx	76.6	75.0	143
	+ Bot prompt	77.6	77.7	176

Performance with contextual information: We now compare the performance of our HuBERT based barge-in models when augmented with spoken dialogue contextual features. From Table 3, we observe that adding contextual cues such as bot prompt and dialogue context helps improve the performance of both frozen and fine-tuned HuBERT models. With fine tuning, the best performing model with both dialogue context and bot prompts achieves 2.5% relative improvement in recall (improved from 75.7 to 77.6). This shows that the bot prompt and dialogue contexts aid in classification, especially when they have correlation with the user responses. Though adding these contextual representations led to an increase in latency, the best performing model is still 22% faster than the baseline model that uses ASR hypotheses.

Performance with language infusion: In this section, we discuss the improvements obtained from pre-training with language infusion. From Table 4, we observe that our proposed *LI-HuBERT* model to validate barge-ins directly from the audio is not only significantly better (7.5% relative F1 improvement: 73.6 to 79.1) but also 38% faster than a baseline system that uses lexical information from ASR hypotheses. We also observe that adding additional language layers for language infusion yields improvements in barge-in classification with increased latency. We can attribute these improvements to the additional parameters incurred with additional layers to learn language information. Finally, we combine the contextual features with our *LI-HuBERT* model in order to validate if each of the techniques would yield additional gains when applied together. From Table 4, it is clear that using lexically infused speech representations with contextual embeddings provides us with the best performance (10.4% relative F1 improvement: 73.6 to 81.3). Overall, the improved performance suggests that language infusion can help carry sufficient semantic information to distinguish between true and false barge-ins.

Table 4: *Performance of Lexically Infused HuBERT (LI-HuBERT) with language layers and contextual features*

Model	Input	Recall	F1	Latency
LI-HuBERT (Fine-Tuned)	Audio	78.8	79.1	140
+ 2 layers		79.7	79.9	144
+ 4 layers		80.2	80.3	152
LI-HuBERT (Fine-Tuned)	+ Bot Prompt	80.2	81.1	172
	+ Dialogue Ctx	78.9	79.0	143
	+ Bot Prompt	81.3	81.3	176

Qualitative Analysis: For further analysis, we listened to some of the audios from test data pertaining to misclassified examples from each model. We observed that the VAD model has very high false barge-in rate (indicated by low average recall), as it cannot differentiate non chat-bot directed utterances. The baseline model that uses ASR hypotheses has relatively poor true barge-in detection rate, especially in cases of longer utterances due to ASR errors. Our proposed model has better detection under these conditions as the model does not rely on ASR performance, and rather relies on pretraining step for linguistic knowledge infusion. However, our proposed model has poor false barge-in classification performance under certain noise conditions (e.g., audio cuts) and can be mitigated to a certain degree by performing data augmentation with external noise during training, and we will pursue this as future work.

6. Conclusion

In this work, we introduced the task of contextual acoustic barge-in where we detect user interruptions in close-talk environments. For this task we have proposed a classification approach using HuBERT features. Our experiments have shown that fine-tuned HuBERT features achieve better performance compared to an ASR hypothesis based models with significantly lower latency. We have also shown that using dialogue information available from a chat-bot prompt and dialogue context further improves performance with some latency hit. We further proposed a new technique for infusing language information over HuBERT representations which gave us the best performance. In our future work, we will explore further improving the robustness of existing solution.

7. References

[1] R. Pieraccini, D. Suendermann, K. Dayanidhi, and J. Liscombe, "Are we there yet? research in commercial spoken dialog systems," in *International Conference on Text, Speech and Dialogue*. Springer, 2009, pp. 3–13.

[2] P. Budzianowski, T.-H. Wen, B.-H. Tseng, I. Casanueva, S. Ultes, O. Ramadan, and M. Gašić, "MultiWOZ–A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling," *arXiv preprint arXiv:1810.00278*, 2018.

[3] E. Hosseini-Asl, B. McCann, C.-S. Wu, S. Yavuz, and R. Socher, "A simple language model for task-oriented dialogue," *Advances in Neural Information Processing Systems*, vol. 33, pp. 20 179–20 191, 2020.

[4] N. Ström and S. Seneff, "Intelligent barge-in in conversational systems." in *INTERSPEECH*, 2000, pp. 652–655.

[5] R. C. Rose and H. K. Kim, "A hybrid barge-in procedure for more reliable turn-taking in human-machine dialog systems," in *2003 IEEE Workshop on Automatic Speech Recognition and Understanding (IEEE Cat. No. 03EX721)*. IEEE, 2003, pp. 198–203.

[6] E. Shriberg, A. Stolcke, D. Hakkani-Tür, and L. Heck, "Learning when to listen: Detecting system-addressed speech in human-human-computer dialog," in *Proc. Interspeech*, 2012, pp. 334–337.

[7] S. H. Mallidi, R. Maas, K. Goehner, A. Rastrow, S. Matsoukas, and B. Hoffmeister, "Device-directed utterance detection," *arXiv preprint arXiv:1808.02504*, 2018.

[8] A. Norouzian, B. Mazoure, D. Connolly, and D. Willett, "Exploring attention mechanism for acoustic-based classification of speech utterances into system-directed and non-system-directed," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7310–7314.

[9] E. Selfridge, I. Arizmendi, P. A. Heeman, and J. D. Williams, "Continuously predicting and processing barge-in during a live spoken dialogue task," in *Proceedings of the SIGDIAL 2013 Conference*, 2013, pp. 384–393.

[10] K. Komatani and H. G. Okuno, "Online error detection of barge-in utterances by using individual users' utterance histories in spoken dialogue system," in *Proceedings of the SIGDIAL 2010 Conference*, 2010, pp. 289–296.

[11] Y. Tian and P. J. Gorinski, "Improving end-to-end speech-to-intent classification with Reptile," *arXiv preprint arXiv:2008.01994*, 2020.

[12] Y. Cao, N. Potdar, and A. R. Avila, "Sequential end-to-end intent and slot label classification and localization," *arXiv preprint arXiv:2106.04660*, 2021.

[13] M. Saxon, S. Choudhary, J. P. McKenna, and A. Mouchtaris, "End-to-end spoken language understanding for generalized voice assistants," *arXiv preprint arXiv:2106.09009*, 2021.

[14] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units," *arXiv preprint arXiv:2106.07447*, 2021.

[15] A. Van den Oord, Y. Li, O. Vinyals *et al.*, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, vol. 2, no. 3, p. 4, 2018.

[16] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *arXiv preprint arXiv:2006.11477*, 2020.

[17] S. Ling and Y. Liu, "Decoar 2.0: Deep contextualized acoustic representations with vector quantization," *arXiv preprint arXiv:2012.06659*, 2020.

[18] S.-w. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin *et al.*, "Superb: Speech processing universal performance benchmark," *arXiv preprint arXiv:2105.01051*, 2021.

[19] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *arXiv preprint arXiv:2110.13900*, 2021.

[20] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[21] S. Srinivasan, Z. Huang, and K. Kirchhoff, "Representation learning through cross-modal conditional teacher-student training for speech emotion recognition," *arXiv preprint arXiv:2112.00158*, 2021.

[22] 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.

[23] Y.-S. Chuang, C.-L. Liu, H.-Y. Lee, and L.-s. Lee, "Speechbert: An audio-and-text jointly learned language model for end-to-end spoken question answering," *arXiv preprint arXiv:1910.11559*, 2019.

[24] Y.-A. Chung, C. Zhu, and M. Zeng, "SPLAT: Speech-language joint pre-training for spoken language understanding," *arXiv preprint arXiv:2010.02295*, 2020.

[25] I. Guyon, J. Makhoul, R. Schwartz, and V. Vapnik, "What size test set gives good error rate estimates?" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 52–64, 1998.

[26] M. Sugiyama, H. Sawai, and A. H. Waibel, "Review of TDNN (time delay neural network) architectures for speech recognition," in *1991., IEEE International Sympoisum on Circuits and Systems*, 1991, pp. 582–585.

[27] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an ASR corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[28] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of EMNLP: System Demonstrations*, 2020, pp. 38–45.

[29] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi Speech Recognition Toolkit," in *Workshop on Automatic Speech Recognition and Understanding*, 2011.