

END-TO-END SPOKEN LANGUAGE UNDERSTANDING USING JOINT CTC LOSS AND SELF-SUPERVISED, PRETRAINED ACOUSTIC ENCODERS

Jixuan Wang Martin Radfar Kai Wei Clement Chung

Amazon Alexa AI

ABSTRACT

It is challenging to extract semantic meanings directly from audio signals in spoken language understanding (SLU), due to the lack of textual information. Popular end-to-end (E2E) SLU models utilize sequence-to-sequence automatic speech recognition (ASR) models to extract textual embeddings as input to infer semantics, which, however, require computationally expensive auto-regressive decoding. In this work, we leverage self-supervised acoustic encoders fine-tuned with Connectionist Temporal Classification (CTC) to extract textual embeddings and use joint CTC and SLU losses for utterance-level SLU tasks. Experiments show that our model achieves 4% absolute improvement over the state-of-the-art (SOTA) dialogue act classification model on the DSTC2 dataset and 1.3% absolute improvement over the SOTA SLU model on the SLURP dataset.

Index Terms— Spoken language understanding, intent classification, dialogue act classification

1. INTRODUCTION

Spoken language understanding (SLU) models infer semantics from spoken utterances [1–3]. Common SLU tasks include intent classification, slot filling and dialogue act classification. Traditionally, SLU models consist of an ASR model that transcribes audio signals into text and a natural language understanding (NLU) model that extracts semantics from the text [4]. Since the ASR and NLU models are optimized independently, the errors from the ASR component will be propagated to the NLU component. For example, if “turn on TV” is recognized as “turn off TV” by ASR, it will be challenging for NLU to predict the right intent.

Recently, there is growing interest in building E2E SLU models where the acoustic and textual models are jointly optimized, leading to more robust SLU models [5–16]. Early works [5, 6] learn an utterance-level semantic representation directly from audio signals without performing speech recognition. This is challenging because semantic information may be lost without the textual information. To guide the learning of semantic representations from audio signals, multi-modal losses are used to tie the utterance-level embeddings from a pretrained BERT model and an acoustic encoder [7]. Token-

level cross-modal alignment and joint space learning is studied in [8].

The limitation of the above approaches is that they cannot be used for sequence labeling tasks, like slot filling. To address this issue, another stream of works build unified models, which can be trained end-to-end and used for both intent classification and slot filling. One way to achieve E2E training is to re-frame SLU as a sequence-to-sequence task, where semantic labels are treated as another sequence of output labels besides the transcript [9–12]. Another way is to unify ASR and NLU models and train them together via differentiable neural interfaces [13–16]. One commonly used neural interface is to feed the token level hidden representations from ASR as input to the NLU model [13–16]. [14, 15] utilizes pretrained ASR and NLU models with shared vocabulary. Different neural interfaces are compared and a novel interface for RNN-Transducer (RNN-T) based ASR model is proposed in [16]. However, to produce token-level representations at inference time, those approaches need auto-regressive decoding, which is computationally expensive.

In ASR models trained with the CTC loss, labels are predicted at the audio frame level in parallel, which are more efficient than those requiring auto-regressive decoding [17, 18]. In this work, we use the output of a CTC-based ASR model as input to infer semantics and joint CTC and SLU losses to train the model end-to-end. We show that our approach outperforms both the approaches that infer semantics directly from audio without ASR supervision and the approaches that rely on auto-regressive ASR models. Compared with the approach proposed in [19], our work demonstrates the effectiveness of using acoustic encoders pretrained with self-supervised tasks for E2E SLU, and highlights the importance of joint training with both CTC and SLU losses and the use of logits instead of probabilities as input to extract utterance embeddings. We show that this simple approach achieves SOTA results on three public datasets of three different tasks. Notably, our model outperforms the best reported intent classification accuracy on the SLURP dataset by 1.3%.

2. E2E SLU WITH CTC

Our model mainly consists of two parts as shown in Figure 1: an ASR model and an utterance encoder. The ASR model

is based on an acoustic encoder pretrained by self-supervised tasks and fine-tuned using CTC. In this work, we leverage the pretrained Wav2Vec2.0 [20] and HuBERT [21] models. The output sequence of the ASR model is maxpooled and fed into the utterance encoder, which is based on fully connected layers. The whole model can be trained end-to-end.

Let's denote the audio sequence of an utterance as $X = (x_1, x_2, \dots, x_T)$ and the corresponding transcript as $W = (w_1, w_2, \dots, w_U)$, where T and U are the sequence length of the acoustic features and transcript, respectively. Each utterance is annotated with an utterance-level label y , such as intent or dialogue act. The training data is denoted by $D^{tr} = \{(X_i, W_i, y_i)\}_{i=1}^{|D^{tr}|}$. To train an ASR model on D^{tr} , we would like to maximize the expected conditional probability $\mathbb{E}_{(X,W) \sim p^{tr}} p(W|X)$ with respect to the training data distribution p^{tr} , which is approximated by $\sum_{i=1}^{|D^{tr}|} p(W_i|X_i)$.

CTC does not assume that the alignment between input and output sequences is given but considers all possible alignments when calculating the training loss. For each pair of (X, W) , we calculate $p(W|X)$ as the average probability of all valid alignments $\mathcal{A}_{X,W}$:

$$p(W|X) = \sum_{A \in \mathcal{A}_{X,W}} p(A|X) \quad (1)$$

where $A = (a_1, a_2, \dots, a_T)$ and each a_i can take any value from the set of all possible output tokens and a special token ϵ , which refers to a blank symbol. By removing repeating tokens and ϵ from an alignment A , we can recover the output sequence W , then A is regarded as a valid alignment.

To calculate $p(A|X)$, we use an acoustic encoder to extract a sequence of frame-level hidden representation $H = (h_1, \dots, h_{T'})$. Note T' will be smaller than T if the acoustic encoder contains subsampling layers, but we assume $T' = T$ for notation simplicity. Frame level prediction is given by:

$$p(a_i|X) = \text{softmax}(Wh_i + b), i = \{1, 2, \dots, T\}. \quad (2)$$

where W and b are parameters of a linear classifier.

The conditional probability $p(W|X)$ can be efficiently calculated through dynamic programming and used as training objective to optimize the ASR model. The ASR objective is defined by:

$$\mathcal{L}^{CTC} = \frac{1}{|D^{tr}|} \sum_{(W,X) \in D^{tr}} -p(W|X), \quad (3)$$

To predict utterance level labels for tasks like intent classification and dialogue act classification, we use the frame-level logits $(Wh_1 + b, \dots, Wh_T + b)$ as input to the utterance encoder. We also try using the hidden representations H as input to the utterance encoder. We observe the two approaches perform similarly in the experiments.

Let's denote the input to the utterance encoder as $H^u = (h_1^u, \dots, h_T^u)$. Our utterance encoder contains a maxpooling

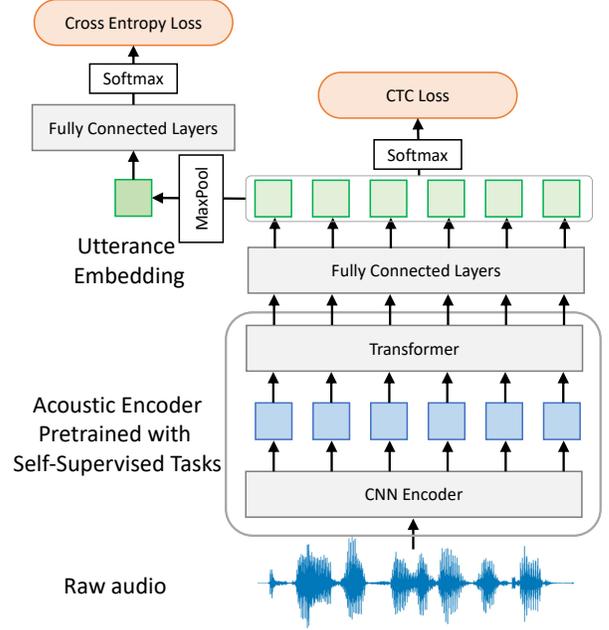


Fig. 1. The block diagram of the proposed approach.

layer and two fully connected layers:

$$\begin{aligned} h^{pool} &= \text{maxpool}(h_1^u, \dots, h_T^u) \\ h^{utt} &= g(W^{(2)}g(W^{(1)}h^{pool} + b^{(1)}) + b^{(2)}) \end{aligned} \quad (4)$$

where $W^{(1)}, b^{(1)}$ and $W^{(2)}, b^{(2)}$ are the parameters of the two fully connected layers, respectively, and g is an activation function. For classification, we use a linear classifier and the cross entropy loss for training:

$$\begin{aligned} p(y|X) &= \text{softmax}(W^u h^{utt} + b^u) \\ \mathcal{L}^{SLU} &= \frac{1}{|D^{tr}|} \sum_{(y,X) \in D^{tr}} -p(y|X). \end{aligned} \quad (5)$$

We train the whole model in a multi-task fashion using a linear combination of \mathcal{L}^{CTC} and \mathcal{L}^{SLU} :

$$\mathcal{L} = \alpha^{CTC} \mathcal{L}^{CTC} + \alpha^{SLU} \mathcal{L}^{SLU} \quad (6)$$

where we treat scalars α^{CTC} and α^{SLU} as hyperparameters.

Good ASR performance is the key to understanding the semantics, so we first fine-tune the ASR model using in-domain data during training, by setting $\alpha^{SLU} = 0, \alpha^{CTC} = 1$. After the ASR model stops improving on the validation set, we set $\alpha^{SLU} > 0, \alpha^{CTC} > 0$, tune their values based on validation performance and train the whole model end-to-end.

3. EXPERIMENTS

We conduct experiments on dialogue act classification, keyword spotting and intent classification tasks, where the inputs

are audio signals and the outputs are utterance-level labels.

3.1. Datasets

We use three public datasets listed in Table 1, showing the number of utterances used for training, validation and testing for each dataset.

Dialogue act classification: We use the DSTC2 dataset [22] for this task, where each utterance is annotated with dialogue acts containing action type, slot name and slot value. Similar to [23], we focus on utterance level classification without considering the adjacent utterances in the same dialogue and combine multiple labels appearing in the same utterance into a single label.

Keyword spotting: We use the Google Speech Commands (GSC) dataset V2 [24], which has 105K snippets in total. Each audio snippet is 1 second long and contains a single word. There are 35 unique words in total that are also used as labels. The goal is to predict the word given an audio snippet.

Intent classification: For this task, we use the SLURP dataset [25]. SLURP is a challenging dataset with linguistically more diverse utterances and a bigger label space across 18 domains. Each utterance is annotated with a *scenario* and *action* label. Following previous works [25, 26], we define *intent* as the combination of *scenario* and *action*. In total, there are 69 unique intents in the training set.

3.2. Baselines

For the *dialogue act classification* task, we use the model proposed in [23] as the baseline, which is the most recent work on this dataset. The authors introduce a neural prosody encoder for dialogue act classification that uses both prosodic features and raw audio signals as input. We follow the same procedure to prepare the DSTC2 dataset.

For the *keyword spotting* task, we compare with a recent work called HTS-AT [27], which is transformer-based model with a hierarchical structure.

For *intent classification*, we compare with the following SOTA models. *ESPNET-SLU*: The ESPnet-SLU toolkit has several SLU recipes for SLURP [26]. Their SLU model is a sequence-to-sequence model where the intent is decoded as one word. We include their results achieved by using pretrained HuBERT as feature extractors for SLURP. *Seo et al., 2022*: This work used a Wav2Vec2.0-based ASR model [15]. The output of the decoder is fed into a RoBERTa-based NLU model for intent classification and slot filling. Each component of this model is first pretrained separately and then fine-tuned jointly. We compare with their results using both the original and synthetic training sets.

3.3. Implementation & Hyperparameters

Our approach can benefit from any pretrained ASR models trained with the CTC loss. We leverage the pretrained

Table 1. Dataset information.

Task	Dataset	Train	Valid	Test
DAC	DSTC2	12,930	1,437	9,116
KWS	Speech Commands	84,843	9,981	11,005
IC	SLURP-Synth	119,881	8,690	13,078

Wav2Vec2.0 [20] and HuBERT [21] acoustic encoders fine-tuned using CTC, as they were used by previous SOTA SLU models [15, 26]. Depending on the size of the datasets, we use a different sized pretrained ASR model. We used the pretrained ASR models provided by TorchAudio [28]. For smaller datasets like DSTC2 and Speech Commands, we used the smallest models available in TorchAudio “WAV2VEC2 ASR BASE 960H” [20]. For more challenging SLURP dataset, we use “HUBERT ASR LARGE” [21]. Both models are pretrained on unlabeled audio data and fine-tuned on the LibriSpeech dataset [29]. For the input to the utterance encoder, we used maxpooling followed by two fully connected layers, each of which has 128 hidden units and uses the GELU activation function [30].

When fine-tuning the ASR models, we stop training if the ASR loss has not been improved for 5 epochs. During joint training, we run training for 50 epochs and select the checkpoint with the best performance on the validation set. Since the tasks we study are all utterance-level classification tasks, we use accuracy as the metric for evaluation. We tune learning rate, batch size and α^{CTC} based on validation performance. For DSTC2, we use learning rate of 0.00001 and batch size of 16 and α^{CTC} of 0.5. For Speech Commands, we use learning rate of 0.00001 and batch size of 128 and α^{CTC} of 1.0. For SLURP, we use learning rate of 0.00005, batch size of 128 and α^{CTC} of 0.5. We use AdamW optimizer implemented in PyTorch with the default configurations.

4. EXPERIMENTS

4.1. Comparison with previous works

The results on the three datasets are shown in Table 2.

Dialogue act classification: Our model achieves 97.6% and 97.5% accuracy on DSTC2 using hidden representations and frame-level logits as input, respectively, which significantly outperforms the previous work [23] that directly predicts dialogue acts from audio input by 4%.

Keyword Spotting: There is little improvement space on GSC V2 as previous work already achieves very good performance. Although our approach is not designed for keyword spotting, it still matches the performance of HTS-AT [27], achieving 98.0% accuracy. Note that all the utterances in GSC V2 only contain a single word, thus the textual information recovered by the ASR model in our approach might not be as helpful as on other datasets, like DSTC2 and SLURP, which contain linguistically more complex utterances.

Table 2. Test results on DSTC2, Speech Commands and SLURP. *Numbers are obtained from the original papers.

Dataset	Approach	Accuracy
DSTC2	Wei et al., 2022 [23]	93.6*
	Ours (Wav2Vec2.0)	
	+ Hidden as input	97.6
	+ Logits as input	97.5
GSC	HTS-AT [27]	98.0*
	Ours (Wav2Vec2.0)	
	+ Hidden as input	98.0
	+ Logits as input	98.0
SLURP	ESPnet-SLU [26]	86.3*
	Seo et al., 2022 [15]	86.9*
	Ours (HuBERT)	
	+ Hidden as input	88.1
	+ Logits as input	88.2

Intent Classification: On the SLURP dataset, our approach achieves 88.1% and 88.2% accuracy using hidden representation and frame-level logits as input, respectively, resulting in 1.3% absolute improvement over the SOTA result.

4.2. Ablations & Analysis

We conduct ablation studies on the SLURP dataset to investigate: 1) whether our E2E models can perform better than cascade models; 2) whether CTC loss is useful for E2E SLU modeling; 3) whether fine-tuning the ASR model for SLU is necessary; 4) whether we need more powerful utterance encoders. Results are listed in Table 3.

HuBERT + BiLSTM/BERT NLU: We train BiLSTM and BERT-based NLU model on the one-best hypothesis generated by the fine-tuned HuBERT ASR model. The number of parameters in the BiLSTM and BERT NLU model is 24.7M and 110M, respectively. The textual encoder of our model only contains two fully connected layers (20K parameters), which is much smaller than the BiLSTM and BERT models and trained from scratch without pretraining on textual data as by BERT. Still, our approach outperforms the two cascade approaches: the test accuracy of the two approaches is 83.95% and 87.44%, respectively, while ours is 88.18%.

HuBERT + linear classifier w/o ASR loss: The only difference between this approach and our proposed approach is that this approach does not utilize the CTC loss for SLU but predicts SLU labels directly from audio signals. The test accuracy of this approach is 84.81%, which is lower than both our approach and the previous work based on the LAS ASR model [15]. This result demonstrates the importance of the textual information recovered by the ASR loss for SLU.

HuBERT (frozen) + utterance encoder: After fine-tuning the ASR model on the in-domain data, instead of further fine-tuning it with the SLU task, we freeze the ASR

Table 3. Ablation study results on SLURP.

Approach	Accuracy
HuBERT + BiLSTM NLU	83.95
HuBERT + BERT NLU	87.44
HuBERT + linear classifier w/o ASR loss	84.81
HuBERT (frozen) + utterance encoder	72.16
HuBERT with probability as input	87.00
HuBERT with CNN textual encoder	88.05
Ours (HuBERT with logits as input)	88.18

model and only train the utterance encoder. This approach only achieves 72.16% accuracy. This shows that further fine-tuning the ASR model for SLU tasks in an E2E fashion is one of the keys to good performance.

HuBERT with probabilities as input Similar with previous work [19], we experiment with using probabilities after the softmax layer as input to the utterance encoder. This approach performs worse than using logits or hidden representations as input, resulting in 87.0% accuracy.

HuBERT with CNN textual encoder: We investigate whether using a more powerful CNN-based utterance encoder can lead to better performance. We apply four CNN layers with different kernel sizes on top of the logits and apply max pooling and fully-connected layers to extract the utterance-level embeddings. We observe that although the CNN encoder (1.9M parameters) is much larger than our utterance encoder (20K parameters), the accuracy using the CNN encoder is 88.05% and slightly worse than our approach.

We further conduct analysis on ASR performance. The WER and CER of the ASR model before SLU training are 18.2% and 8.5%, respectively, while they decrease to 17.4% and 7.8%, respectively, after SLU training. This shows that the SLU loss does not conflict with the ASR loss but can benefit the ASR performance. On the other hand, our model can still predict the intent labels correctly for 83.4% of the utterances containing ASR errors, showing the robustness of our approach against ASR errors.

5. CONCLUSION

In this work, we investigate the use of CTC-based ASR models for utterance level SLU tasks. Experimental results show that joint training with CTC and SLU losses achieves SOTA results on several datasets. With pretrained acoustic encoders, a small fully connected layer-based utterance encoder can achieve very good performance. Our approach is also non-auto-regressive, thus efficient at inference time. For future work, we will investigate how to extend this framework for sequence labeling tasks, like slot filling. The key question is how to recover the entity names from the output of CTC-based ASR models in a differentiable and efficient manner without auto-regressive decoding.

6. REFERENCES

- [1] Y.-Y. Wang, L. Deng, and A. Acero, "Spoken language understanding," *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 16–31, 2005.
- [2] G. Tur and R. D. Mori, "Spoken language understanding: Systems for extracting semantic information from speech," *Wiley*, 2011.
- [3] A. Bhargava, A. Çelikyilmaz, D. Z. Hakkani-Tür, and R. Sarikaya, "Easy contextual intent prediction and slot detection," *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 8337–8341, 2013.
- [4] M. Larson, G. J. Jones *et al.*, "Spoken content retrieval: A survey of techniques and technologies," *Foundations and Trends® in Information Retrieval*, vol. 5, no. 4–5, pp. 235–422, 2012.
- [5] L. Lugosch, M. Ravanelli, P. Ignoto, V. S. Tomar, and Y. Bengio, "Speech model pre-training for end-to-end spoken language understanding," *arXiv preprint arXiv:1904.03670*, 2019.
- [6] D. Serdyuk, Y. Wang, C. Fuegen, A. Kumar, B. Liu, and Y. Bengio, "Towards end-to-end spoken language understanding," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5754–5758.
- [7] B. Agrawal, M. Müller, S. Choudhary, M. Radfar, A. Mouchtaris, R. McGowan, N. Susanj, and S. Kunzmann, "Tie your embeddings down: Cross-modal latent spaces for end-to-end spoken language understanding," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7157–7161.
- [8] M. Kim, G. Kim, S.-W. Lee, and J.-W. Ha, "St-bert: Cross-modal language model pre-training for end-to-end spoken language understanding," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7478–7482.
- [9] P. Haghani, A. Narayanan, M. Bacchiani, G. Chuang, N. Gaur, P. Moreno, R. Prabhavalkar, Z. Qu, and A. Waters, "From audio to semantics: Approaches to end-to-end spoken language understanding," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 720–726.
- [10] H.-K. J. Kuo, Z. Tüske, S. Thomas, Y. Huang, K. Audhkhasi, B. Kingsbury, G. Kurata, Z. Kons, R. Hoory, and L. Lastras, "End-to-end spoken language understanding without full transcripts," *arXiv preprint arXiv:2009.14386*, 2020.
- [11] M. Radfar, A. Mouchtaris, and S. Kunzmann, "End-to-end neural transformer based spoken language understanding," *arXiv preprint arXiv:2008.10984*, 2020.
- [12] M. Radfar, A. Mouchtaris, S. Kunzmann, and A. Rastrow, "Fans: Fusing asr and nlu for on-device slu," *arXiv preprint arXiv:2111.00400*, 2021.
- [13] M. Rao, A. Raju, P. Dheram, B. Bui, and A. Rastrow, "Speech to semantics: Improve asr and nlu jointly via all-neural interfaces," *arXiv preprint arXiv:2008.06173*, 2020.
- [14] 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.
- [15] S. Seo, D. Kwak, and B. Lee, "Integration of pre-trained networks with continuous token interface for end-to-end spoken language understanding," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7152–7156.
- [16] A. Raju, M. Rao, G. Tiwari, P. Dheram, B. Anderson, Z. Zhang, C. Lee, B. Bui, and A. Rastrow, "On joint training with interfaces for spoken language understanding," *Proc. Interspeech 2022*, pp. 1253–1257, 2022.
- [17] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.
- [18] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *ICASSP*. IEEE, 2016, pp. 4960–4964.
- [19] Y.-P. Chen, R. Price, and S. Bangalore, "Spoken language understanding without speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 6189–6193.
- [20] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 449–12 460, 2020.
- [21] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhota, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.
- [22] M. Henderson, B. Thomson, and J. D. Williams, "The second dialog state tracking challenge," in *Proceedings of the 15th annual meeting of the special interest group on discourse and dialogue (SIGDIAL)*, 2014, pp. 263–272.
- [23] K. Wei, D. Knox, M. Radfar, T. Tran, M. Müller, G. P. Strimel, N. Susanj, A. Mouchtaris, and M. Omologo, "A neural prosody encoder for end-to-end dialogue act classification," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7047–7051.
- [24] P. Warden, "Speech commands: A dataset for limited-vocabulary speech recognition," *arXiv preprint arXiv:1804.03209*, 2018.
- [25] E. Bastianelli, A. Vanzo, P. Swietojanski, and V. Rieser, "Slurp: A spoken language understanding resource package," *arXiv preprint arXiv:2011.13205*, 2020.
- [26] S. Arora, S. Dalmia, P. Denisov, X. Chang, Y. Ueda, Y. Peng, Y. Zhang, S. Kumar, K. Ganesan, B. Yan *et al.*, "Espnet-slu: Advancing spoken language understanding through espnet," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 7167–7171.
- [27] K. Chen, X. Du, B. Zhu, Z. Ma, T. Berg-Kirkpatrick, and S. Dubnov, "Hts-at: A hierarchical token-semantic audio transformer for sound classification and detection," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 646–650.
- [28] Y.-Y. Yang, M. Hira, Z. Ni, A. Chourdia, A. Astafurov, C. Chen, C.-F. Yeh, C. Fuhrsch, D. Pollack, D. Genzel, D. Greenberg, E. Z. Yang, J. Lian, J. Mahadeokar, J. Hwang, J. Chen, P. Goldsborough, P. Roy, S. Narenthiran, S. Watanabe, S. Chintala, V. Quenneville-Bélaïr, and Y. Shi, "Torchaudio: Building blocks for audio and speech processing," *arXiv preprint arXiv:2110.15018*, 2021.
- [29] 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)*, 2015, pp. 5206–5210.
- [30] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.