INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Speak or Chat with Me: End-to-End Spoken Language Understanding System with Flexible Inputs

Sujeong Cha[1*], *Wangrui Hou*[1*], *Hyun Jung*[1*], *My Phung*[1*], *Michael Picheny*[1], *Hong-Kwang Kuo*[2], *Samuel Thomas*[2], *Edmilson Morais*[3]

[1]New York University, USA
[2]IBM Research AI, USA
[3]IBM Research AI, Brazil

{sjc433, wh916, hj1399, mtp363, map22}@nyu.edu,
{hkuo, sthomas}@us.ibm.com, edmorais@br.ibm.com

Abstract

A major focus of recent research in spoken language understanding (SLU) has been on the end-to-end approach where a single model can predict intents directly from speech inputs without intermediate transcripts. However, this approach presents some challenges. First, since speech can be considered as personally identifiable information, in some cases only automatic speech recognition (ASR) transcripts are accessible. Second, intent-labeled speech data is scarce. To address the first challenge, we propose a novel system that can predict intents from flexible types of inputs: speech, ASR transcripts, or both. We demonstrate strong performance for either modality separately, and when both speech and ASR transcripts are available, through system combination, we achieve better results than using a single input modality. To address the second challenge, we leverage a semantically robust pre-trained BERT model and adopt a cross-modal system that co-trains text embeddings and acoustic embeddings in a shared latent space. We further enhance this system by utilizing an acoustic module pre-trained on LibriSpeech and domain-adapting the text module on our target datasets. Our experiments show significant advantages for these pre-training and fine-tuning strategies, resulting in a system that achieves competitive intent-classification performance on Snips SLU and Fluent Speech Commands datasets.

Index Terms: end-to-end spoken language understanding, BERT, automatic speech recognition, transfer learning

1. Introduction

Traditional spoken language understanding (SLU) systems use a cascaded automatic speech recognition (ASR) system to convert input speech signals into transcripts, followed by a natural language understanding (NLU) model to predict the intent from text transcripts [1]-[4]. While this approach has been widely adopted, there are significant limitations. The intermediate step of mapping audio to text is expensive to train and creates errors in the transcripts. Such errors greatly affect the final intent classification. Additionally, since these two models are trained independently, the primary metric of interest (intent classification accuracy) cannot be directly optimized. Due to this problem, end-to-end (E2E) SLU models that directly map a speech signal input to an SLU output have become popular [5]-[10].

However, this approach also presents some challenges. First, in some cases, only automatic speech recognition (ASR) transcripts are available since speech is considered as personally identifiable information and may not be stored. This implies that deployable E2E systems must still be able to handle text (ASR) input. Second, an E2E approach requires a large amount of data to produce comparable results to traditional cascaded ASR-NLU systems. This can be addressed by using pre-training to reduce the amount of training data required. For example, researchers have pre-trained models on large ASR datasets such as LibriSpeech [10][11] to relax audio data requirements, and have used pre-trained BERT networks [12]-[15] to relax text data requirements.

One way to address both challenges is described in [12]. A single architecture is used to jointly train an E2E system across a variety of tasks including ASR, SLU, masked LM prediction, and hypothesis prediction from text. Alternatively, authors in [13] successfully co-train a text-to-intent (T2I) model and speech-to-intent (S2I) model to closely align the acoustic embeddings with BERT-based text embeddings. Further improving on this cross-modal approach, authors in [14] employ the triplet loss function to learn a robust cross-modal latent space. Training the acoustic and text embeddings in the shared latent space makes it easier to combine separate audio and text data.

In this paper, we propose an architecture that extends the architectures in [13][14] because we believe that the common latent space approach with triplet loss is an easy but powerful way to utilize discriminative approaches in the training process. We also broaden the application scenarios of our system to perform intent classification with ASR transcripts to address the first challenge - when original audio files are not available due to privacy reasons. This also allows us to produce an intent prediction for a chat transcript without the audio counterpart. We choose the cross-modal system in [14] as our baseline model and further investigate and improve upon their approach. Their model consists of two branches: an acoustic branch that takes audio input, and a BERT text branch that takes the ground truth transcripts of the audio files. Triplet loss ties the acoustics embeddings and text embeddings together, thus forcing samples in the same class to be closer in the feature space and pushing samples from different classes to be further apart [14].

The novelties we introduce to the baseline model include 1) improvements to the acoustic and text branches by pre-training the acoustic branch and domain-adapting the BERT text branch, 2) the ability to take flexible types of input data (i.e. speech, ASR-transcripts/chat, or both), and 3) system combination [16][17] to combine intent predictions from acoustic and text branches, which performs better than an individual branch. We obtain significant improvements in performance over the baseline model on the Snips [18] and Fluent Speech Commands (FSC) datasets [10]. To welcome researchers to improve upon our work similar to [10][14], we are releasing our codebase.[1]

* equal contribution

[1]https://github.com/CoraJung/flexible-input-slu

http://dx.doi.org/10.21437/Interspeech.2021-788

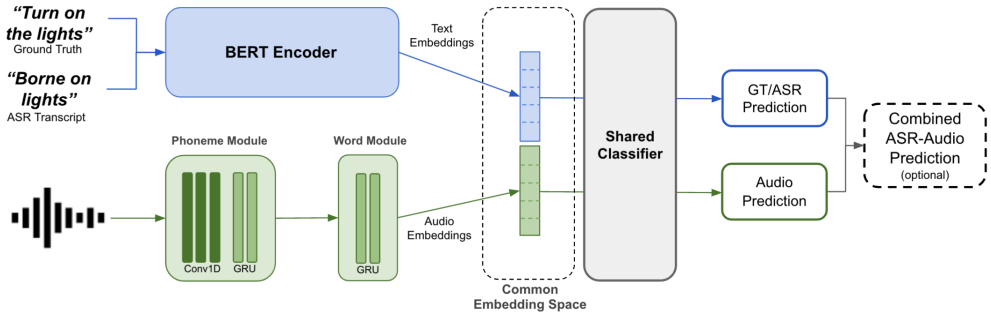

Figure 1: *Diagram of ASR-Text-Speech Model.*

2. Model

A major challenge when training an E2E SLU model is the scarcity of publicly available SLU datasets with paired utterance audio and semantic labels. ASR and NLU specific training data are much more accessible. Our approaches to overcome this challenge are 1) creating a tied cross-modal latent space that enables us to co-train the acoustic and text embeddings and 2) leveraging ASR and NLU data to pre-train the acoustic and text modalities independently before fine-tuning the joint model. Our cross-modal SLU system consists of two branches: an acoustic branch that takes audio input to produce acoustic embeddings and a text branch that takes in either ground truth or ASR transcripts of the audio files to produce text embeddings, as shown in Figure 1.

The acoustic branch is pre-trained on LibriSpeech, and the text branch is a pre-trained BERT model. The two embeddings are tied together in a common embedding space through a triplet loss. The two branches share the final classification layer that is trained to predict intent classes from acoustic embeddings, text embeddings, or both. Unlike the baseline model, our goal is to achieve higher audio accuracy while preserving the performance of the pre-trained text branch. The final input to the model training can include either 1) ground truth text transcripts and speech (Text-Speech Model) or 2) ASR transcripts, ground truth transcripts, and speech (ASR-Text-Speech Model). During inference time, the model can flexibly perform intent classification from any of these three types of input: 1) speech, 2) ASR/ground truth transcripts, and 3) speech combined with ASR/ground truth transcripts. In practice, an example of ground truth transcripts can be a chat utterance typed by a user.

2.1. BERT Text Branch

We follow the text branch architecture of the baseline model, which has a pre-trained BERT-base-cased model [19] as the text encoder. The text branch takes either ground truth or ASR-generated transcripts to create text embeddings. To boost the performance of our text branch, we also experiment with domain-adapting the text branch on our target datasets before co-training it with the acoustic branch.

2.2. LibriSpeech Pre-trained Audio Branch

The acoustic branch of the baseline model is a multi-layer Bi-LSTM network with random initialization that computes acoustic embeddings based on audio input. Because this acoustic branch functions similarly to an ASR model, it can greatly benefit from pre-training on an ASR-specific dataset, especially when E2E SLU training data is generally scarce [10]. Therefore, we directly utilize the LibriSpeech pre-trained phoneme and word modules from [10] for our acoustic branch. The phoneme module processes raw input waveform and outputs a hidden representation of the phonemes, and the word module takes the phoneme embeddings as input and produces word embeddings. Ultimately, the word embeddings output from the word module is passed through a linear layer to project the final acoustic embeddings into the same embedding space as the text embeddings.

2.3. Shared Embedding Space and Triplet Loss

The key component of the cross-modal architecture is the shared embedding space that allows the acoustic embeddings to "learn" from the text embeddings. This objective is achieved through a triplet loss function, denoted L_e:

$$L_e(x_a, x_+, x_-) = max(0, m + d(x_a, x_+) - d(x_a, x_-)) \quad (1)$$

where x_a represents an acoustic embedding of the current sample, x_+, x_- represents text embeddings of the positive and negative sample, m represents the margin, and d represents the squared distance. The benefit of this triplet loss is to move the acoustic embedding of the current example closer to the text embedding of the positive example and away from that of the negative example [14]. In brief, the triplet loss uses three inputs: one acoustic embedding of the current training example and two text embeddings coming from one negative and one positive example. A positive example is drawn randomly from the same intent class as the current example and a negative example is chosen arbitrarily from other intent classes in the training data. Different from the baseline model, our acoustic embedding of the current example will be the word embedding output from the word module in the acoustic branch. The text embeddings of positive and negative examples are outputs of the BERT encoder.

2.4. Model Selection Process

The final loss is a weighted sum of the triplet loss and cross-entropy losses from audio- and text-based predictions of the intents obtained from the last classification layer, which are denoted $L_{cls}^{acoustic}$ and L_{cls}^{text} respectively in the equation below:

$$L = L_{cls}^{acoustic} + \lambda_1 L_{cls}^{text} + \lambda_2 L_e \quad (2)$$

where λ_1, λ_2 are weights associated with the loss components. Originally, the baseline model chooses the best model based on the acoustic branch's validation accuracy only, which causes the text prediction accuracy to degrade. To preserve the performance of the text branch, we select our best model based on the

simple average of validation accuracy from both acoustic and text branches. This enables us to improve the performance of the text branch compared to the baseline model and ultimately allows the model to perform well on not only speech but also ASR-transcript inputs.

3. Experiments

3.1. Datasets

All of our experiments are performed on two publicly available datasets, namely Snips and FSC.

Snips SLU We use the "SmartLights" close-field subset of the Snips SLU dataset, which contains English spoken commands related to smart light appliances, such as "Can you turn on the light in the kitchen, please?", which are paired with intent labels, such as "TurnOnLight" [18]. It contains 1,660 unique utterances spoken by 21 speakers, mapping to six unique intent labels. To make our results comparable, we follow the data preparation procedures used in [14]. In particular, the dataset is split into training (80%), validation (10%), and test (10%) sets.

Fluent Speech Commands The FSC dataset contains 30,043 spoken English commands for a smart home or virtual assistant, such as "turn on the kitchen lights". It comes with full-text transcripts and intent labels in the form of "action", "object", "location". There are 248 unique transcripts mapping to 31 unique intents. We remove a total of 28 utterances in the training, validation, and test sets that have empty audio and obtain a final dataset of 30,015 utterances [10].

ASR Transcript Generation The Snips and FSC datasets only contain ground truth transcripts; no ASR transcripts are provided. We generate ASR transcripts by passing the audio inputs to an existing ASR model trained on LibriSpeech using the Kaldi Speech Recognition toolkit [20]. The average word-error-rate (WER) for generating ASR transcripts for Snips and FSC data are 34.1% and 25.6% respectively. Note that we use an off-the-shelf LibriSpeech system without any adaptation because we want to specifically examine the robustness of our system in the presence of relatively noisy ASR transcripts.

3.2. Training Setup

We first present the training setup for the Text-Speech model, trained on ground truth transcripts and speech. This model serves as the groundwork upon which we build our best models, ASR-Text-Speech-1 (without text encoder domain adaptation) and ASR-Text-Speech-2 (with such adaptation). Note that we perform hyper-parameter tuning for learning rates and set all other hyper-parameters the same as [10] and [14] to make the results comparable.

Text-Speech Model As explained in Section 2.1, a pre-trained BERT-base-cased model [19] with default configurations (12 layers, 768 final embedding dimension) acts as our text encoder. The acoustic branch has phoneme and word modules (three Conv1D layers; four GRU layers, 128 hidden units/layer) pre-trained on LibriSpeech, and one final linear layer (768 final embedding dimension). The two branches share the same final linear classifier and the output dimension varies according to the number of unique labels in training data. All pre-trained modules (BERT encoder, phoneme module, and word module) are then fine-tuned on our target dataset. The learning rate for both the acoustic encoder and the final classifier is 1e-3, and the learning rate for the BERT encoder is 2e-5.

ASR-Text-Speech-1 Model ASR-Text-Speech-1 builds upon the Text-Speech model. One major change is that the training data for the text encoder of ASR-Text-Speech-1 includes both ground truth and ASR transcripts, while Text-Speech is only trained on ground truth transcripts. Furthermore, our experiments have shown that, for ASR-Text-Speech-1, freezing the phoneme module of the acoustic branch produces better performance than fine-tuning all modules.

ASR-Text-Speech-2 Model Similar to the ASR-Text-Speech-1 model, ASR-Text-Speech-2 also improves upon Text-Speech by taking ASR transcripts as its input. However, ASR-Text-Speech-2 uses ASR transcripts in addition to ground truth transcripts to domain-adapt the BERT branch before joint training it with the acoustic branch. In this domain adaptation phase, we build a separate BERT model that has a structure identical to the text encoder in Text-Speech. We domain-adapt this BERT model on the ground truth and ASR transcripts from our target datasets with a learning rate of 2e-5. We then use this domain-adapted BERT model as our text branch without any further fine-tuning. By freezing the BERT text branch in the final training round, we let the domain-adapted text embeddings guide our audio embeddings. This is why, in this stage, we only use pairs of audio and ground truth transcripts, which provides better-quality text embeddings compared to ASR transcripts.

System Combination During inference, when both speech and ASR transcripts are available, we use system combination to leverage the "opinions" of the two branches, as shown in Figure 1, and achieve a better performance than using the prediction from each branch. This combined prediction is an element-wise average of the classifier's outputs (probabilistic predictions for each class label) from the acoustic branch and the text branch. Moreover, considering the flexible nature of our inputs, system combination helps our SLU system to consistently produce one single prediction. In summary, this system combination process ensures that 1) we achieve more accurate predictions and 2) our system can produce a single output when receiving both speech and ASR transcripts.

4. Results

4.1. Snips Results

We first apply our models to the Snips dataset. The **GT**, **Audio**, and **ASR** columns of Table 1 show the intent prediction accuracy when the models are tested on ground truth transcripts, audio input, and ASR transcripts, respectively. The **Combined** column shows the intent accuracy from system combination which combines the predictions of the acoustic and text branch to produce a final intent prediction. We also take the simple average of audio, ASR, and combined accuracy for each model to create the **Average** column. This column serves as an indicator of a model's overall performance, making comparisons easier and cleaner.

Based on the results, our three proposed models, which replace the unpre-trained LSTM-based acoustic branch in [14] with the GRU-based acoustic branch pre-trained on LibriSpeech in [10], outperform the baseline model on audio accuracy. Since the baseline model only generates predictions based on audio

Table 1: *Snips Results, Accuracy (%)*

Model	GT	Audio	ASR	Combined	Average
Agrawal et.al. [14]	-	69.88	-	-	-
Rongali et.al. [12]	-	84.88*	-	-	-
Text-Speech	98.19	75.90	78.31	87.95	80.72
ASR-Text-Speech-1	97.59	78.31	83.73	89.76	83.93
ASR-Text-Speech-2	97.59	80.12	80.72	86.75	82.53

*Hard to compare because run on different (unpublished) train and test sets.

input, the audio accuracy is the only category where we can compare our results with the baseline. To be comprehensive, we also included results from [12]. However, [12] utilizes a larger training set (including Snips far-field subset) and a different test set from ours, which makes direct comparisons difficult.

Among our three models, ASR-Text-Speech-1 has the highest average accuracy on Snips. Moreover, by comparing our two ASR-Text-Speech models with the Text-Speech model, we observe that adding ASR transcripts to our training data improves our model's performances on audio, ASR, and combined accuracy. There are only trivial differences in the results of our two ASR-Text-Speech models, which implies that applying domain adaptation to our text branch does not provide notable benefits to our model. Compared to ASR-Text-Speech-1, ASR-Text-Speech-2 performs better on audio input but worse on ASR transcripts. This result confirms our hypothesis in 3.2 that fixing the text branch after domain-adaptation guides audio embeddings better; however, this comes with a trade-off in text predictions because the text branch is not fine-tuned in the final training stage. In summary, we would recommend the ASR-Text-Speech-1 based on its competitive performance and simplicity in training setup.

As expected, system combination (the **Combined** column) generally achieves higher accuracy than using acoustic or ASR transcript information alone for prediction.

4.2. FSC Results

Table 2: *FSC Results, Accuracy (%)*

Model	GT	Audio	ASR	Combined	Average
Agrawal et.al. [14]	-	95.65	-	-	-
Lugosch et.al. [10]	-	98.75	-	-	-
Rongali et.al. [12]	-	99.50	-	-	-
Text-Speech	100.00	99.13	87.92	97.36	94.80
ASR-Text-Speech-1	100.00	99.18	98.18	99.45	98.94
ASR-Text-Speech-2	100.00	98.95	98.23	99.53	98.90

Next, we test the efficacy of our proposed models on the FSC dataset (Table 2). Similar to the results on Snips, our models perform well in all four categories and ASR-Text-Speech-1 is still our best performing model. Additionally, our models give competitive results, outperforming results reported in [10] and [14]. Our models perform slightly worse than the model proposed in [12] since their model is pre-trained with additional SLU datasets such as their internal datasets.

Also notice that we observe a larger improvement in ASR accuracy from Text-Speech model to ASR-Text-Speech models in FSC (around 10%) than in Snips (around 2-5%). We hypothesize that, because ASR transcripts of FSC has a lower WER than those of Snips, the addition of better-quality ASR transcripts benefits FSC more during training.

4.3. Additional Results

Table 3: *Snips Text Branch Results, Accuracy (%)*

Model	GT	ASR
Text-Only	95.18	77.11
ASR-Text	95.78	78.92
ASR-Text-Speech-1	97.59	83.73

We also examine the impact of varying the input type on the performance of our text branch using Snips (Table 3) and FSC. The **Text-Only** model is a pre-trained BERT-base-cased model fine-tuned on only the ground truth transcripts, and the **ASR-Text** model is the same BERT model fine-tuned on both the ground truth and the ASR transcripts. By comparing the Text-Only model with the ASR-Text model, we further provide evidence that adding ASR transcripts in the training stage improves model performance on both ground truth and ASR test sets. Interestingly, adding the acoustic branch to the ASR-Text model further improves the performance of our text branch when tested on either ground truth or ASR transcripts. Similar results are observed when we run these experiments on FSC.

5. Conclusion

In general, we can clearly see the improvements achieved by the novelties proposed in this paper.

First, our ASR-Text-Speech models can predict intents from flexible types of inputs: speech, ASR transcripts (or chat utterances), or a combination of both. Our model gives excellent performance when only the ASR transcripts are available to predict intents. If both speech and ASR transcripts are available at test time, combining the intent probability predictions of the acoustic and text branches gives more accurate final predictions. Between our two versions of our proposed model, ASR-Text-Speech-1 has better overall predictions and is also more compact and efficient compared to ASR-Text-Speech-2 due to the absence of domain-adapting BERT on target datasets in the pre-training round.

Second, pre-training on LibriSpeech gives our system a more powerful acoustic branch, enabling our model to outperform the baseline. Furthermore, by saving the best model based on both audio- and text-based prediction accuracy, our model preserves the performance of the text branch compared to the baseline model. It is important to note that co-training the text branch and the acoustic branch does not only improve the acoustic branch's performance, which is the original motivation, but also boosts the performance of the text branch. We hypothesize that since the shared classifier between the two modalities is trained to perform predictions on either text or acoustic embeddings, it may be better regularized and more robust to unseen or noisy data at test time. This also explains why adding noisy ASR transcripts improves the results of our Text-Only model.

6. Acknowledgments

We would like to thank Markus Mueller from Amazon Alexa Machine Learning and Loren Lugosch from McGill University and Mila AI Research Institute for releasing their codebases and answering our questions. We would also like to thank Amber Wang from New York University for generating the ASR transcripts for Snips SLU and FSC.

7. References

[1] C. Lee, S. Jung, K. Kim, D. Lee, and G. G. Lee, "Recent approaches to dialog management for spoken dialog systems," *Journal of Computing Science and Engineering*, vol. 4, no. 1, pp. 1–22, 2010.

[2] V. Goel, H.-K. J. Kuo, S. Deligne, and C. Wu, "Language model estimation for optimizing end-to-end performance of a natural language call routing system," in *Proc. ICASSP*, vol. 1, 2005, pp. 565–568.

[3] S. Yaman, L. Deng, D. Yu, Y.-Y. Wang, and A. Acero, "An integrative and discriminative technique for spoken utterance classification," in *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 16, no. 6, pp. 1207–1214, 2008.

[4] 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.

[5] M. Radfar, A. Mouchtaris, and S. Kunzmann, "End-to-end neural transformer based spoken language understanding," in *Interspeech 2020, Annual Conference of the International Speech Communication Association*, 2020, pp. 500–505.

[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] S. Ghannay, A. Caubrière, Y. Estève, N. Camelin, E. Simonnet, A. Laurent, and E. Morin, "End-to-end named entity and semantic concept extraction from speech," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 692-699.

[8] 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.

[9] Y. Qian, R. Ubale, V. Ramanaryanan, P. Lange, D. Suendermann-Oeft, K. Evanini, and E. tsuprun, "Exploring ASR-free end-to-end modeling to improve spoken language understanding in a cloud-based dialog system," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 569–576.

[10] 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.

[11] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "LibriSpeech: An ASR corpus based on public domain audio books," *ICASSP*, pp. 5206-5210, 2015.

[12] S. Rongali, B. Liu, L. Cai, K. Arkoudas, C. Su, and W. Hamza, "Exploring Transfer Learning For End-to-End Spoken Language Understanding," *arXiv preprint arXiv:2012.08549*, 2020

[13] Y. Huang, H.-K. Kuo, S. Thomas, Z. Kons, K. Audhkhasi, B. Kingsbury, R. Hoory, and M. Picheny, "Leveraging unpaired text data for training end-to-end speech-to-intent systems," in *Proc. ICASSP*, 2020, pp. 7984–7988.

[14] B. Agrawal, M. Muller, M. Radfar, S. Choudhary, A. Mouchtaris, and S. Kunzmann, "Tie your embeddings down: Cross-modal latent spaces for end-to-end spoken language understanding," *arXiv preprint arxiv:2011.09044*, 2020.

[15] 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.

[16] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On Combining Classifiers," *IEEE Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226-239, 1998.

[17] R. Price, M. Mehrabani and S. Bangalore, "Improved End-To-End Spoken Utterance Classification with a Self-Attention Acoustic Classifier," *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 8504-8508.

[18] A. Saade, A. Coucke, A. Caulier, J. Dureau, A. Ball, T. Bluche, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril, and M. Primet, "Spoken language understanding on the edge," *arXiv preprint arXiv:1810.12735*, 2018

[19] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al., "Huggingface's transformers: State-of-the-art natural language processing," *arXiv preprint arXiv:1910.03771*, 2019.

[20] 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," *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, IEEE, 2011.