2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU) | 979-8-3503-0689-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ASRU57964.2023.10389786

WHISPER-SLU: EXTENDING A PRETRAINED SPEECH-TO-TEXT TRANSFORMER FOR LOW RESOURCE SPOKEN LANGUAGE UNDERSTANDING

Quentin Meeus[1,2], *Marie-Francine Moens*[1], *Hugo Van hamme*[2]

[1]Dpt. Computer Science [2]Dpt. Electrical Engineering KU Leuven, Belgium.

ABSTRACT

Human-computer interactions require systems that work out of the box without requiring lots of data to adapt to a new task or user. In this research, we address low resource spoken language understanding tasks such as named entity recognition (NER), intent recognition (IR), and slot filling (SF) to research how a pretrained model can be modified for a new task, then finetuned with few labelled data. We propose extending the Whisper model with task-specific modules for NER, SF, and IR, leveraging a Markov network as output structure. We develop a novel approach to finetuning by removing irrelevant weights and reorganizing the embeddings to drastically improve the performance in a low-resource setting. Our approach outperforms previous models without external language models and demonstrates effective transfer learning, even with very limited training data. The models exhibit a small footprint, making them suitable for applications requiring robustness, few-shot learning, and efficiency.

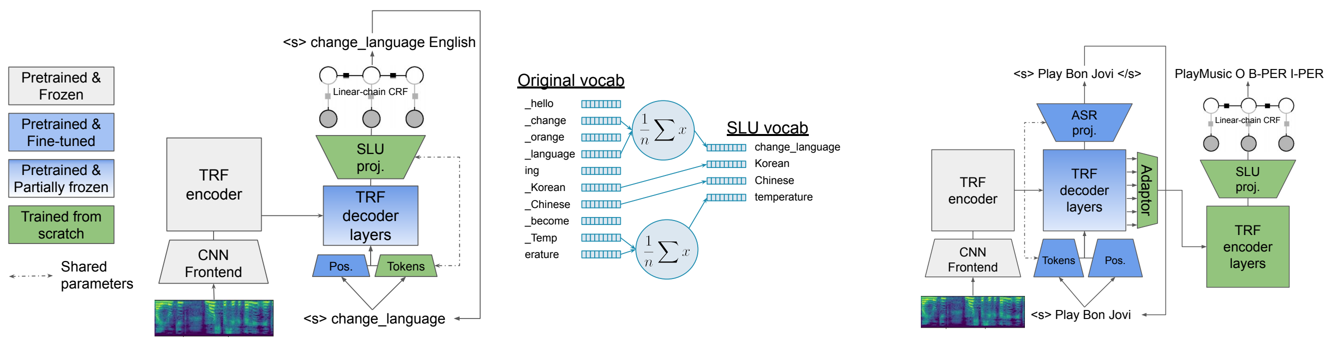
Index Terms— low resource spoken language understanding, named entity recognition, multitask learning

1. INTRODUCTION

Since their introduction in 2017, transformers [1] have become a cornerstone of deep learning, achieving impressive results across a wide range of fields [2, 3, 4]. Research has shown that scaling up the number of parameters and the size of the training set leads to remarkable performance increase [5, 6]. For instance, OpenAI's speech-to-text model, Whisper [6] is a large transformer trained on considerable amounts of filtered data. It improved the state-of-the-art in multilingual speech recognition and speech translation to English. Unfortunately, transformers are notoriously data-hungry and training such models requires significant resources. The good news is that once a model has been trained, it can be finetuned for other tasks. In natural language processing, there are numerous examples of fine-tuned transformers performing well on tasks that differ considerably from what they were trained on [7]. However, the transferability of speech-to-text transformer models to other speech processing tasks has not been extensively explored yet, even more so in spoken language understanding where data is scarce.

Nonetheless, it is often useful to reuse a pretrained model for other tasks for which it was not trained. For example, in commercial applications, the manufacturer wants to avoid as much as possible to adapt the model to specific users and use cases. Data is often expensive to generate, and having to adapt to each customer results in an unpleasant user experience. In this research, we explore low resource spoken language understanding with three specific tasks: named entity recognition (NER), intent recognition (IR) and slot filling (SF). NER aims at identifying and extracting so-called named entities, such as places, person names, dates, or events from a spoken utterance. IR is the identification of a user's intent, in the context of a man-to-machine interaction. SF could be considered as being at the crossroad between IR and NER. The task is to identify the intent together with any number of arguments (slots) relevant to answer a query. In the context of airplane booking, these might be the departure and destination, date and time, etc. Although IR can also have arguments, they can take a limited number of predefined values. The main difference with SF is the large number of values that the arguments typically take. These tasks are particularly important in the development of conversational interfaces, such as those used for controlling robots, where accurate interpretation of spoken language input is crucial. For these systems to be useful, they must meet several key requirements: robustness to speaker and accent variations, ability to learn from few examples, and a small footprint to work on small appliances without sacrificing accuracy or performance.

NER has been widely studied in NLP, where the main approach is to use the BIO framework as output structure [8]. Each language token in a sentence (often single words) must be assigned a tag composed of a flag indicating whether the token consists of the beginning, continuation or the absence of an entity, along with the type of entity (place, person, event, number, etc.). In the context of Spoken NER, very little research exists on this topic. We can mention the SLUE benchmark, which proposed a new dataset to tackle this task [9]. Using this dataset, [10] takes a sequence-to-sequence approach, where special symbols are issued to mark the beginning and the end of an entity phrase. They use a Conformer [4] encoder and a Transformer decoder [1]. Finally, [11] proposed to replace the last layers of a pretrained ASR model with a text-based NER model. Unfortunately, they use a non



(a) IR task

(b) NER and SF tasks

Fig. 1. (a) For intent recognition, the transformer decoder is adapted for the task-specific vocabulary by replacing the embeddings and performing selective finetuning. The embeddings are constructed by using the original token embeddings when available or taking the average of multiple embeddings. (b) A transformer encoder-decoder transcribes speech and generates hidden representations. The NER module predicts entity types from the decoder hidden states following the BIO framework. For SF, we predict an additional token corresponding the intent label class.

freely available dataset and did not release the source code, making any comparison difficult.

By contrast, intent recognition and slot filling have been more extensively studied in the context of spoken language understanding. Pipeline approaches use an ASR model to produce a transcription that is processed by a NLU module, often composed of a pretrained language model with a specific prediction head [12, 10]. Although these systems suffer from error propagation, NER pipelines perform better than end-to-end approaches that directly map the speech to semantics without forcing a transcription. [13] propose an end-to-end model in which three modules are stacked together: a phoneme module, a word module and an intent module. Each module can be pretrained independently by following an unfreezing schedule. The intent recognition task is reformulated as a multilabel classification task. [14] define the task as a sequence-to-sequence problem with an encoder-decoder architecture and introduces a multitask learning approach where the model learns both the transcript and the semantics. Unlike the present work, they opt for sharing the encoder and use a separate decoder for each task. Finally, multi-modal approaches that use both text and speech encoders have been proposed. ST-BERT [15] enhance their model by pretraining it with cross-modal language modelling on both speech and text data using two different approaches: masked language modelling (MLM) and conditioned language modelling (CLM). In CLM, the objective is to predict one modality based on the other. To enhance the pretraining process, ST-BERT uses large text corpora and domain adaptation pretraining with SLU transcripts. [16] implement cross-modal contrastive learning to learn multi-modal representations from text and speech in parallel. They use a Conformer speech encoder and BERT text encoder and use contrastive learning to align the resulting embeddings in the same latent space. Multitask learning [17] and multimodal approaches [15, 16] allow the model to learn shared representations from both modalities, which can be useful in a low resource scenario. In almost all mentioned research, pretrained models are shown to improve the performance considerably [12, 10, 15, 16, 18].

Recently, Whisper [6], a pretrained encoder-decoder transformer that was trained on large amounts of curated data. has been open-sourced. In this paper, we propose to extend this model with dedicated modules for each considered task. Although other candidate models exist, like Wav2Vec2 [19] and HuBERT [20], those are encoder-only models that we think mostly encode acoustic information rather than linguistic information, as it is the case for Whisper's decoder. For NER, we propose to add a transformer encoder that processes and tags the hidden states produced by Whisper's decoder. We take a similar approach for SF, with the difference that an additional token is predicted for the intent label, before predicting the slots. Finally, for IR, we replace Whisper's embedding and carefully select which parameters to finetune and which remain frozen. For all tasks, we enforce the specific output structure with a Markov network. We also use test sets with withdrawn speakers and unseen phrasing to evaluate the ability of our models to generalize on challenging data. Our main contributions are as follows: We propose an end-to-end model for low resource Spoken NER and SF that outperforms previous models without requiring an external language model. We also show how a transformer decoder can be modified for a new task by replacing the token embeddings and gradually unfreezing the parameters. This method allows transferring and completely adapting a model to a new task, even when we dispose of only a few training samples. The small footprint for both finetuning and inference makes it attractive for many applications.

2. METHODOLOGY

The models presented in this work build upon the state-of-the-art Whisper model. We are exploring three tasks, defined in the previous section: NER, SF and IR. In this section, we provide for each of them a brief description of the task, the specific architecture and the objective function.

2.1. Shared architecture

All three proposed models share a speech encoder in the form of a transformer encoder with convolutional frontend, that takes mel-spectogram inputs and outputs speech representations. The subsequent modules are specific to the SLU tasks and are described below. All three tasks have a very specific output structure that we enforce during inference with a Markov network. In low-resource scenarios, the amount of training data is insufficient to learn the transition probabilities of a dynamic CRF [21]. Therefore, we set the transition probability from i to j as $p_{ij} = 1/n_i$, where n_i is the number of legal transitions from i. Illegal transitions receive a null probability. We use Viterbi algorithm [22] to decode the prediction given the SLU output (Fig. 1).

2.2. Intent Recognition

For this task, the number of intents / arguments combination is relatively small, and the structure is well-defined. The applications of intent recognition are often specific to one use case, such as a smart home device or an automated guided vehicle. These can only perform a predefined set of actions. Moreover, data gathering is expensive, and we want to train our model with as few examples as possible. Consequently, we chose a different approach for this model. We define a vocabulary as the set of possible actions (intent) together with the set of all possible argument values for each argument. For example, if a robot can only turn left or right and move forward or backward, the vocabulary size is 8 (2 intents, 4 arguments, start- and end-of-sequence tokens). We then reformulate the task as a sequence prediction problem. In the spirit of recycling as much as we can from the pretrained decoder, we keep all layers but the token embeddings, which we initialize by selecting relevant vectors in the original embeddings (Fig. 1 (left and center)). We also experiment with random initial embeddings that we train with a large learning rate while keeping the rest of the parameters. We then partially unfreeze the pretrained layers and finetune the model with a low learning rate to allow the layers to adapt to the new task. The objective of this model is to minimize the cross-entropy loss between the predicted tokens and the true intent sequence. The evaluation metric is the accuracy, where an example is considered correct when both the intent and all slots are predicted correctly.

2.3. Named Entity Recognition

For NER, the model predicts two outputs: the verbatim transcription of the speech and a sequence of the same length containing the BIO entity tags. For the transcription task, we keep Whisper's decoder as is. An adaptor transforms the decoder's hidden states, which are then processed and tagged by a transformer encoder (Fig. 1 (right)). The adaptor has two roles: it learns to weigh the contributions of each decoder layer and sums them together, and it maps Whisper's embedding dimension to the module dimension when these numbers differ. We adopt a multitask learning approach $\mathcal{L}_{\text{NER}} = \alpha\mathcal{L}_{\text{token}} + (1-\alpha)\mathcal{L}_{\text{tag}}$ where the model loss, $\mathcal{L}_{\text{NER}}$, is a linear combination of the cross-entropy loss relative to the transcription, $\mathcal{L}_{\text{token}}$, and the focal loss relative to the named entity tags, $\mathcal{L}_{\text{tag}} = (1-p)^f \cdot \text{CE}(\hat{y}, y^*)$, where CE is the cross-entropy). The focal loss encourages the model to focus on difficult examples, which is useful when the labels are unevenly distributed. The metric used to evaluate the transcriptions is the word error rate (WER). For the entities, we use the unordered micro-averaged F1-score and the label-F1. The former takes into account both the tag and the entity transcription, while the latter only takes the tag into account. The spread between both scores gives an indication of the ability of the model to spell the entities correctly.

2.4. Slot Filling

As mentioned above, SF could be seen as a more difficult case of IR, where the number of slots and the possible slot values are not always known in advance. In this regard, SF is very similar to predicting entities together with an intent class. Therefore, it makes sense to keep our NER architecture. To predict the intent, we simply insert a token corresponding to the intent class in the first position of the target sequence. The loss is the same as the NER task. The evaluation metric is the accuracy, where an example is considered correct when both the intent and all slots are predicted correctly.

3. EXPERIMENTAL SETUP

3.1. Datasets

SLUE-VoxPopuli [9] is a subset of the VoxPopuli dataset [23] from the European Parliament sessions, together with named entities. The SLUE benchmark [9] provides limited training (5,000 ex, 14 h) and development sets (1,753 ex, 5h) and a blind test set (1,842 ex, 5h).
Smartlights [24] contains instructions to control smart lights. One can turn them on and off, set the brightness, or change the color. Slots include the rooms of the house, color and brightness (a number from 0 to 100). In its original form, the dataset (1,660 ex, 1h13) does not contain splits. Following [25], we use random splits (90/10/10).
SL Challenge [24, 26]: New splits of the Smartlights dataset

have been proposed, with two challenging test sets. One test set (Spk.) is composed of recordings from unseen speakers and the other (Utt.) contains utterances for which the wording was not observed for a particular target. The dataset has 1,251 training examples (1h), 156 validation examples (7 min), and each test set has 126 examples (6min).

Fluent Speech Commands [13] is a widely used dataset with spoken commands for a virtual assistant. Each intent can have up to 2 arguments, with a total of 31 commands. In its original form, the train, validation a test splits have respectively 23k, 3k 3.8k examples (14h40, 2h, 2h35).

FSC Challenge [13, 26]: The challenge splits contain 19,262 training examples (12h), 2,597 validation examples (1h40), 3,349 test examples with unseen speakers (2h12) and 4,204 examples with unknown utterances (2h40).

3.2. Low Resource Scenario

To simulate a low resource scenario, we sample from the original training set a predefined number of examples corresponding to 1% or 10% of the original dataset size. We repeat the operation 10 times with new random seeds to get 10 different training set. We do the same on the validation set. Once trained, the model is evaluated on the full test set.

Dataset	Fluent Speech Commands			SmartLights
Train size	1%	10%	100%	100%
Whisper-SLU	**97.19**	**99.46**	99.66	**95.35**
Lugosch et al. [13]	82.78	97.96	98.80	-
CMCL [16]	96.89	99.37	**99.69**	92.5

Table 1. Average accuracy scores on Fluent Speech Command (original splits) and SmartLights (random splits). Standard deviation for Whisper-SLU on FSC 1%, 10% and 100% are respectively 0.31, 0.05 and 0.03 and 1.28 on SmartLights

3.3. Implementation details

The models were implemented in PyTorch [27] with the Transformers library [7]. The pretrained model is whisper-large-v2 for NER and whisper-small for SF and IR. We use different models because we want to keep the IR and SF model sizes reasonable, as those models will typically be used on small appliances. The task modules for NER and SF have 2 layers and 12 attention heads with each 64 hidden units for a total dimension of 768. Since whisper-large-v2 has a hidden dimension of 1,280, we apply a linear transformation to get the desired 768 dimension, after having applied the convex linear combination of the hidden representations of the decoder (adaptor in Fig. 1). The weights for the SLU and ASR losses are respectively 0.1 and 0.9. We set the focus of the focal loss to 1. We train the models with AdamW [28] and a learning rate that peaks at 4e-6 for NER and 1e-4 for SF and IR, with a cosine schedule with warmup corresponding to 10% of the total number of steps. The large model was trained with an effective batch size of 128 for 3,000 steps, or about 40 epochs. Following [29], after training the models, we average the weights of the last 5 checkpoints. When training in the low-resource settings, we sample either 1% or 10% of the training set with a different seed and train the model with the resulting datasets, and we measure the performance on the test set. We report the average performance along with the standard deviation obtained over 10 runs. The number of steps for 1% and 10% subsets was 500 and 1000 for the full SF and IR datasets. The batch size, learning rate and dropout were chosen experimentally by measuring the performance on the validation set.

4. EXPERIMENTAL RESULTS

	F1-score (↑)	label-F1 (↑)	WER (↓)
QUANT	73.3	87.3	-
PLACE	91.3	94.1	-
WHEN	63.0	83.7	-
ORG	70.7	82.4	-
LAW	30.3	65.5	-
NORP	84.2	90.9	-
PERSON	47.6	89.5	-
overall_micro	76.3	88.1	-
overall_macro	65.8	84.8	7.3
Oracle text [9]	87.5	91.1	0.0
Pipeline w/ LM [9]	74.9	87.4	9.1
E2E w/ LM [9]	70.2	79.0	9.1
SSL pretraining [10]	74.5	88.0	9.3
SLU pretraining [10]	75.7	87.5	9.0
Ensemble [10]	77.2	88.7	8.6

Table 2. Named Entity Recognition errors breakdown (dev).

In this section, we present the results of our experiments on the three considered tasks using the proposed models. We compare our models to several baselines defined in Section 1.

4.1. Intent Recognition

We explore the robustness of our models to small training sets. We perform this analysis on the challenge splits in order to evaluate the robustness of our models to unknown speakers and utterances in these data scarcity scenarios. Since the training set has 19,262 examples, we train our 1% and 10% models on 192 and 1926 examples. Since these splits have few published results, we perform the same analysis on the original splits and compare our findings with two other papers, described in Section 1. We sample 10x from the training set and report the average score and standard deviation on the test sets (Table 1 and 3).

To finetune the models, we first train the embeddings for 300 steps with a learning rate that reaches 5e-3. The fully-connected layers and layer norm are then unfrozen and finetuned at a lower learning rate of 1e-4 for an additional 100 steps. We observe that if this unfreezing schedule is not followed, the performance drops drastically (from

Test set	FSC Challenge (Spk.)			FSC Challenge (Utt.)			SL Challenge	
Train size	1%	10%	100%	1%	10%	100%	Spk.	Utt.
Whisper-SLU	93.0 (1.38)	96.1 (0.26)	98.3 (0.06)	88.7 (1.49)	91.6 (2.09)	93.3 (1.19)	90.6	81.1
Lugosch et al. [13]	-	-	92.3	-	-	78.3	82.6	78.5
ESPnet-SLU [18]	-	-	97.5	-	-	78.5	-	-
MTL [17]	-	-	98.2	-	-	88.1	-	-

Table 3. Average accuracy and standard deviation (%) on FSC Challenge and SL Challenge test sets.

$94.79 \pm 1.56\%$ to $37.9 \pm 18.5\%$ on the 1% dev sets). This is because when training starts, the embeddings are still random while the other parameters are pretrained, leading to the first few updates being counterproductive. We compare the resulting embeddings with Whisper's and discover that for each token vector, the closest embedding is a token from the original vocabulary that is semantically close (e.g. down / decrease, language / change_language, more / increase, etc.). This motivates us to build the initial embeddings from the pretrained model by selecting tokens with a similar meaning and extracting their representations.
After training the embeddings, we finetune the fully-connected layers from the transformer decoder with a low learning rate. Although the performance was already high from only training the embeddings ($94.91 \pm 1.20\%$ on the 1% dev sets), this additional step gets us to $96.18 \pm 1.67\%$. We also try to unfreeze the whole decoder and finetune the self- and cross-attention layers, but the performance does not increase any further. These experiments are conducted on the 1% datasets. Similar results are observed on the larger datasets, although with a smaller effect as more training data is available.

4.2. Named Entity Recognition

Our proposed model achieves an F1-score of 76.3% and a label-F1 of 88.0% on the development set, outperforming the end-to-end and pipeline baselines of the SLUE benchmark and all approaches from [10] except for the ensemble of their four best models. The WER on the development set is 7.3%, or 1.3 absolute points lower than the ensemble in [10], 1.8 lower than the best baseline in [9]. The organizer of the challenge communicated us a WER of 8.8% and a F1-score of 70.1% on the test set. Our breakdown analysis (on the dev set, as the test set is blind) shows that most errors are due to transcription errors. This is shown by the difference between F1-score and label-F1 in Table 2. In other words, the model is able to correctly identify entities, although it sometimes spells them incorrectly. This is especially the case for names, laws, and events.

4.3. Slot Filling

SLChallenge [26] only contains 1 hour of training data. The evaluation is done on two test sets: speaker and utterance. The former contains utterances by speakers that are not present in the training set, to evaluate the model's robustness to new speakers. The latter contains utterances whose choice of word and vocabulary is different from the training set, to evaluate how the model generalizes to unseen sentences. Only the original paper reports results on these splits, by reevaluating the model proposed in [13] to get respectively 82.6% and 78.5%. Our method gives an accuracy of 90.55% and 81.1% respectively (Table 3). This shows that our model generalizes well to unknown utterances and is robust to new speakers. We also present results on random splits, averaged over 10 runs (Table 1). As in the previous section, a breakdown of the errors shows that the intent classification and the entity identification are mostly correct, and entity spelling mistakes are almost 3x more frequent.

5. DISCUSSION AND CONCLUSIONS

In this article, we explored the transferability of pretrained speech-to-text transformer models for the tasks of intent recognition, named entity recognition, and slot filling with a particular focus on low-resource datasets. For the first task, we proposed a new approach to finetuning by modifying a pretrained model's embeddings to fit a new vocabulary. We observed that, when done correctly, we can reach impressive performance by updating only a handful of parameters. We showed that there is much to gain by carefully selecting and building the embeddings from the original model rather than random initialization. For the other tasks, we added dedicated modules to the pretrained model. In all tasks, we proposed a Markov network that does not require training to enforce the task-specific output structure. The proposed models outperformed several baselines in all three tasks, using varying amounts of training data.
Overall, our study contributes to the development of robust and efficient conversational interfaces by addressing key challenges in SLU, such as working with very few training data, robustness to accents and speakers, and small footprint for training and inference. In future work, we will continue exploring this topic through other language understanding tasks and different languages.

6. ACKNOWLEDGEMENTS

This research received funding from the Flemish Government under the "Onderzoeksprogramma Artificiele Intelligentie (AI) Vlaanderen" programme.

7. REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar *et al.*, "Attention is all you need," in *NeurIPS*, 2017.

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov *et al.*, "An image is worth 16x16 words: Transformers for image at scale," in *ICLR*, 2021.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*. ACL, Jun. 2019.

[4] A. Gulati, J. Qin, C.-C. Chiu *et al.*, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Interspeech*, 2020.

[5] T. Brown, B. Mann, N. Ryder *et al.*, "Language models are few-shot learners," in *NeurIPS*, 2020.

[6] A. Radford, J. W. Kim, T. Xu *et al.*, "Robust speech recognition via large-scale weak supervision," *CoRR*, 2022.

[7] T. Wolf, L. Debut, V. Sanh *et al.*, "Transformers: State-of-the-art natural language processing," in *EMNLP*. ACL, 2020.

[8] L. Ramshaw and M. Marcus, "Text chunking using transformation-based learning," in *Third Workshop on Very Large Corpora*, 1995.

[9] S. Shon, A. Pasad, F. Wu *et al.*, "SLUE: new benchmark tasks for spoken language understanding evaluation on natural speech," *CoRR*, 2021.

[10] Y. Peng, S. Arora, Y. Higuchi *et al.*, "A study on the integration of pre-trained ssl, asr, lm and slu models for spoken language understanding," in *SLT*. IEEE, 2022.

[11] S. Mdhaffar, J. Duret, T. Parcollet, and Y. Estève, "End-to-end model for named entity recognition from speech without paired training data," in *Interspeech*, 2022.

[12] C. Lee, S. Jung, K. Kim *et al.*, "Recent approaches to dialog management for spoken dialog systems," *JCSE*, vol. 4, 2010.

[13] L. Lugosch, M. Ravanelli, P. Ignoto *et al.*, "Speech Model Pre-Training for End-to-End Spoken Language Understanding," in *Interspeech*, 2019.

[14] P. Haghani, A. Narayanan, M. Bacchiani *et al.*, "From audio to semantics: Approaches to end-to-end spoken language understanding," in *SLT*. IEEE, 2018.

[15] 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.

[16] J. Dong, J. Fu, P. Zhou *et al.*, "Improving spoken language understanding with cross-modal contrastive learning," in *Interspeech*, 2022.

[17] Q. Meeus, M.-F. Moens, and H. Van hamme, "Multitask Learning for Low Resource Spoken Language Understanding," in *Interspeech*, 2022.

[18] S. Arora, S. Dalmia, P. Denisov *et al.*, "ESPnet-SLU: Advancing spoken language understanding through espnet," in *ICASSP*. IEEE, 2022.

[19] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *NeurIPS*. Curran Associates Inc., 2020.

[20] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai *et al.*, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," 2021.

[21] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML. Morgan Kaufmann Publishers Inc., 2001.

[22] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, 1967.

[23] C. Wang, M. Riviere, A. Lee *et al.*, "VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation," in *IJCNLP*. ACL, 2021.

[24] A. Saade, A. Coucke, A. Caulier *et al.*, "Spoken language understanding on the edge," *CoRR*, 2018.

[25] B. Agrawal, M. Müller, S. Choudhary *et al.*, "Tie your embeddings down: Cross-modal latent spaces for end-to-end spoken language understanding," in *ICASSP*, 2022.

[26] S. Arora, A. Ostapenko, V. Viswanathan *et al.*, "Rethinking end-to-end evaluation of decomposable tasks: A case study on spoken language understanding," in *Interspeech*, 2021.

[27] A. Paszke, S. Gross, F. Massa *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *NeurIPS*. Curran Associates, Inc., 2019.

[28] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *ICLR*, 2019.

[29] S. Watanabe, T. Hori, S. Karita *et al.*, "ESPnet: End-to-End Speech Processing Toolkit," in *Interspeech*, 2018.