



TOKENVERSE: UNIFYING SPEECH AND NLP TASKS VIA TRANSDUCER-BASED ASR

Shashi Kumar Srikanth Madikeri
Zuluaga-Gomez. Juan Nigmatulina Iuliia
Esaú VILLATORO-TELLO^a Sergio Burdisso
Petr Motliceck Karthik Pandia D S
Aravind Ganapathiraju

Idiap-RR-07-2024

AUGUST 2024

^aIdiap

TokenVerse: Unifying Speech and NLP Tasks via Transducer-based ASR

Shashi Kumar^{1,4}, Srikanth Madikeri¹, Juan Zuluaga-Gomez^{1,4}, Iuliia Nigmatulina^{1,2}, Esaú Villatoro-Tello¹, Sergio Burdisso¹, Petr Motlicek^{1,3}, Karthik Pandia⁵, Aravind Ganapathiraju⁵

¹Idiap Research Institute, Switzerland ²University of Zurich, Switzerland

³ Faculty of Information Technology, Brno University of Technology, Czech Republic

⁴LIDIAP, Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland

⁵Uniphore, India

Abstract

In traditional conversational intelligence from speech, a cascaded pipeline is used, involving tasks such as voice activity detection, diarization, transcription, and subsequent processing with different NLP models for tasks like semantic endpointing and named entity recognition (NER). Our paper introduces TokenVerse, a single Transducer-based model designed to handle multiple tasks. This is achieved by integrating task-specific tokens into the reference text during ASR model training, streamlining the inference and eliminating the need for separate NLP models. In addition to ASR, we conduct experiments on 3 different tasks: speaker change detection, endpointing, and NER. Our experiments on a public and a private dataset show that the proposed method improves ASR by up to 7.7% in relative WER while outperforming the cascaded pipeline approach in individual task performance. Additionally, we present task transfer learning to a new task within an existing TokenVerse.

Index Terms: multitask training, speech recognition, speaker change detection, named entity recognition, XLSR-Transducer

1. Introduction

Automated analysis of conversational audios has a wide range of practical applications, including in contact center analytics [1, 2]. Traditionally, conversational audios are transcribed with intermediate voice activity detection (VAD) [3] or endpointing [4] and diarization [5]. Afterward, separate NLP pipelines are employed on the transcripts to perform tasks such as named entity recognition (NER) [6], among others, to comprehend the conversation’s structure and content [7, 8]. Using separate models for each subtask (optimized independently) has drawbacks [9] such as error propagation and a potential mismatch between automatic speech recognition (ASR) metrics and the final task. For instance, the best ASR hypothesis may not be optimal for the final task. Moreover, the cascaded approaches could translate to increased compute and latency, which will be exacerbated by the introduction of a new task.

In this paper, we introduce TokenVerse, a neural Transducer [10] model capable of learning ASR and multiple additional tasks through the incorporation of task tokens. In contrast to the multi-head based multitasking approaches explored in previous studies [11, 12, 13], TokenVerse distinguishes itself by generating tokens directly within the ASR hypothesis, as illustrated in Fig. 1a. Leveraging the transducer architecture [10], we can attain text-audio alignment for each output token, including those designated as task tokens. For example, we can perform NER directly in the acoustic domain, presenting potential utility in scenarios such as audio de-identification [14]. To address challenges in low-resource settings, we use self-supervised (SSL) trained XLSR-53 [15] model as an encoder in

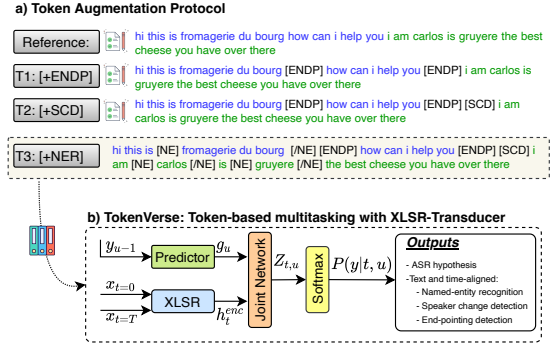


Figure 1: a) Proposed unified token augmentation protocol for SCD, ENDP, and NER. b) TokenVerse unifies multiple speech and NLP tasks (e.g., T1+T2+T3) in a single model within the neural transducer framework.

the transducer setup, leading to the XLSR-Transducer (Fig. 1b). Previous works aim at modeling several tasks directly from speech using special tokens [16, 17], or ASR with speaker change detection (SCD) [18, 19, 13], VAD [20], speech-to-text translation [21], or timestamps [22], NER [9, 23] and multi-speaker ASR [24, 25]. Token-based multitasking offers multiple benefits, e.g., it has a fixed number of parameters while all tasks are predicted with standard decoding without increased latency. However, NLP tasks like NER in conjunction with other tasks from audio domains have not received much attention in the literature. Therefore, we consider 3 additional tasks alongside ASR: SCD, endpointing and NER. These tasks are selected to represent both audio and NLP domains. SCD is an audio task [26]. Endpointing can be viewed as an NLP task when conducting semantic endpointing [27], or as an audio task [4]. NER is an NLP task [6, 9]. They serve as suitable benchmarks for evaluating our proposed method.

2. TokenVerse

Through TokenVerse, we aim to train a single model for ASR (main task), speaker change detection (SCD), endpointing, and named entity recognition (NER). This is achieved by augmenting the reference text, with task tokens that denote special events at the acoustic level. In the following sections, we discuss the annotation protocol, dataset preparation, details of our ASR model and ablation experiments.

2.1. Token Augmentation Protocol

We introduce "tokens" for tasks apart from ASR: [SCD] (speaker change detection), [NE] and [/NE] (named entity

recognition), and [ENDP] (endpointing). An illustrative example is depicted in Figure 1a. We insert [SCD] token during text concatenation if there is a speaker change from one segment to another within an utterance. The [ENDP] token is inserted at the end of a segment text, considered as a semantic endpoint from the conversational context perspective. Note that occurrence of [ENDP] will be a superset of [SCD] because a speaker change indicates the completion of the previous speaker’s sentence. For NER, we insert [NE] before the start of a named entity (NE) and [/NE] after it is concluded, since it can comprise multiple words.

2.2. Dataset Preparation

Our work is focused on conversational audios which is typically long in duration (avg 5 minutes) and can’t be directly used for ASR training due to high GPU memory requirements. The dataset provides audio-text transcripts together with timestamp information for every segment within the long-form audio. For each sample, we begin with the first segment *start* and find the farthest segment *end* such that the duration is up to 20 seconds. Audios within this range are extracted as one utterance and this procedure is repeated until the last segment is consumed. Note that an utterance may span over multiple segments, potentially containing silences, noise, speaker changes, endpoints and numerous named entities. Afterward, we concatenate the text corresponding to all segments within an utterance, inserting token at appropriate positions according to our tasks, described in §2.1. This multitask dataset preparation approach applies universally across all datasets (see §4.1) used in our experiments.

2.3. Training & Inference

TokenVerse Training We train the XLSR-Transducer model on the multitask data which consists of XLSR encoder, state-less predictor [28] and joint networks (linear layer). The model is trained with pruned transducer loss [29]. We utilize SentencePiece [30] tokenizer to train subwords from the training text [31]. It is important to note that the text includes task-specific tokens, and splitting them into multiple subwords may degrade their prediction accuracy because the entire sequence of subwords for a token must be predicted correctly to count it as a valid token prediction. Hence, we ensure that tokens are represented by a single subword during their training.¹

TokenVerse Inference We generate hypothesis with beam search. From the hypothesis, we can extract and align the predicted task tokens in the time domain. Since NER consists of two tokens, we extract words between a matched pair of [NE] and [/NE]. We discard any unpaired tag from the hypothesis. To obtain timestamps for [SCD] or [ENDP], we note the acoustic frame index for which these tokens are emitted and calculate time information, i.e., XLSR acoustic embeddings have a frame duration of 25ms and a stride of 20ms. Particularly for [SCD], the time-level token prediction enables subsequent tasks, e.g., diarization [19].

2.4. Ablations within TokenVerse

We conduct ablation experiments to understand how including or excluding tasks affects other tasks in the TokenVerse. Note that ASR is our primary task and is always included.

Single task For each task, we retain only the tokens specific to that task in the multitask dataset and train our XLSR-

Table 1: *Datasets statistics with token metadata per subset for the public and private datasets.*

subset	Datasets metadata		Token-based metadata [%]				
	#utt/word	dur [h]	[SCD]	[NE]	[ENDP]	#NE	#uniq
DefinedAI dataset							
train	10k/359k	40	1.9	3.6	2.1	6.5k	2350
dev	559/20k	2.25	2.0	3.6	2.1	379	232
test	1.1k/42k	4.5	1.9	3.4	2.0	727	378
CallHome dataset							
train	2.7k/198k	13	6.3	2.9	8.7	2.8k	1414
dev	641/52k	3	7.2	3.0	10.4	779	466
test	339/23k	1.5	6.0	3.0	9.6	351	220

Transducer model. This helps eliminate any detractor tasks that may affect the performance of the task being evaluated and serves as a baseline in this paper.

Leave-one-task-out We systematically exclude tokens corresponding to a single task from the multitask dataset and proceed to train our ASR model. These experiments aim to examine how the removal of a task affects all other tasks, including ASR. This provides insights into whether we should retain or discard any task in TokenVerse for optimal performance on a given task.

Task-Transfer Learning In conventional multi-head multi-task architectures [11, 12], integrating a new task typically necessitates fine-tuning the model on the specific task while keeping the base encoder and other heads frozen. We explore the viability of this extension for TokenVerse by fine-tuning the model, derived from the removal of a task, specifically on the new task. Furthermore, we evaluate its impact on both existing tasks and the performance of the new task in comparison to the overall performance when all-tasks are included.

3. Task-Specific Baselines, Metrics & Evaluation Protocol

In this section, we describe strong independent baselines for each task considered in this work.

Automatic Speech Recognition We train our XLSR-Transducer model after removing all task tokens from the multitask dataset. This serves as a baseline for comparison with the multitask models on the ASR task. **Evaluation** It is evaluated with WER. For TokenVerse models, we remove task tokens from both the reference and hypothesis to compute WER for a fair comparison. We also report WER including task tokens, which reflects its prediction errors.

Named-Entity Recognition We finetune pretrained BERT² [32] model on our datasets for subword-level NER classification. We evaluate the models on both reference and hypothesis from the ASR model. **Evaluation** NER systems are usually evaluated by comparing their outputs against human annotations, either using an exact-match or soft-match approach [6]. We adapted these metrics to a scenario where the text comes from an ASR system. *Exact-Match*: Let $P = \{P_1, P_2, \dots, P_n\}$ be the set of predicted entities, and $A = \{A_1, A_2, \dots, A_n\}$ be the set of actual entities, where each P_i and A_i is accompanied by its corresponding [NE]-[/NE] tokens (See Fig.1). Thus, an entity P_i is considered correctly identified if and only if: $\forall i \in \{1, 2, \dots, n\}, P_i = A_i$, includ-

¹<https://github.com/google/sentencepiece>

²<https://huggingface.co/google-bert/bert-base-uncased>

ing the tokens. *Soft-Match*: in this case we only count for the paired sets of [NE]-[/NE] tokens without considering if the predicted entity value P_i was correctly transcribed. After obtaining each pair, we evaluate NER with F1-score.

Speaker Change Detection For the SCD baseline, we utilize the diarization pipeline³ from PyAnnote [33] to extract speaker change timestamps from the audio. In literature, the SCD is predominantly regarded as a task within the audio domain [26], we opt not to establish an independent text-based baseline for this task. **Evaluation** We evaluate SCD in two ways: text-based (only valid for `TokenVerse`) and time-based. In text-based evaluation, we align the reference and hypothesis using edit-distance. For each occurrence of the [SCD] token in the reference, matching with the same token in the hypothesis counts as True Positive; else, False Negative. Unmatched tokens in the hypothesis are considered False Positive. F1 score is calculated by standard definitions. In time-based evaluation, we obtain the timestamps where [SCD] tokens are predicted in the hypothesis. We calculate F1 score [13], using a collar of 250ms during timestamp matching, following common practice in speaker diarization literature [5]. Additionally, segment coverage, purity [26], and their F1 score are also reported. We use `pyannote.metrics` [34] to compute all time-based metrics.

Endpointing Considering semantic endpointing, we fine-tune BERT [32] for [ENDP] token classification on the multitask training text, termed as BERT-ENDP. Results are reported on both reference text and hypothesis text obtained from `TokenVerse`. From the audio perspective, we use segmentation pipeline⁴ from PyAnnote to obtain endpoint timestamps.

Evaluation Endpointing is also evaluated in two ways: text-based and time-based. The text-based evaluation follows the same approach as described previously for SCD. In the time-based evaluation, the F1 score computation also follows the same approach as for SCD. Additionally, we also report false alarms (FA), missed speech (MS), and detection error rate (DER), which are common metrics in endpointing literature [3].

4. Experimental Setup

4.1. Dataset

To train `TokenVerse`, we require conversational audio data with corresponding transcripts, NER and segment timestamps, and speaker annotations. We could not find a large-scale public dataset satisfying all the tasks. Thus, we opt for a private dataset (*DefinedAI*⁵) which contains stereo-audio/transcript pairs for contact center conversations between agents and customers. We upsampled audio from 8 kHz to 16 kHz to align with the XLSR-53 model’s requirements. Each segment includes transcripts, speaker ID and NE annotations, facilitating multitask dataset preparation (Sec 2.2). This dataset spans health, banking and finance domains, which makes it particularly challenging due to variations in NEs. Additionally, we train and evaluate `TokenVerse` on the open-source *CallHome* English dataset (LDC97S42), which contains natural conversational stereo-audios between multiple speakers. The transcript includes named entities annotation. This dataset poses challenges due to its natural conversational nature, known to be challenging for ASR modeling, and a large number of short segments without entities, differing from the *DefinedAI* dataset. Further details about these datasets are provided in Table 1.

³huggingface.co/pyannote/speaker-diarization-3.1

⁴huggingface.co/pyannote/segmentation-3.0

⁵<https://www.defined.ai/>

Table 2: WERs (%) for ASR on *DefinedAI* with `TokenVerse`.
†task tokens are removed from both referene and hypothesis.

Exp	Model	w/ token	w/o token [†]
1)	ASR (baseline)		15.3
2)	all-tasks	15.6	14.7
3-a)	single-[SCD]	15.2	15.1
3-b)	single-[NE]	15.3	14.7
3-c)	single-[ENDP]	14.8	14.7

4.2. Training `TokenVerse`

We train `TokenVerse` on the multitask dataset. It involves XLSR-transducer model, which is constructed from the Icefall’s Transducer recipe⁶ adapted with XLSR from fairseq [35] as the encoder. The fine-tuning uses Scaled Adam [36] and a learning rate scheduler that consists of a 500-step warmup phase followed by a decay phase directed by the number of steps and epochs. The model is optimized with pruned RNN-t loss [29]. The learning rate is set to $lr=1.25e^{-3}$ and we train the model for 50 epochs. For each dataset, the best epoch is selected based on the WER on respective dev sets and results are presented on the eval sets. The task-transfer experiments, described in §2.4, are trained for additional 10 epochs on the new task.

5. Results & Discussion

Automatic Speech Recognition For the *DefinedAI* (Tab. 2) set, WERs are reported both with and without task tokens in the reference and hypothesis for multitask models. However, the baseline ASR model is trained without task tokens in transcripts, so there is no distinction between them. Including all tasks in `TokenVerse` (exp 2) leads to a 4% relative improvement in WER compared to the baseline ASR model (exp 1). For models trained on a single task (exp 3a-c), ASR results remain similar except for SCD. When comparing WERs before and after token removal, we observe a relatively large gap between all-tasks and single-task models, potentially due to higher token insertion or deletion as compared to non-token words in the hypothesis. In single-task models, a larger gap is observed for [NE] as the model must accurately predict both tokens, introducing additional error sources. On the *CallHome* dataset (Tab. 5), the multitask model with all tokens yields a 7.7% relative improvement. Overall, the results on both datasets indicate that the all-tasks `TokenVerse` improves ASR performance.

Named-Entity Recognition As expected, compared to evaluating BERT-NER on reference text, a significant degradation is observed when evaluated on hypothesis (Tab. 3) due to ASR errors [9]. In exact-match, on both the *DefinedAI* (Tab. 3) and *CallHome* (Tab. 5) test sets, the all-tasks `TokenVerse` outperforms the baseline BERT-NER models trained on their respective datasets and evaluated on hypothesis in F1 score. This is not the case for soft-match evaluation on the *DefinedAI* test set, where the F1 score is similar. This degradation is mostly attributed to the incorrect prediction of [/NE] tag by the baseline, resulting in only a partial match of the named entity words. The absolute F1 score is low on the *CallHome* dataset due to higher ASR errors on named entities, attributed to their low repetition in the training text (see Tab. 1).

Speaker Change Detection On the *DefinedAI* (Tab. 4), in-

⁶<https://github.com/k2-fsa/icefall/tree/master/egs/librispeech/ASR/zipformer>

Table 3: Text-based performances of *TokenVerse* on the the [NE] (exact- and soft-match) and [ENDP]. P: precision; R: recall. [†]upper-bound: BERT model evaluated on text references. [‡]model trained on [ENDP] or [NE] task.

Exp	Model	[NE]-Exact			[NE]-Soft			[ENDP]
		@P	@R	@F1	@P	@R	@F1	@F1
BERT: fine-tuned on DefinedAI								
b-1)	Eval. on Ref. [†]	80.0	77.0	78.5	91.6	87.9	89.7	81.6
b-2)	Eval on Hyp.	52.9	53.0	52.9	82.0	81.3	81.6	80.5
2)	all-tasks	65.0	51.7	57.6	93.0	73.2	81.9	89.9
3-b/c)	single [‡]	61.7	49.9	55.2	91.4	73.3	81.4	88.5

Table 4: [SCD] and [ENDP] time-based evaluation. FA: false alarm; MS: missed speech; DER: detection error rate. [†]F1-score computed from the Coverage-Purity perspective. [‡]single-task model per task, i.e., SCD and ENDP.

Exp	Model	SCD		EndPointing			
		F1	CP-F1 [†]	F1	FA	MS	DER
b-1/2)	PyAnnote	69.6	92.2	73.5	1.1	8.5	9.6
2)	all-tasks	79.7	97.7	85.7	4.7	1.4	6.1
3-a/c)	single [‡]	87.5	97.6	84.1	1.9	2.0	3.9

cluding all tasks in *TokenVerse* outperforms the baseline PyAnnote model in time-based evaluations. Interestingly, models trained for single-task SCD perform better than the all-tasks model in terms of F1, but show similar results for Coverage-Purity based F1. Upon closer scrutiny, we found that including [ENDP] delays the prediction for [SCD] tokens, causing the hypothesis timestamps of these tokens to fall outside the tolerance window (250ms). Increasing the tolerance window further improves the F1 for both models, with a much higher rate of increase for the all-tasks model. This observation is reinforced in the text-based F1 score, where the all-tasks model achieves an F1 score of 90.3% compared to 88.5% from the single-[SCD] model. On the *CallHome* test set (Tab. 5), the all-tasks model outperforms the PyAnnote baseline. These evaluations suggest that excluding [SCD] from *TokenVerse* is preferable for precise speaker change timestamps, while including all tasks improves speaker-attributed text segmentation.

Endpointing In text-based evaluation on the *DefinedAI* (Tab. 3) and *CallHome* (Tab. 5) test sets, the all-tasks *TokenVerse* outperforms the BERT-ENDP models trained on respective datasets. Additionally, on the *DefinedAI* dataset, we evaluate the BERT-ENDP model on both reference and hypothesis to understand the effect of ASR errors on [ENDP] token prediction. Interestingly, we do not observe a significant degradation when evaluating on the hypothesis compared to the reference. This suggests that errors introduced by ASR may not drastically affect the semantic meaning of the sentences. In time-based evaluation on the *DefinedAI* test set (Tab 4), the all-tasks model outperforms the baseline PyAnnote segmentation model. However, single-task ENDP is better than including all tasks in DER due to lower false alarms.

5.1. Ablation results

In ASR, we observed degradation for all ablation experiments (see §2.4), with the largest relative degradation of 2.4% in WER when [ENDP] was removed. Transfer learning on any of the 3

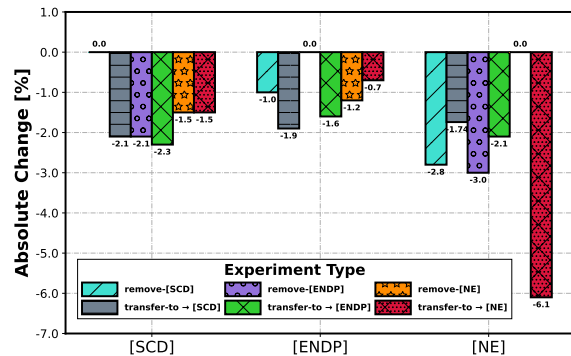


Figure 2: Absolute changes in text-based evaluation w.r.t all-tasks *TokenVerse* in @F1. We either remove a task, e.g., remove-[NE], or transfer to the removed task, e.g., transfer-to → [NE]. Note that all-tasks *TokenVerse* performs better in all scenarios.

Table 5: F1-score and WERs for *CallHome* Eval set on different tasks with *TokenVerse*. [†]time-based F1 score. [‡]baselines are computed with PyAnnote for SCD or with fine-tuned BERT on ENDP and NER (exact-match).

Exp	ASR	SCD [†]	ENDP	NER
	WER (↓)	F1 (↑)	F1 (↑)	F1 (↑)
baselines [‡]	24.6	91.7	55.9	27.4
all-tasks	22.7	92.5	73.3	30.6

tasks do not degrade ASR performance further. The text-based evaluations of other tasks on *DefinedAI* are reported in Figure 2; absolute change is calculated from the all-tasks model. Removing a task adversely affects other tasks. Specifically, for SCD and endpointing, [NE] removal has the least impact on performance. Learning it afterward either improves or maintain their performance, indicating a stronger correlation between these tasks than with NER; supported by the degradation in [SCD] performance when [ENDP] is removed. Task transfer on [ENDP] degrades the performance further, possibly due to confusion during prediction caused by the insertion of the token before [SCD] during training. Transfer to NER shows relatively large degradation compared to other tasks, likely because the model must predict both [NE] and [/NE] accurately. This suggests that tasks encoded with multiple tokens may not transfer as effectively as those encoded with a single token.

Overall, all-tasks *TokenVerse* outperforms specialized models for each task and single-task models suggesting that additional tasks improve each other. Moreover, our task transfer experiments suggest that a new task can be learned effectively.

6. Conclusions

In this paper, we demonstrate the effectiveness of a token-based multitask model on speech and NLP using XLSR-Transducer as our ASR model, termed *TokenVerse*. We consider speaker change detection, endpointing and named entity recognition as 3 additional tasks alongside ASR. Results on 2 datasets show that our approach improves ASR performance while outperforming strong task-specific baselines. Ablation experiments suggest that multitask training across different domains can enhance performance on all tasks. Our approach offers flexibility for extension to numerous tasks across various domains.

7. References

- [1] M. Saberi, O. Khadeer Hussain, and E. Chang, "Past, present and future of contact centers: a literature review," *Business Process Management Journal*, vol. 23, no. 3, pp. 574–597, 2017.
- [2] J. Mamou, D. Carmel, and R. Hoory, "Spoken document retrieval from call-center conversations," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, 2006, pp. 51–58.
- [3] I. Medennikov, M. Korenevsky, T. Prisyach, Y. Khokhlov, M. Korenevskaya, I. Sorokin, T. Timofeeva, A. Mitrofanov, A. Andrusenko, I. Podluzhny *et al.*, "Target-speaker voice activity detection: a novel approach for multi-speaker diarization in a dinner party scenario," *arXiv preprint arXiv:2005.07272*, 2020.
- [4] S.-Y. Chang, R. Prabhavalkar, Y. He, T. N. Sainath, and G. Simko, "Joint endpointing and decoding with end-to-end models," in *ICASSP*. IEEE, 2019, pp. 5626–5630.
- [5] T. J. Park, N. Kanda, D. Dimitriadis, K. J. Han, S. Watanabe, and S. Narayanan, "A review of speaker diarization: Recent advances with deep learning," *Computer Speech & Language*, vol. 72, p. 101317, 2022.
- [6] J. Li, A. Sun, J. Han, and C. Li, "A survey on deep learning for named entity recognition," *IEEE transactions on knowledge and data engineering*, vol. 34, no. 1, pp. 50–70, 2020.
- [7] Y. Zou, L. Zhao, Y. Kang, J. Lin, M. Peng, Z. Jiang, C. Sun, Q. Zhang, X. Huang, and X. Liu, "Topic-oriented spoken dialogue summarization for customer service with saliency-aware topic modeling," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 16, 2021, pp. 14 665–14 673.
- [8] Y. Xu, H. Zhao, and Z. Zhang, "Topic-aware multi-turn dialogue modeling," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 16, 2021, pp. 14 176–14 184.
- [9] S. Ghannay, A. Caubriere, Y. Esteve, A. Laurent, and E. Morin, "End-to-end named entity extraction from speech," *arXiv preprint arXiv:1805.12045*, 2018.
- [10] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.
- [11] Y.-C. Chen, S.-w. Yang, C.-K. Lee, S. See, and H.-y. Lee, "Speech representation learning through self-supervised pretraining and multi-task finetuning," *arXiv preprint arXiv:2110.09930*, 2021.
- [12] S. wen Yang, P.-H. Chi *et al.*, "SUPERB: Speech Processing Universal PERFORMANCE Benchmark," in *Proc. Interspeech 2021*, 2021, pp. 1194–1198.
- [13] S. Kumar, S. Madikeri, N. Iuliia, E. VILLATORO-TELLO, P. Motlicek, K. P. D. S, S. P. Dubagunta, and A. Ganapathiraju, "Multitask speech recognition and speaker change detection for unknown number of speakers," in *Proceedings of the 49th IEEE International Conference on Acoustics, Speech, & Signal Processing (ICASSP) 2024*, 2024.
- [14] I. Cohn, I. Laish, G. Beryozkin, G. Li, I. Shafran, I. Szpektor, T. Hartman, A. Hassidim, and Y. Matias, "Audio de-identification: A new entity recognition task," *arXiv preprint arXiv:1903.07037*, 2019.
- [15] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised cross-lingual representation learning for speech recognition," *arXiv preprint arXiv:2006.13979*, 2020.
- [16] Y. Wu, S. Maiti, Y. Peng, W. Zhang, C. Li, Y. Wang, X. Wang, S. Watanabe, and R. Song, "Speechcomposer: Unifying multiple speech tasks with prompt composition," *arXiv preprint arXiv:2401.18045*, 2024.
- [17] K.-W. Chang, Y.-K. Wang, H. Shen, I.-t. Kang, W.-C. Tseng, S.-W. Li, and H.-y. Lee, "Speechprompt v2: Prompt tuning for speech classification tasks," *arXiv preprint arXiv:2303.00733*, 2023.
- [18] L. E. Shafey, H. Soltan, and I. Shafran, "Joint speech recognition and speaker diarization via sequence transduction," *arXiv preprint arXiv:1907.05337*, 2019.
- [19] W. Xia, H. Lu, Q. Wang, A. Tripathi, Y. Huang, I. L. Moreno, and H. Sak, "Turn-to-diarize: Online speaker diarization constrained by transformer transducer speaker turn detection," in *ICASSP*. IEEE, 2022, pp. 8077–8081.
- [20] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International Conference on Machine Learning*. PMLR, 2023, pp. 28 492–28 518.
- [21] J. P. Zuluaga-Gomez, Z. Huang, X. Niu, R. Paturi, S. Srinivasan, P. Mathur, B. Thompson, and M. Federico, "End-to-end single-channel speaker-turn aware conversational speech translation," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023, pp. 7255–7274.
- [22] S. Cornell, J.-w. Jung, S. Watanabe, and S. Squartini, "One model to rule them all? towards end-to-end joint speaker diarization and speech recognition," *arXiv preprint arXiv:2310.01688*, 2023.
- [23] H. Yadav, S. Ghosh, Y. Yu, and R. R. Shah, "End-to-end named entity recognition from english speech," *arXiv preprint arXiv:2005.11184*, 2020.
- [24] N. Kanda, J. Wu, Y. Wu, X. Xiao, Z. Meng, X. Wang, Y. Gaur, Z. Chen, J. Li, and T. Yoshioka, "Streaming multi-talker asr with token-level serialized output training," *arXiv preprint arXiv:2202.00842*, 2022.
- [25] J. Wu, N. Kanda, T. Yoshioka, R. Zhao, Z. Chen, and J. Li, "t-sot fnt: Streaming multi-talker asr with text-only domain adaptation capability," *arXiv preprint arXiv:2309.08131*, 2023.
- [26] H. Bredin, C. Barras *et al.*, "Speaker change detection in broadcast tv using bidirectional long short-term memory networks," in *Interspeech 2017*. ISCA, 2017.
- [27] A. Raux and M. Eskenazi, "Optimizing endpointing thresholds using dialogue features in a spoken dialogue system," in *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*, 2008, pp. 1–10.
- [28] M. Ghodsi, X. Liu, J. Apfel, R. Cabrera, and E. Weinstein, "Rnn-transducer with stateless prediction network," in *ICASSP*. IEEE, 2020, pp. 7049–7053.
- [29] F. Kuang, L. Guo, W. Kang, L. Lin, M. Luo, Z. Yao, and D. Povey, "Pruned rnn-t for fast, memory-efficient asr training," *arXiv preprint arXiv:2206.13236*, 2022.
- [30] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," *arXiv preprint arXiv:1808.06226*, 2018.
- [31] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016, pp. 1715–1725.
- [32] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. [Online]. Available: <https://aclanthology.org/N19-1423>
- [33] H. Bredin, "pyannote. audio 2.1 speaker diarization pipeline: principle, benchmark, and recipe," in *24th INTERSPEECH Conference*. ISCA, 2023, pp. 1983–1987.
- [34] Bredin, Hervé, "pyannote.metrics: A toolkit for reproducible evaluation, diagnostic, and error analysis of speaker diarization systems," in *Proc. Interspeech 2017*. ISCA, 2017, pp. 3587–3591.
- [35] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," in *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.
- [36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.