Encoder-decoder multimodal speaker change detection

Jee-weon Jung[1,2], Soonshin Seo[1,2], Hee-Soo Heo[2], Geonmin Kim[2],
You Jin Kim[2], Young-ki Kwon[2], Minjae Lee[2], and Bong-Jin Lee[2]

[1]NAVER Corporation, Republic of Korea [2]NAVER Cloud Corporation, Republic of Korea

jeeweon.jung@navercorp.com

Abstract

The task of speaker change detection (SCD), which detects points where speakers change in an input, is essential for several applications. Several studies solved the SCD task using audio inputs only and have shown limited performance. Recently, multimodal SCD (MMSCD) models, which utilise text modality in addition to audio, have shown improved performance. In this study, the proposed model are built upon two main proposals, a novel mechanism for modality fusion and the adoption of a encoder-decoder architecture. Different to previous MMSCD works that extract speaker embeddings from extremely short audio segments, aligned to a single word, we use a speaker embedding extracted from 1.5s. A transformer decoder layer further improves the performance of an encoder-only MMSCD model. The proposed model achieves state-of-the-art results among studies that report SCD performance and is also on par with recent work that combines SCD with automatic speech recognition via human transcription.

Index Terms: speaker change detection, speaker segmentation, sequence-to-sequence, speaker embedding

1. Introduction

Speaker change detection (SCD), also known as speaker segmentation, is a critical task that divides an input into multiple speaker-homogeneous segments by detecting the points where speakers change. SCD is essential in several applications, with speaker diarisation being the most common use case [1–3]. However, SCD can also be applied to various other tasks such as automatic speech recognition [4].

Pioneer works in the SCD literature have developed models using audio recordings as input [5, 6]. Various frameworks have been proposed to address this task. However, these systems have had limited performance. In our analysis, this is because audio-only SCD is a fundamentally difficult task to solve without additional information; we analyse audio-only SCD as a sequential speaker verification with extremely short segments. Despite recent advances in speaker recognition, stable performance can only be achieved for segments longer than one second. However, audio-only SCD needs to conduct speaker verification for every two consecutive frames where each frame typically conveys 10 to 20 ms of speech.

The challenge of audio-only SCD is further evident in current speaker diarisation solutions. Speaker diarisation, which solves "who spoke when," comprises speaker segmentation (i.e., SCD) and clustering. Reliable SCD would allow us to apply clustering algorithms straightforwardly to SCD outputs. However, the majority of current solutions instead adopt endpoint detection and extract speaker embedding extractors from the voiced regions using a sliding window [7, 8].

As a natural progression, researchers have investigated text-based or audio-text multimodal SCD (MMSCD) solutions [9]. Text-based and audio-text SCD models have shown to improve performance compared to audio-only models. However, reported ablations have shown that text contributes much more than the audio modality in the SCD task. Additionally, several studies only report diarisation performances, which hinders future researchers to compare and select which SCD to employ when constructing a process pipeline for different tasks using SCD.

In this study, we focus on MMSCD itself, excluding diarisation or other applications out of scope. We believe that if MMSCD becomes reliable and robust, it can be easily adopted for several other tasks, including speaker diarisation. Compared to other multimodal combinations, audio-text is relatively easy to implement due to advances in automatic speech recognition that transcribe and derive text from speech[1].

Our proposed MMSCD model is built upon two main proposals. Firstly, we analyse the limitations of existing modality fusion mechanisms and propose a novel mechanism that enables audio modality to contribute more. Secondly, we propose to adopt the encoder-decoder (i.e., sequence-to-sequence (seq2seq)) architecture instead of encoder-only architecture in SCD for the first time and prove its effectiveness. We conduct experiments using two widely adopted datasets in the literature: Switchboard1 [10] and AMI corpus [11]. Our proposed model demonstrates competitive performance, achieving state-of-the-art results among studies that report SCD performance. Our model is also on par with another line of recent work in SCD that combines SCD with automatic speech recognition via large-scale human transcription [12].

The rest of this paper is organised as follows. Section 2 describes the proposed MMSCD model. Experiments and corresponding results are discussed in Sections 3 and 4, followed by a conclusion.

2. Proposed MMSCD model

The overall framework of our MMSCD model is illustrated in Figure 1.

2.1. Prerequisites: single-modal embedding extractors

Our proposed MMSCD model is built upon two single-modal embedding extractors each of which extracts speaker and text embeddings from an input.

Speaker embedding extractor. We utilise the speaker em-

---

[1]It is worth noting that this paper deals with ground truth transcriptions. In real-world scenarios, these transcriptions should be replaced by automatic speech recognition systems' outputs.
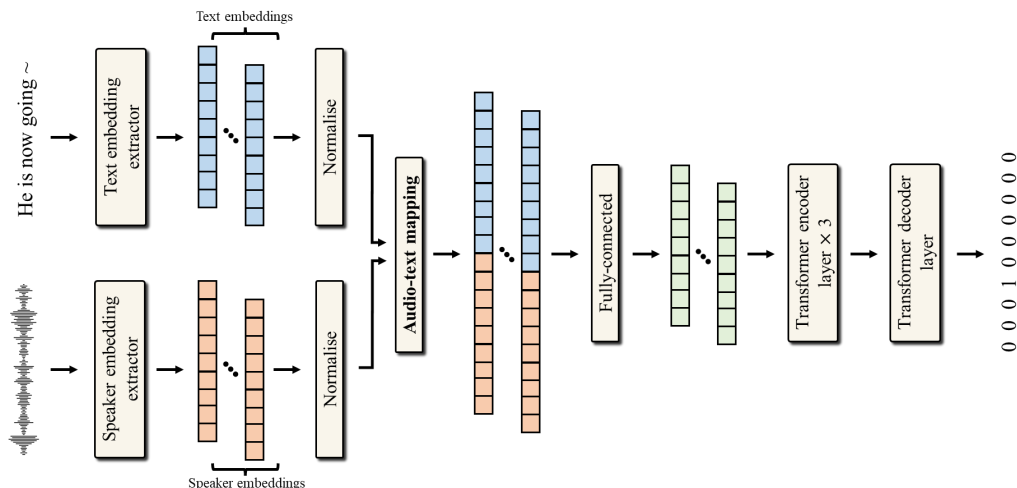
Figure 1: *Overall framework of the proposed multimodal speaker change detection (MMSCD).*
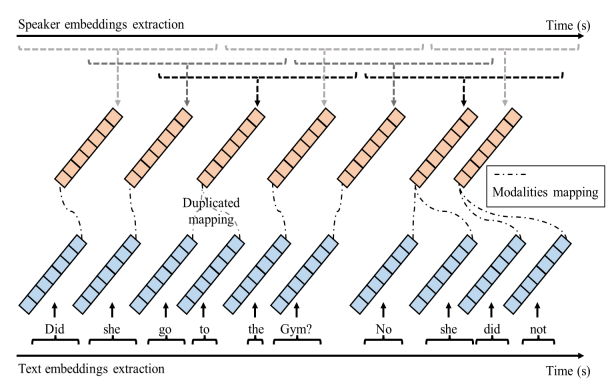

Figure 2: *Proposed audio-text mapping for MMSCD input. Different from existing methods that extract speaker embedding using the onset and offset of a word, we extract speaker embeddings in a sliding window fashion and then map a speaker embedding closest to the corresponding word.*

bedding extractor proposed in [13, 14]. It adopts the MFA-Conformer [15] architecture, a variant of the Conformer [16] model. MFA-Conformer does not modify the Conformer encoders, but concatenates all encoder outputs and then employs an attentive statistics pooling [17] sequentially. The model is trained using a massive amount of labelled data and with data augmentations designed to handle multiple speakers being present in a short audio segment. The model extracts 256-dimensional speaker embedding from a segment of 1.5s with a shift size of 0.5s.

RoBERTa. We adopt RoBERTa, proposed in [18], as the text embedding extractor. We use the pre-trained RoBERTa-base model[2], without further modification or fine-tuning. The model extracts 768-dimensional text embeddings from each sub-word input.

2.2. Modality fusion with stable speaker embeddings

Integrating two different modalities is one of the most critical components of multimodal models. Several strategies have

---

[2]Available at: `https://huggingface.co/roberta-base`.

been explored [19]. In terms of timing, both early and late fusions have been reported to be effective depending on different cases, with early-fusion being more dominant. In the case of MMSCD, previous works [9, 20] have mostly adopted early-fusion. We also choose early fusion following the majority of preceding works.

We focus on the unit of each modality when composing an input for the MMSCD model. Previous works have adopted the word token as a unit, as it is a competitive candidate, compared to "*audio frames*," where the typical shift size is 10ms or 20ms. By using word (or sub-word) as the input unit, audio or speaker embeddings can be extracted from extremely short durations; the duration of a typical word is no longer than 300ms. However, speaker embeddings extracted from such a short duration are highly vulnerable; typically a duration of 3 seconds or longer is required in speaker recognition. Although speaker embedding extraction has experienced rapid breakthroughs, extracting embeddings from a segment less than 2 or 1.5 seconds are is a well known cause of performance degradation [21, 22].

We propose a novel audio-text mapping mechanism for the MMSCD task, illustrated in Figure 2, inspired by the speech segment selection process of [20]. Firstly, speaker embeddings are derived in a sliding window fashion with a 1.5s window and 0.5s shift. When composing an audio-text pair input for the MMSCD model, the speaker embedding which has the closest midpoint to the midpoint of the text in terms of time is selected. Since the proposed mechanism extracts speaker embedding from a longer duration, the information from audio modality can be more reliable and stable. There is a limitation that a few (typically, two) tokens may have identical speaker embedding as a pair. Nevertheless, this should not negatively affect the performance since our system makes the decision token-wise, meaning that our framework assumes audio modality complementing text modality.

The specific process for fusing the two embeddings before feeding them to encoder layers is as follows. We first extract speaker and text embeddings. Each embedding is normalised to have a magnitude of $sqrt(dim)$ following Kaldi recipe [23], where dim is the dimensionality, and then concatenated. Next, a fully-connected layer projects the concatenated vector into another space, thereby fusing the two embeddings. Finally, we add positional encoding to the fused embedding.

Table 1: *Statistics of train and evaluation sets of used datasets.*

Dataset	Domain	Trn dur (h)	Eval dur (h)
Switchboard1	Telephone	248.45	10.68
AMI	Meeting	80.67	9.06

2.3. Seq2seq model architecture

Various deep neural network (DNN) architectures have been explored for the SCD task. In studies of both audio-only and multimodal SCD [24–27], fully-connected, long short-term memory, or gated recurrent unit layers have been widely utilised. Transducer architectures [28, 29], another widely utilised automatic speech recognition framework, have been recently explored in the SCD task. Yet, compared to automatic speech recognition or other seq2seq tasks, existing works only have leveraged encoder layers, remaining decoder layers unexplored.

We aim to investigate whether employing decoder layers to the SCD encoder output can further improve the performance of the MMSCD model. Our underlying assumption is that posing inductive bias to consider past frames and predicting whether the speaker has changed in the current timestep (i.e., autoregressive decoding) may be advantageous for SCD, as it has been proven effective in automatic speech recognition and other tasks.

To extend the model architecture from encoder-only to encoder-decoder [30, 31], we make the following modifications. Firstly, we add a `<bos>` token before the first token. For the corresponding speaker embedding, we duplicate the first speaker embedding of the input. For the label, we append `<eos>` token at the end. Note that during the inference phase, we disregard the `<eos>` token and infer a pre-defined length. Furthermore, we explore whether beam search decoding [32, 33] can enhance the performance of encoder-decoder models.

Autoregressive training. In seq2seq models, a discrepancy exists between training and inference phases. During the training phase, teacher forcing [34] is commonly used as it helps to increase the convergence speed and performance. In this approach, the ground truth labels are fed as input at each timestep instead of model prediction from the previous timestep. However, in the inference phase, such mechanism is not possible, and the model relies upon its own prediction from the previous step. Since the model has no experience of having made incorrect predictions in the previous timesteps, error accumulation caused by recurrent prediction may significantly deteriorate [35, 36]. This effect may be more pronounced in the SCD task since the output is binary and imbalanced. Drawing inspiration from [37, 38], we therefore investigate the possibility of training the model in an autoregressive manner from a certain epoch and compare the results.

3. Experiments

3.1. Datasets and metrics

Two widely adopted datasets are used for verifying the proposals of this study: Switchboard1-release 2 [10] and AMI corpus Mixed headset [11]. For the Switchboard1 dataset, we upsample all recordings to 16kHz and mix them into mono recordings. In addition, since Switchboard1 does not have an official train and evaluation split, we randomly select 100 audio recordings

for evaluation [3]. Table 1 describes the duration specifications of the two datasets. We adopt F1 score and equal error rate (EER) as the primary and the supplementary metrics. EER is used because label imbalance is severe and it equally weights false alarm and false rejections [4].

3.2. Configurations

Speaker embedding extractor. We train the MFA-Conformer model using the *v4* configuration of [14], which includes 88k speakers' 10k hours of speech. It adopts six 512-dimensional Conformer encoder blocks with eight attention heads. 80-dimensional log mel-spectrograms extracted every 10ms using a 25ms window are fed to the model. During the training phase, data augmentations from [13] are adopted as well as adding noise and reverberations.

MMSCD model. The encoder input projection module, which digests concatenated speaker and text embeddings, consists of a fully-connected layer with 512 dimensions, followed by dropout [39] and a GELU [40] non-linearity. The encoder part includes three Transformer [41] layers with 512 dimensions and eight multi-heads for attention. The decoder input projection module comprises identical layers to the encoder input projection module. The only difference is the input dimension; encoder projection module inputs 1024-dimensional vector (256 for speaker and 768 for text) whereas decoder projection module inputs 768-dimensional vector because it only digests text embedding indicating same of different speakers. The decoder part comprises one Transformer decoder layer with 512 dimensions and eight attention heads.

Training details. Models trained for either of the two datasets employ AdamW [42] optimiser with an initial learning rate of $1e^{-3}$ and a weight decay of $5e^{-5}$. We adopt a cosine learning rate scheduler with a minimum learning rate of $5e^{-6}$. Learning rate warm-up is also adopted during the first 1,000 iterations. For models trained on the AMI corpus, the batch size is 32 and the number of epochs is 8,000 [5]. Models trained on the Switchboard corpus adopt a batch size of 64 and 400 epochs.

4. Results

4.1. Main results

Table 2 addresses the main results on two datasets. Firstly, we confirm that including a decoder layer for the SCD task can be advantageous. In both datasets, the inclusion of a decoder layer improved both the F1 score and EER. For instance, in AMI, the F1 score increased from 68.05 to 73.61.

Next, we observe that training and inferring without either modality leads to severe performance degradation. Removing the audio modality significantly deteriorates the performance in our model, with EER on Switchboard1 increasing from 5.90% to 10.45%. This finding contrasts with previous works which report that the audio modality only has a minor contribution to the performance of MMSCD [20]. Therefore, we conclude that the proposed modality matching mechanism is effective.

---

[3]Certain number of seen speakers may exist in the evaluation set because Switchboard1 dataset has reoccurring speakers.

[4]For both datasets, "*speaker change*" label occupies less than 10%.

[5]This large number of epochs is due to the limited number of samples of the AMI train set, which is 136.

Table 2: *Main results. Through comparison experiments conducted on Switchboard1 and AMI datasets, it is shown that adopting seq2seq architecture with a decoder layer, instead of composing an encoder-only architecture is beneficial. We also demonstrate that both modalities contribute to performance, where the audio modality also plays a crucial role.*

	Switchboard1				AMI			
	Prec	**Recall**	**F1**	**EER**	**Prec**	**Recall**	**F1**	**EER**
Proposed	82.28	83.09	82.68	5.90	66.73	80.49	73.61	13.00
w/o decoder	80.84	80.62	80.73	6.36	57.33	78.76	68.05	15.97
w/o audio modality	80.59	68.69	74.60	10.45	61.51	76.72	69.12	15.13
w/o text modality	25.65	88.56	57.10	17.53	20.01	96.77	58.39	27.34

Table 3: *Ablations on the effect of autoregressive training and its starting point. Evaluated on Switchboard1. Total number of epochs is 400 (TF: teacher forcing, AR: autoregressive training).*

# TF/AR epochs	Prec	Recall	F1	EER
400/0	77.94	80.63	79.28	6.63
200/200	**82.91**	82.84	**82.87**	5.94
300/100	82.28	**83.09**	82.68	**5.90**
350/50	83.79	81.35	82.57	5.94

Table 4: *Exploration of beam search. Results reported on Switchboard1. Performance of greedy equals to beam search with width=1 (w: beam width).*

	decoding	Prec	Recall	F1	EER
w/o autoreg trn	greedy	77.94	80.63	79.28	6.63
	beam(w=100)	**78.87**	**81.72**	**80.30**	**6.44**
w autoreg trn	greedy	**82.28**	83.09	82.68	**5.90**
	beam(w=100)	82.26	**83.14**	**82.70**	5.91

4.2. Autoregressive training

Table 3 describes experiments on the effect of autoregressive training. The first row addresses performance without autoregressive training. The other rows depict the result when the training starts with teacher forcing but trains the model in an autoregressive manner during a certain period. The results show that regardless of the starting timing, autoregressive training improves the performance by more than 5%. However, the performance difference between different starting epochs was negligible. Hence it may be a good strategy to begin autoregressive training at the last moment, since autoregressive training slows down the training phase.

4.3. Exploration on beam search

In automatic speech recognition, applying the beam search algorithm [44, 45], often combined with language models, can significantly improve the performance. We apply a typical beam search; however, we leave all nodes for the first few timesteps, since the SCD task has only two labels (i.e., speaker change and non-change), and thus the total beam width can be smaller than the allowed beam width. Table 4 describes the result of our exploration of beam search decoding. When autoregressive training is not applied, beam search was effective in decreasing the EER and increasing the F1 score. However, when autoregressive training was applied, beam search could not bring further improvements. We believe that beam search has future potentials worth exploring; however, at this stage, it only contributes

Table 5: *Comparison with other works. Direct comparisons should be avoided. Refer to explanation in the main text for full details (NR: not reported).*

	Prec	Recall	F1	EER
Switchboard1				
India et al. [43]	62.54	66.70	64.55	NR
Ours	82.28	83.09	82.68	5.90
AMI				
Zhao et al. [12]	79.40	68.1	73.30	NR
Ours	66.73	80.49	73.61	13.00

when the performance is less stable.

4.4. Comparison with other works

Table 5 compares our work with previous works in the literature. However, direct comparisons are not available due to a few different configurations. For comparison on Switchboard1, the authors of [43] composed an evaluation set with mixed Fisher and Switchboard1 datasets. They also adopt forgiveness collars of 500ms and the unit is audio frames. For comparison on AMI, the authors of [12] adopt a custom definition of both precision and recall which makes the task easier. They also adopt forgiveness collars of 250ms and the unit is audio frames. In contrast, our unit is sub-word and we do not adopt any forgiveness collars, hence we argue that our evaluation is the most stringent. Although direct comparisons are not available, even with much more stringent evaluation setting, our model outperforms [43] by a large margin in Switchboard1 and demonstrates on par performance with [12] in AMI.

5. Conclusion

We conducted an investigation into the MMSCD task, validating two main hypotheses. Firstly, we assumed that incorporating decoder layers into an encoder-decoder architecture would be advantageous for the SCD task. We demonstrated that including decoder layers can significantly enhance performance by almost 10%. Secondly, we proposed a novel mechanism for using speaker embeddings. Rather than extracting speaker embeddings using the onset and offset of words, we extracted them using a sliding window and then mapped them to the corresponding words. This mechanism produced more stable speaker embeddings, as they have been extracted from longer durations. Through experiments on two widely-used datasets, we confirmed competitive performance, outperforming previous audio-only and MMSCD models and achieving comparable performance with the recent automatic speech recognition re-annotation-based SCD method.

6. References

[1] M. Hrúz and Z. Zajíc, "Convolutional neural network for speaker change detection in telephone speaker diarization system," in *Proc. ICASSP*, 2017.

[2] H. Bredin, R. Yin, J. M. Coria et al., "Pyannote. audio: neural building blocks for speaker diarization," in *Proc. ICASSP*, 2020.

[3] S.-S. Cheng, H.-M. Wang and H.-C. Fu, "Bic-based speaker segmentation using divide-and-conquer strategies with application to speaker diarization," *IEEE/ACM TASLP*, vol. 18, no. 1, pp. 141–157, 2009.

[4] D. Liu and F. Kubala, "Fast speaker change detection for broadcast news transcription and indexing," in *Sixth European Conference on Speech Communication and Technology*. Citeseer, 1999.

[5] H. Bredin, C. Barras et al., "Speaker change detection in broadcast tv using bidirectional long short-term memory networks," in *Proc. Interspeech*, 2017.

[6] A. S. Malegaonkar, A. M. Ariyaeeinia and P. Sivakumaran, "Efficient speaker change detection using adapted gaussian mixture models," *IEEE/ACM TASLP*, vol. 15, no. 6, pp. 1859–1869, 2007.

[7] T. J. Park, N. Kanda, D. Dimitriadis et al., "A review of speaker diarization: Recent advances with deep learning," *Computer Speech & Language*, vol. 72, pp. 101317, 2022.

[8] Y. Kwon, H.-S. Heo, J.-w. Jung et al., "Multi-scale speaker embedding-based graph attention networks for speaker diarisation," in *Proc. ICASSP*. IEEE, 2022.

[9] T. J. Park and P. Georgiou, "Multimodal speaker segmentation and diarization using lexical and acoustic cues via sequence to sequence neural networks," in *Proc. Interspeech*, 2018.

[10] J. Godfrey and E. Holliman, "Switchboard-1 release 2 ldc97s62," *Linguistic Data Consortium*, p. 34, 1993.

[11] J. Carletta, "Unleashing the killer corpus: experiences in creating the multi-everything ami meeting corpus," *Language Resources and Evaluation*, vol. 41, pp. 181–190, 2007.

[12] G. Zhao, Q. Wang, H. Lu et al., "Augmenting transformer-transducer based speaker change detection with token-level training loss," *arXiv preprint arXiv:2211.06482*, 2022.

[13] J.-w. Jung, H.-S. Heo, B.-J. Lee et al., "In search of strong embedding extractors for speaker diarisation," *Proc. ICASSP*, 2023.

[14] J.-w. Jung, H.-S. Heo, B.-J. Lee et al., "Large-scale learning of generalised representations for speaker recognition," *arXiv preprint arxiv:2210.10985*, 2022.

[15] Y. Zhang, Z. Lv, H. Wu et al., "Mfa-conformer: Multi-scale feature aggregation conformer for automatic speaker verification," in *Proc. Interspeech*, 2022.

[16] A. Gulati, J. Qin, C.-C. Chiu et al., "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech*, 2020.

[17] K. Okabe, T. Koshinaka and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Proc. Interspeech*, 2018.

[18] Y. Liu, M. Ott, N. Goyal et al., "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[19] A. Nagrani, S. Yang, A. Arnab et al., "Attention bottlenecks for multimodal fusion," in *NeurIPS*, 2021.

[20] T. J. Park, K. J. Han, J. Huang et al., "Speaker diarization with lexical information," in *Proc. Interspeech*, 2019.

[21] N. Tawara, A. Ogawa, T. Iwata et al., "Frame-level phoneme-invariant speaker embedding for text-independent speaker recognition on extremely short utterances," in *Proc. ICASSP*, 2020, pp. 6799–6803.

[22] J.-w. Jung, H.-S. Heo, H.-j. Shim and H.-J. Yu, "Short utterance compensation in speaker verification via cosine-based teacher-student learning of speaker embeddings," in *Proc. ASRU*. IEEE, 2019.

[23] D. Povey, A. Ghoshal, G. Boulianne et al., "The kaldi speech recognition toolkit," in *Proc. ASRU*. Dec. 2011, IEEE Signal Processing Society.

[24] R. Wang, M. Gu, L. Li et al., "Speaker segmentation using deep speaker vectors for fast speaker change scenarios," in *Proc. ICASSP*, 2017.

[25] H. Bredin and A. Laurent, "End-to-end speaker segmentation for overlap-aware resegmentation," in *Proc. Interspeech*, 2021.

[26] O. H. Anidjar, I. Lapidot, C. Hajaj et al., "Hybrid speech and text analysis methods for speaker change detection," *IEEE/ACM TASLP*, vol. 29, pp. 2324–2338, 2021.

[27] L. Sarı, S. Thomas, M. Hasegawa-Johnson and M. Picheny, "Pre-training of speaker embeddings for low-latency speaker change detection in broadcast news," in *Proc. ICASSP*, 2019, pp. 6286–6290.

[28] A. Graves, "Sequence transduction with recurrent neural networks," in *ICML Workshop on Representation Learning*, 2012.

[29] Q. Zhang, H. Lu, H. Sak et al., "Transformer transducer: A streamable speech recognition model with transformer encoders and rnn-t loss," in *Proc. ICASSP*, 2020, pp. 7829–7833.

[30] J. K. Chorowski, D. Bahdanau, D. Serdyuk et al., "Attention-based models for speech recognition," in *Proc. NIPS*, 2015, pp. 577–585.

[31] W. Chan, N. Jaitly, Q. Le and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. ICASSP*, 2016, pp. 4960–4964.

[32] Z. Tu, Y. Liu, L. Shang et al., "Neural machine translation with reconstruction," in *Proc. AAAI*, 2017.

[33] M. Freitag and Y. Al-Onaizan, "Beam search strategies for neural machine translation," in *Proc. Workshop on NMT*, 2017, pp. 56–60.

[34] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural computation*, vol. 1, no. 2, pp. 270–280, 1989.

[35] B. Liu, S. Yan, J. Li et al., "A sequence-to-sequence air quality predictor based on the n-step recurrent prediction," *IEEE Access*, vol. 7, pp. 43331–43345, 2019.

[36] A. Zeng, M. Chen, L. Zhang and Q. Xu, "Are transformers effective for time series forecasting?," *arXiv preprint arXiv:2205.13504*, 2022.

[37] X. Liang, L. Wu, J. Li and M. Zhang, "Janus: Joint autoregressive and non-autoregressive training with auxiliary loss for sequence generation," in *Proc. EMNNLP*, 2022.

[38] M. Ghazvininejad, O. Levy and L. Zettlemoyer, "Semi-autoregressive training improves mask-predict decoding," *arXiv preprint arXiv:2001.08785*, 2020.

[39] N. Srivastava, G. Hinton, A. Krizhevsky et al., "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[40] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," *arXiv preprint arXiv:1606.08415*, 2016.

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

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

[43] M. À. India Massana, J. A. Rodríguez Fonollosa and F. J. Hernando Pericás, "Lstm neural network-based speaker segmentation using acoustic and language modelling," in *Proc. Interspeech*, 2017, pp. 2834–2838.

[44] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. PMLR*, 2014, pp. 1764–1772.

[45] A. Hannun, C. Case, J. Casper et al., "Deep speech: Scaling up end-to-end speech recognition," in *arXiv preprint arXiv:1412.5567*, 2014.