CASS-NAT: CTC Alignment-based Single Step Non-Autoregressive Transformer for Speech Recognition

Ruchao Fan[2], Wei Chu[1], Peng Chang[1], Jing Xiao[1]

IEEE ICASSP 2021
June, 2021

fanruchao@g.ucla.edu

[1]PAII Inc., USA
[2]Dept. of Electrical and Computer Engineering, University of California, Los Angeles, USA

Motivation

- Attention-based encoder decoder (AED) speech recognition models achieve great success in recent years, especially for transformer architecture-autoregressive transformer (AT).
- However, the autoregressive mechanism in decoder slows down the inference speed.
- Non-autoregressive transformer (NAT) was proposed for parallel generation to accelerate the inference.
- **Limitations** for current NAT models
  - Iterative NAT still needs multiple generation steps, which **cannot fully exploit** the potential of NAT for faster inference.
  - Single step NAT extracts incomplete acoustic representations on the decoder side, and thus **the WER performance is worse than AT.**

This work

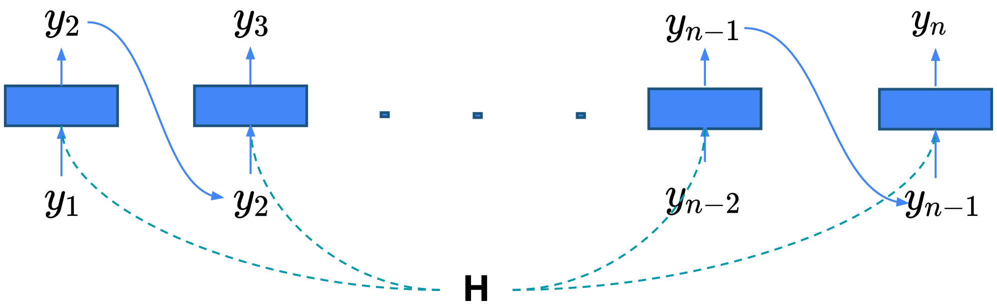
- We propose a novel framework, CTC alignment-based single step NAT (CASS-NAT).
  - CTC alignment provides auxiliary information to extract **token-level acoustic embedding**, which **replaces word embedding** on the decoder side.
- An **error-based sampling alignment strategy during inference** is proposed to improve the WER performance.
- The proposed methods achieve WERs of **3.8%/9.1%** on Librispeech test clean/other dataset without an external LM, and a CER of **5.8% on Aishell1** Mandarin corpus, respectively.
- Compared to the AT baseline, the CASS-NAT has a performance reduction on WER, but is **51.2x faster in terms of RTF.**

Outline

- Non-autoregressive transformer (NAT)
    - Iterative NAT
    - Single step NAT
- Proposed CTC alignment-based single step NAT (CASS-NAT)
    - Framework
    - Training criterion
    - Inference - Error-based sampling alignment (ESA)
- Experiments
- Conclusions

Non-autoregressive Transformer

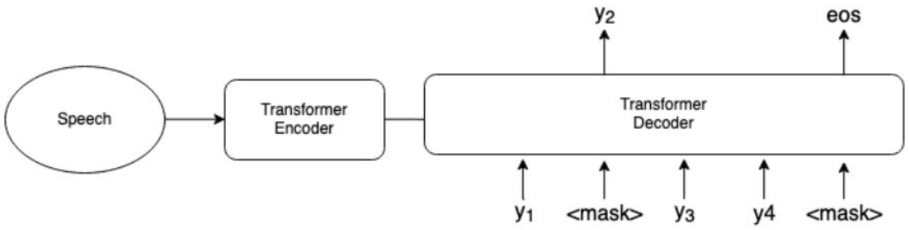
- Autoregressive decoder (neural language model)



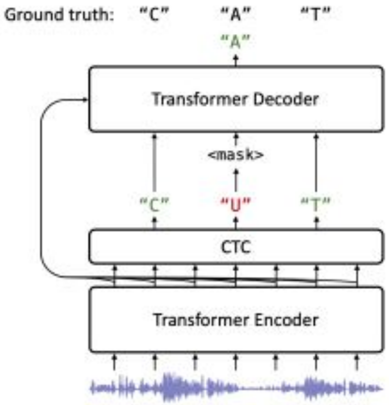
- Non-autoregressive Transformer (NAT) depending on number of iterations
  - Iterative NAT
  - Single step NAT

Iterative NAT

- Decoder training as Mask language model (MLM) [Chen et al. 2019]

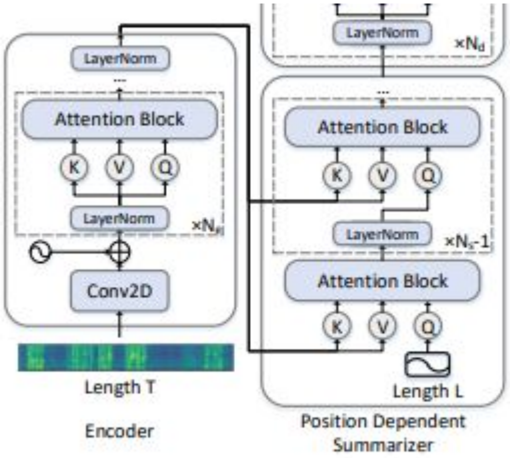


- Refinement based on CTC output [Higuchi et al. 2020]
  a. Mask CTC output with low confidence
  b. Predict masked frames using unmasked frames
  c. Conduct the two steps for several iterations.

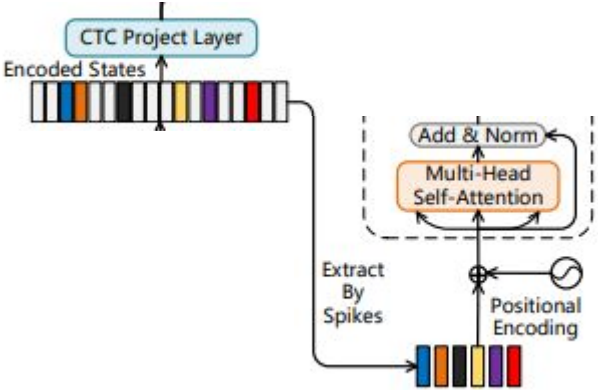


Single step NAT

- Use acoustic representation for semantic modeling.
- Assumption: acoustic representation can also capture language semantics, like word embedding.



[Bai et al. 2020]



[Tian et al. 2020]

Can we find a better acoustic representation to fit the assumption?

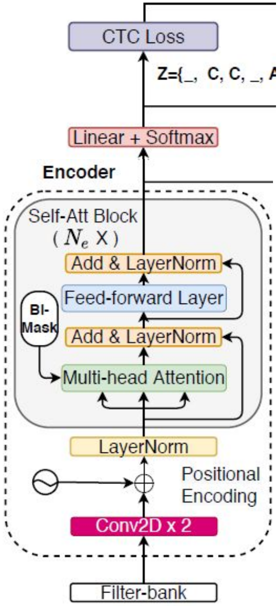
Token-level acoustic embedding!

Proposed CASS-NAT

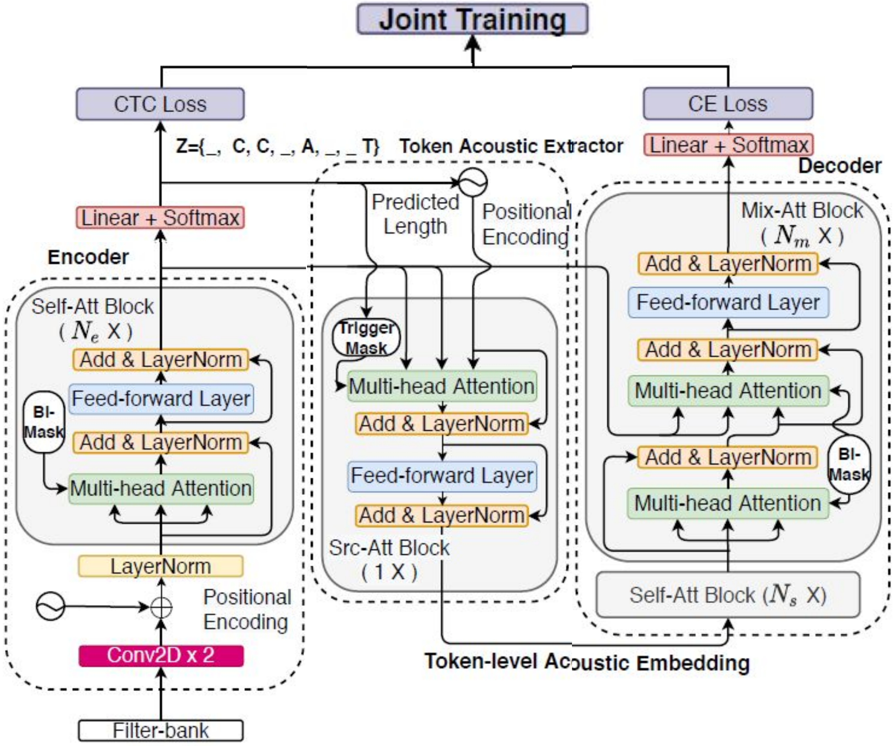
- CTC alignment-based single step non-autoregressive transformer (CASS-NAT)
- The idea is to replace word embedding with the token-level acoustic embedding.
- Encoder: extract high level representation H
- CTC: optimize CTC alignment that offers information for acoustic embedding extraction.
  - Time boundary for each token (trigger mask)
  - Number of tokens for decoder input (NoT)
- Fix mapping rule when obtaining trigger mask
- For example, first index of each token is end acoustic boundary

Alignment: $Z = \{\_, C, \boxed{C}, \_, \boxed{A}, \_, \_, T, \_\}$

Trigger mask: $[0, 0, 1, 1, 1, 0, 0, 0, 0].$



Framework of CASS-NAT



- Token-acoustic extractor:
  - one self-attention block
  - Q: sinusoidal positional embedding with **NoT**
  - K: encoder output H
  - V: encoder output H
  - Mask: **trigger mask** from CTC alignment
- Decoder:
  - self-att block (not considering H)
  - mix-att block (considering H)
- CE: cross entropy loss to optimize the final WER.

Training Criterion

- The CTC alignment Z is introduced as a latent variable.
- Given $X = \{x_1, ..., x_t, ..., x_T\}$ and $Y = \{y_1, ...y_u, ..., y_U\}$, the objective function is:

$$\log P(Y|X) = \log \mathbb{E}_{Z|X}[P(Y|Z,X)], \quad Z \in q.$$

where q is the set of alignments which can be mapped to Y

- To further reduce the computational cost, the maximum approximation is applied:

$$\log P(Y|X) \geq \mathbb{E}_{Z|X}[\log P(Y|Z,X)]$$
$$\approx \max_{Z} \log \prod_{u=1}^{U} P(y_u|z_{t_{u-1}+1:t_u}, x_{1:T})$$

where t_u is the end boundary of token u.

Training Criterion

- The final objective function is:

$$L_{\text{dec}} = \max_Z \log \prod_{u=1}^{U} P(y_u | z_{t_{u-1}+1:t_u}, X)$$

$$L_{\text{joint}} = L_{\text{dec}} + \lambda \cdot \log \sum_{Z \in q} \prod_{i=1}^{T} P(z_i | X)$$

- Viterbi-alignment over CTC output space is used to optimize L_{dec}
- λ is empirically set to 1.
- Semantic modelling is relied on decoder with token-level acoustic embedding.

Inference strategies

- How to obtain CTC alignment during inference
  - Ideally, oracle alignment (forced alignment with ground truth)
  - Best path alignment (BPA)
    - Pro: one step inference, fast Con: alignment is not accurate.
  - Beam search alignment (BSA)
    - Pro: alignment is accurate Con: beam search inference, slow
- How to obtain an accurate alignment in a fast way?
  - Error-based sampling alignment (ESA)
    - Sampling over CTC output space is time consuming.
    - Sampling based on best path alignment is easier.
    - For those frames with low probability, consider other tokens.

Inference strategies - ESA

C(0.90) indicates $P(z_i = C|X) = 0.90$

CTC Output → **CTC Alignments**

	1	2	3	4-
z_1	_ (0.95)	C (0.03)	K (0.01)	...
z_2	C (0.90)	_ (0.07)	Z (0.02)	...
z_3	C (0.50)	_ (0.35)	K (0.10)	...
z_4	_ (0.97)	C (0.01)	K (0.01)	...
z_5	_ (0.61)	A (0.23)	O (0.12)	...
z_6	_ (0.48)	A (0.29)	O (0.10)	...
z_7	I (0.41)	_ (0.30)	A (0.20)	...
z_8	_ (0.95)	T (0.02)	D (0.02)	...
z_9	T (0.95)	_ (0.03)	D (0.01)	...
z_{10}	_ (0.96)	T (0.02)	D (0.01)	...

Best Path Alignment (BPA):

$\{\_, C, C, \_, \_, \_, I, T, \_\}$

Error-based sampling Alignment (ESA):

$\{\_, C, \_, \_, \_, \_, I, T, \_\}$

$\{\_, C, C, \_, A, \_, I, T, \_\}$

$\{\_, C, C, \_, A, \_, \_, T, \_\}$

$\{\_, C, C, \_, A, A, \_, T, \_\}$

......

- If the probability is lower than the threshold (0.7), consider sampling within top2 tokens.
- It is possible to sample alignments with the same number of tokens as oracle alignment.
- Use AT or LM for ranking different alignments based on decoder outputs.

Experiments - Librispeech

- Input and output:
  - 80-dim log-mel filter bank features, computed every 10ms with a 25ms window.
  - Every 3 consecutive frames are concatenated to form a 240-dim input.
  - The output labels consist of 5k word-pieces obtained by SentencePiece [24].
- Model
  - 2 CNNs: 64 filter, kernel size 3, stride 2 => 4x frame rate reduction
  - AT baseline: $N_e = 12, N_d = 6, d_{FF} = 2048, H = 8, d_{MHA} = 512$
  - CASS-NAT:
    - 1-layer token-acoustic extractor
    - Decoder: 3 self-att blocks and 4 mix-attn blocks
- SpecAug, Label smoothing, Encoder initialization

Experiments - Librispeech

A comparison of accuracy and speed of Autoregressive Transformer (AT) and non-AT (NAT) algorithms on Librispeech.

		Type	WER (%) dev-clean	WER (%) dev-other	WER (%) test-clean	WER (%) test-other	RTF test-clean
Without LM							
RETURNN [1]		AT	4.3	12.9	4.4	13.5	-
ESPNet		AT	3.2	8.5	3.6	8.4	-
AT (ours)		AT	3.4	8.5	3.6	8.5	0.562
Imputer [16]		NAT	-	-	4.0	11.1	-
CASS-NAT	BPA	NAT	4.4	10.6	4.5	10.7	0.005
	BSA	NAT	3.9	9.6	3.9	9.6	0.655
	ESA	NAT	3.7	9.2	3.8	9.1	0.011
With LM							
RETURNN [1]		AT	2.6	8.4	2.8	9.3	-
ESPNet [25]		AT	2.3	5.6	2.6	5.7	-
AT (ours)		AT	2.5	5.7	2.7	5.8	-
CASS-NAT	ESA	NAT	3.3	8.0	3.3	8.1	-

- ESA decoding reduces WER significantly compared to both BPA and BSA and has a moderate increase of RTF over BPA.
- When no external LM is used, the proposed CASS-NAT is 51.2x faster than AT in terms of RTF, while has ~6% relative WER reduction.
- When using an external LM, the gap of WER between AT baselines and CASS-NAT is increasing.

Experiments - Librispeech

- To analyze the alignment used for inference, two metrics are used
  - Mismatch rate (MR)
    - Deletion and insertion errors compared to the oracle alignment
    - Substitution errors do not affect token-level acoustic embedding extraction.
  - Length prediction error rate (LPER)
    - Taking the alignment as output and removing blank and repetitions, the ratio of utterances with different length compared to ground truth.
    - The length is equivalent to the number of tokens as the length of decoder input.
- Goal: find out why ESA works well.

Experiments - Librispeech

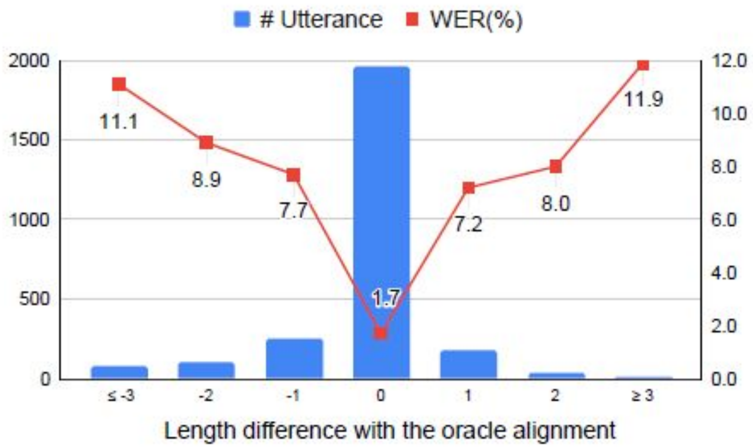
A comparison of different alignment generation methods in CASS-NAT decoding without LM.

Alignment	S	WER (%)		MR (%)		LPER (%)	
		test-clean	test-other	test-clean	test-other	test-clean	test-other
Oracle	n/a	2.3	5.8	n/a	n/a	n/a	n/a
BSA	n/a	3.9	9.6	2.2	5.8	27.9	48.3
BPA	n/a	4.5	10.7	2.1	4.9	31.0	51.8
ESA	10	3.9	9.4	2.9	5.7	26.4	42.8
	50	3.8	9.1	3.1	5.8	25.3	41.9
	100	3.8	9.0	3.0	5.8	25.1	41.8
	300	3.8	9.0	3.1	5.8	25.1	41.9

- With oracle alignment, the lower bound of WER can be 2.3% for test-clean set.
- For ESA, no further gains are observed when the number of sampled alignments is over 50.
- Correct estimation of the decoder input length is more important for NAT.

Experiments - Librispeech

Length prediction error distributions and corresponding WERs with ESA(s=50) decoding on the test-clean dataset.



- The WER can be lowered than 2% for the utterances with correct token number estimation.
- The figure shows the importance of length prediction accuracy on the encoder side again.

Experiments - Aishell1

- Input and output:
  - 80-dim log-mel filter bank features, computed every 10ms with a 25ms window.
  - Every 3 consecutive frames are concatenated to form a 240-dim input.
  - The output labels consist of **4230 Chinese characters** obtained from training set.
- Model
  - 2 CNNs: 64 filter, kernel size 3, stride 2 => 4x frame rate reduction
  - AT baseline: $N_e = 6, N_d = 6, d_{FF} = 2048, H = 8, d_{MHA} = 512$
  - CASS-NAT:
    - 1-layer token-acoustic extractor
    - Decoder: 3 self-att blocks and 4 mix-attn blocks
- SpecAug, **Speed perturbation**, Label smoothing, Encoder initialization

Experiments - Aishell1

A comparison of WERs on Aishell1 with the existing works.

CER(%)	NAT Type	Dev	Test
AT (ours)	n/a	5.5	5.9
Masked-NAT [13]	iterative	6.4	7.1
Insertion-NAT [15]	iterative	6.1	6.7
ST-NAT [18]	single step	6.9	7.7
LASO [17]	single step	5.8	6.4
CASS-NAT (ours)	single step	5.3	5.8

- Our proposed CASS-NAT is better than previous work.
- CASS-NAT is slightly better than AT, which is promising.
- Our framework generalizes well according to the AT baseline.

Conclusions

- This work presented a novel CASS-NAT framework
  - CTC alignment is used as auxiliary information to extract token-level acoustic embedding.
  - The word embedding in AT is replaced with acoustic embedding for parallel generation.
  - Viterbi-alignment is used for training.
  - An error-based sampling alignment is proposed for inference.
- The importance of length prediction for decoder input is shown by analyzing the relationships between different alignments and the oracle alignment.
- We decrease the gap between AT and NAT, and maintain the acceleration for NAT.

References

The number is appeared as the same in the paper.

[1] C. Luscher, E. Beck, K. Irie, et. al. “Rwth asr systems for librispeech: Hybrid vs attention–w/o data augmentation,” Interspeech, pp. 231–235, 2019.

[13] Nanxin Chen, Shinji Watanabe, Jes´us Villalba, and Najim Dehak, “Non-autoregressive transformer automatic speech recognition,” arXiv preprint arXiv:1911.04908, 2019.

[14] Y. Higuchi, S. Watanabe, N. Chen, T. Ogawa, and T. Kobayashi, “Mask ctc: Non-autoregressive end-to-end asr with ctc and mask predict,” Proc. Interspeech 2020, pp. 3655–3659, 2020.

[15] Yuya Fujita, Shinji Watanabe, Motoi Omachi, and Xuankai Chang, “Insertion-based modeling for end-to-end automatic speech recognition,” Proc. Interspeech 2020, pp. 3660–3664, 2020.

[16] W. Chan, C. Saharia, G. Hinton, M. Norouzi, and N. Jaitly, “Imputer: Sequence modelling via imputation and dynamic programming,” in International Conference on Machine Learning. PMLR, 2020, pp. 1403–1413.

[17] Y. Bai, J. Yi, et. al., “Listen attentively, and spell once: Whole sentence generation via a non-autoregressive architecture for low-latency speech recognition,” Proc. Interspeech 2020, pp. 3381–3385.

[18] Z. Tian, J. Yi, J. Tao, Y. Bai, S. Zhang, and Z. Wen, “Spike-triggered non-autoregressive transformer for end-to-end speech recognition,” Proc. Interspeech 2020, pp. 5026–5030.

[24] Taku Kudo and John Richardson, “Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing,” EMNLP 2018, p. 66, 2018.

[25] S. Karita, N. Chen, T. Hayashi, et al., “A comparative study on transformer vs rnn in speech applications,” in ASRU. IEEE, 2019, pp. 449–456.