ADVANCING CTC MODELS FOR BETTER SPEECH ALIGNMENT: A TOPOLOGICAL APPROACH

Zeyu Zhao, Peter Bell

The Centre for Speech Technology Research, University of Edinburgh, United Kingdom

ABSTRACT

Automatic Speech Recognition (ASR) systems often face challenges in alignment quality, particularly with the Connectionist Temporal Classification (CTC) approach, which frequently results in a high number of blank frames, known as the “peaky” issue. In this study, we explore the impact of modifying ASR model topologies on alignment quality without compromising Word Error Rate (WER) performance. Our findings demonstrate that introducing additional states to the CTC topology significantly improves alignment quality and mitigates the peaky issue. Conversely, increasing the minimum traversal frame can degrade alignment quality in our specific settings. These insights emphasise the critical importance of topology design in balancing alignment accuracy and recognition performance in ASR systems.

Index Terms— ASR, CTC, Topology, Alignment

1. INTRODUCTION

Since its inception in 2006, Connectionist Temporal Classification (CTC) has emerged as a cornerstone in the field of End-to-End (E2E) Automatic Speech Recognition (ASR), enabling the training of Neural Network (NN) based ASR models directly from transcribed speech datasets without explicit alignment information [1–5]. This innovative approach models the posterior probability by marginalising over all possible alignments that correspond to a target sequence, leveraging a remarkably simple yet effective one-state topology for each modelling unit (e.g., grapheme, phoneme, or Byte Pair Encoding (BPE)). This simplicity has facilitated widespread adoption in various ASR applications. The foundational one-state topology, along with a shared blank state, was implicit in the original formulation of CTC, despite not being explicitly mentioned [3, 6].

However, despite its significant contributions, CTC is not without its limitations. One notable issue is the “peaky” problem, which means, due to its unique topology, CTC tends to classify most frames as blank labels, leading to sparse and potentially less informative predictions [7]. To address this, in [3], the authors proposed a new loss function by introducing an individual blank label for each modelling unit, aiming at changing the topology and the training behaviour. Furthermore, the nature of CTC’s conditional independence assumption leads it to assign too much probability mass to alignments that are less probable at a sequence-level, resulting in a high entropy in the probability distribution of alignments. This characteristic is suboptimal for tasks requiring precise alignment, such as Viterbi decoding, prompting the exploration of entropy regularization as a potential solution [6].

Whilst for most applications, Word Error Rate (WER) is the most important measure of ASR performance, in many cases the quality of word alignments is also significant [8]. In view of the issues outlined above, this paper investigates the potential of enhancing alignment quality in CTC-like ASR models through modifications to the model topology. Inspired in part by [3,9], we investigate seven more topologies to see if the change of topology can make a difference to overall WER performance and word level alignment quality.

To facilitate a comprehensive and fair comparison between topologies, we employ a Differentiable Weighted Finite-State Transducer (DWFST) framework as the foundational architecture for implementing all models involved in our experiments [10]. This choice allows us to seamlessly integrate the CTC loss function as a specific instance within a broader experimental framework, thereby ensuring that our comparative analysis is both robust and insightful.

2. METHOD

2.1. CTC

Given an input speech X, represented either as a sequence of feature vectors or as a raw waveform, we first process it through an acoustic encoder. This encoder transforms X into a sequence of hidden features or high-level representations. Subsequently, one or more linear layers, culminating in a softmax activation function, are applied. This process yields a probability distribution over each token c in the set of output tokens $\mathbb{C}$ (which includes the blank label) at each time step t, denoted as $p_t(c|X)$, where $t = 1, 2, \dots, T$. With the conditional independence assumption, CTC models the posterior probability $p(Y|X)$ by marginalising all possible alignments between X and the target tokenised sequence Y (e.g., graphemes, phonemes, BPEs),

$$p(Y|X) = \sum_{\pi \in B^{-1}(Y)} \prod_{t=1}^{T} p_t(\pi_t|X). \quad (1)$$

where $B(\pi)$ is the operation of compressing repeated neighbouring tokens and then removing blank labels from an alignment sequence π.

2.2. DWFST Implementation

We can easily implement CTC with DWFST [9,10], and more generally the posterior $p(Y|X)$ is modelled as

$$p(Y|X) = \frac{\sum_{\pi \in \Pi(Y;\boldsymbol{T}\circ\boldsymbol{L})} p(\pi|X)}{\sum_{Y'} \sum_{\pi \in \Pi(Y';\boldsymbol{T}\circ\boldsymbol{L})} p(\pi|X)}, \quad (2)$$

where $\boldsymbol{T}$ and $\boldsymbol{L}$ denote the **Topology** and the Lexicon Finite-State Transducer (FST), and $\Pi(Y;\boldsymbol{T}\circ\boldsymbol{L})$ represents the set of all the token sequences ("paths"), whose corresponding output word sequence is Y, in the composition resulting FST, $\boldsymbol{T}\circ\boldsymbol{L}$ [11]. We also apply the conditional independence assumption and thus

$$p(\pi|X) = \prod_{t=1}^{T} p_t(\pi_t|X). \quad (3)$$

We compute the numerator in eq. (2) as

$$\sum_{\pi \in \Pi(Y;\boldsymbol{T}\circ\boldsymbol{L})} p(\pi|X) = \mathrm{TotalScore}(\boldsymbol{E}\circ\boldsymbol{S}^{\mathrm{trn}}), \quad (4)$$

where TotalScore denotes the total score operation, and $\boldsymbol{E}$ is the Emission FST constructed from the model's outputs. $\boldsymbol{S}^{\mathrm{trn}}$ is the training graph derived from the target word sequence Y,

$$\boldsymbol{S}^{\mathrm{trn}} = \boldsymbol{T}\circ(\boldsymbol{L}\circ\boldsymbol{Y}), \quad (5)$$

where $\boldsymbol{Y}$ is a linear FST with the sequence Y as input and output labels.

The previous work [3] has underscored the importance of normalisation (the denominator term in eq. (2)) for achieving training convergence. Due to the limitation of computation resources, we approximate the denominator by taking into account the paths that are acceptable for $\boldsymbol{T}$, so we have

$$\sum_{Y'} \sum_{\pi \in \Pi(Y';\boldsymbol{T}\circ\boldsymbol{L})} p(\pi|X) \approx \mathrm{TotalScore}(\boldsymbol{E}\circ\boldsymbol{T}). \quad (6)$$

Note that when $\boldsymbol{T}$ is set as the CTC topology, as we will see in Figure 1a, any arbitrary token sequence can be accepted , so the denominator is always one, resulting in

$$p(Y|X) = \sum_{\pi \in \Pi(Y;\boldsymbol{T}_{\mathrm{CTC}}\circ\boldsymbol{L})} p(\pi|X), \quad (7)$$

which is equivalent to the CTC loss function as eq. (1). However, generally, for most topologies, not all the paths in $\boldsymbol{E}$ can be accepted by $\boldsymbol{T}$, which leads to a denominator not equal to one.

2.3. Topology

To investigate various topologies comprehensively , we introduce seven additional topologies in this paper, as shown in Figure 1. To systematically address these topologies, we name them using the pattern Sx-Ty(★*n). Here, Sx-Ty indicates there are x states for each modelling unit with a minimum traversal frame of y, and an added ★ denotes an additional self-loop compared to the version without any ★. Thus, in our naming system, the CTC topology is referred to as S1-T1.

Although not explicitly mentioned in the foundational paper [1], CTC inherently employs a one-state topology, as illustrated in Figure 1a. We identify several limitations associated with this topology. Firstly, modelling each unit with a single state restricts CTC's modelling capabilities. This limitation arises because acoustical variations within a modelling unit are challenging to represent accurately with a singular state, leading to potential misclassifications. A direct consequence of this limitation is the "peaky" issue [7], where CTC predominantly assigns frames as blank labels, except in cases of high classification confidence. While CTC is effective for optimising overall Word Error Rate (WER) performance [2], it falls short in ensuring high-quality alignment [6, 12]. This aspect is crucial for applications requiring precise temporal alignment between the audio input and the transcribed text.

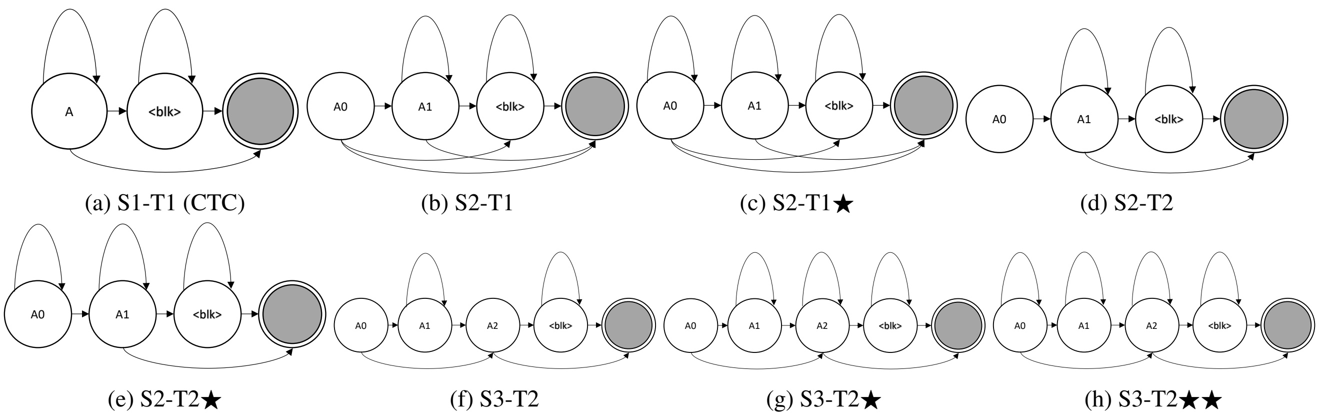
Inspired by [3], we introduce S2-T1 (Figure 1b), where a second state with a self-loop is assigned to each modelling unit, but there is no self-loop on the first state. A natural variant of S2-T1 is S2-T1★, where a self-loop is added to the first state. Compared to the traditional CTC topology, S2-T1★ and S2-T1 can be regarded as enhanced two-state variants. By incorporating a second state, we expect S2-T1 and S2-T1★ to surpass the original CTC in terms of modelling capabilities and achieve better alignment quality. Note that both variants, S2-T1 and S2-T1★, maintain a minimum traversal frame of one, similar to CTC, meaning at least one frame must be absorbed to output one modelling unit.

To further investigate the influence of the minimum traversal frame, we introduce five additional topologies where at least two frames must be absorbed to output one modelling unit: S2-T2, S2-T2★, S3-T2, S3-T2★, and S3-T2★★, as shown in Figures 1d to 1h, respectively. By comparing the S2-T2* and S3-T2* topologies, we can also examine whether adding more states affects modelling ability and alignment quality. Finally, similar to the CTC topology (S1-T1 in Figure 1a), a blank label is introduced and kept identical across all the topologies investigated in this paper.

2.4. Decoding and Alignment

We construct the decoding graph as

$$\boldsymbol{S}^{\mathrm{dec}} = \boldsymbol{T}\circ(\boldsymbol{L}\circ\boldsymbol{G}), \quad (8)$$



(a) S1-T1 (CTC) (b) S2-T1 (c) S2-T1★ (d) S2-T2

(e) S2-T2★ (f) S3-T2 (g) S3-T2★ (h) S3-T2★★

Fig. 1: The topologies investigated in this paper, where <blk> denotes the shared optional blank label. Note that in some cases the blank label is unskippable in CTC but we omit it for simplicity. Sx-Ty means there are x states for each phone, the minimum traversal frame is y, and one ★ means one more self-loop is added.

where $\boldsymbol{G}$ and $\boldsymbol{L}$ are Grammar and Lexicon FST, respectively, and the former is normally obtained from an n-gram language model [13]. For simplicity, operations such as determinisation and minimisation [11] have been omitted in the equation. With outputs from models and a decoding graph, we apply the Viterbi decoder [14, 15] to get the best word sequence

$$W^* = \arg\max_W [\log p(W) + \alpha \max_{\pi \in \Pi(W; \boldsymbol{S}^{\text{dec}})} \log p(\pi|X)], \tag{9}$$

where $p(W)$ is determined by the language model, which is encoded in $\boldsymbol{S}^{\text{dec}}$ as transition weights, and α is the acoustic weight.

To obtain the alignment, we substitute $\boldsymbol{G}$ in eq. (8) with $\boldsymbol{W}^*$ which is the linear FST constructed from the decoded word sequence W^* or the ground thruth Y. We then determine the optimal path using the Viterbi decoder, taking into account the output posterior probabilities. For acquiring word-level alignments, we identify the start timestamp for each word as the timestamp of the transition where the word is the output label. Similarly, the end timestamp for each word is derived from the timestamp of the last transition that has a non-blank input label.

3. EXPERIMENTS

3.1. Settings

In our study, we conduct experiments using the LibriSpeech dataset by fine-tuning the wav2vec 2.0 model [16], which was initially pre-trained on 60,000 hours of unlabelled speech data[1]. Specifically, we fine-tune all 24 transformer layers of the encoder, along with two linear layers that employ a log-softmax activation function. The feature extractor component of the wav2vec 2.0 model remains unchanged. Our models use characters as the modeling units.

For the optimisation process, we employ two different optimisers: the Adam optimiser, with a learning rate of 10^{-4}, for the encoder layers, and the Adadelta optimiser, with a learning rate of 0.9, for the linear layers. The training process is halted when no further reduction in loss is observed for two consecutive epochs on the development set, dev-clean.

Throughout our experimental setup, we utilise Kaldi [17] for data preparation and PyTorch [18] for neural network training, with k2[2] serving as the backend for the DWFST framework. For decoding and alignment generation, we employ the Viterbi decoder ("decode-faster" in Kaldi) and k2, respectively.

To promote the reproducibility of our findings, we have made the code for our experiments publicly available[3]. Furthermore, we use the alignment results[4] obtained from the Montreal Forced Aligner [19] as a benchmark to evaluate the alignment quality of the various topologies under investigation.

3.2. Evaluation Metrics

Primarily, we use the Word Error Rate (WER) metric to evaluate the performance of ASR models featuring various topologies.

To assess alignment quality, we apply two different metrics depending on the circumstances discussed in this paper. First, to measure the alignment quality given the ground truth Y, we use the Time Stamp Error (TSE) [20], which is defined as

$$\text{TSE} = \frac{\sum_w |r_s - h_s| + |r_e - h_e|}{N_w}, \tag{10}$$

where $\sum_w$ is conducted over all the words in the evaluation set, and r_s and r_e denote the reference starting and ending

[1] `WAV2VEC2_LARGE_LV60K` in `torchaudio.pipelines`

[2] https://github.com/k2-fsa/k2

[3] https://github.com/ZhaoZeyu1995/BenNevis

[4] https://github.com/CorentinJ/librispeech-alignments

timestamps, respectively, and the same for the hypothesis h_s and h_e.

In real applications, the ground truth is typically unavailable, making it more valuable to evaluate the quality of alignments based on the hypotheses recognised by the model itself. To this end, we employ Alignment Accuracy (ACC) [6], which is defined as

$$\mathrm{ACC}(\tau) = \frac{\sum_w \mathbf{1}(r_s - \tau \leq h_s \cap h_e \leq r_e + \tau)}{N_w}, \tag{11}$$

where N_w represents the total number of words in the evaluation dataset. The function $\mathbf{1}(*)$ yields one if the specified condition is true, and zero otherwise. The summation $\sum_w$ is performed over words that are correctly recognised by the ASR models. It is important to note that the parameter τ serves as a measure of tolerance for time discrepancies in alignment accuracy assessments.

Upon obtaining the alignment represented as a linear FST, we have precise knowledge of its input label sequence. To verify whether the "peaky" issue can be mitigated by modifying the topology, we calculate the Blank Ratio R_b as

$$R_b = \frac{\sum_u N_b}{\sum_u N}, \tag{12}$$

where N_b denotes the number of occurrences of the blank state within an evaluation utterance, and N represents the total number of frames in that utterance. The summation $\sum_u$ is executed across all evaluation utterances. This evaluation metric allows us to quantify the proportion of frames that are classified as blank states by our models.

3.3. Results and Analysis

3.3.1. WER

Table 1: The WER(%) performance of different topologies, with or without the official 3-gram language model (tgmsall)

	test-clean		test-other	
	noLM	tgsmall	noLM	tgsmall
S1-T1	2.5	2.5	6.0	5.9
S2-T1	2.7	2.6	6.4	5.8
S2-T1★	2.8	2.6	6.4	5.7
S2-T2★	2.6	2.6	6.2	5.7
S2-T2	2.5	2.5	5.9	5.6
S3-T2	2.5	2.5	5.9	5.7
S3-T2★	2.6	2.5	5.9	5.4
S3-T2★★	2.5	2.5	6.0	5.8

Table 1 displays the overall Word Error Rate (WER) performance of the models across various topologies. The WER performance across the models under consideration is notably similar, with the S3-T2★ model achieving marginally better results than the others when using the official 3-gram language model 'tgsmall'. This trivial WER performance difference among different topologies can be attributed to the wav2vec 2.0 model's pre-training on an extensive corpus of unlabelled speech data. Previous research has shown that, in the absence of such pre-training, the performance disparities between different topologies are more pronounced [9].

It is encouraging to observe that the other seven topologies, compared to S1-T1 (CTC), do not significantly compromise WER performance. This outcome is crucial, as a substantial degradation in WER would render these topologies less appealing compared to the standard CTC approach. Maintaining competitive WER performance is essential, as we aim to enhance alignment quality without sacrificing accuracy in speech recognition.

Overall, the findings suggest that our proposed topologies can maintain robust WER performance while potentially offering improvements in alignment quality. This balance is vital for advancing ASR systems that require both high accuracy and precise temporal alignment.

3.3.2. Time Stamp Error

Our results indicate that different topologies, when paired with the pre-trained wav2vec 2.0 model, exhibit nearly identical WER performance on both the test-clean and test-other datasets. This consistency underscores the robustness of the wav2vec 2.0 model, which maintains high accuracy across various topology configurations due to its extensive pre-training on unlabelled speech data.

Having established stable WER performance across topologies, we next compare their alignment quality. To this end, we align the model outputs with respect to the ground truth and compute the Time Stamp Error (TSE), as shown in Table 2. The TSE metric provides a quantitative measure of how accurately the predicted timestamps match the reference timestamps, allowing us to evaluate which topologies offer superior alignment accuracy. This comparison is crucial for identifying configurations that balance both high WER performance and precise alignment, essential for applications requiring accurate temporal synchronization.

Surprisingly, even though the WER performance of the models with different topologies is quite similar, there are clear differences in their alignment quality. Based on the TSE performance, we can classify the topologies into three distinct groups.

The first group includes only S1-T1 (CTC), which serves as the baseline of our experiments. The second group consists of S2-T1 and S2-T1★, which show better TSE performance than the baseline. This improvement suggests that adding an additional state to the original CTC topology enhances the modelling power. Comparing S2-T1 and S2-T1★, where the only difference is the self-loop on the first state,

Table 2: The Time Stamp Error (in msec) of different topologies on test-clean and test-other.

	test-clean	test-other
S1-T1 (CTC)	97	98
S2-T1	79	81
S2-T1★	78	80
S2-T2★	131	135
S2-T2	131	134
S3-T2	131	137
S3-T2★	130	134
S3-T2★	132	137

reveals that the self-loop provides a minor improvement in alignment quality.

The third group contains the remaining five topologies: S2-T2, S2-T2★, S3-T2, S3-T2★, and S3-T2★★. These topologies exhibit similar TSE performance, all worse than the baseline CTC. Interestingly, despite S3-T2★ achieving the best WER with the language model (as shown in Table 1), it delivers low-quality alignments (as shown in Table 2). This indicates that a model with good WER performance does not necessarily guarantee high-quality alignments.

One possible explanation for the poor alignment quality of the Sx-T2* topologies lies in the average ratio between the number of output frames and the number of modelling units per utterance. This ratio is around 3.6 for both the test-clean and test-other datasets, meaning that, on average, there are only about 3.6 frames to be absorbed for outputting one modelling unit (characters in our settings). The Sx-T2* topologies may find it less flexible to learn proper alignments between output frame sequences and the target character sequences compared to the Sx-T1* topologies, thus hampering their alignment quality.

3.3.3. Alignment Accuracy

In real-world applications, ground truth data is typically unavailable, so we also assess alignment quality based on the hypotheses generated by our models. For this condition, we use the Alignment Accuracy (ACC) metric outlined in eq. (11). The results are presented in Tables 3 and 4.

We align the model outputs to the hypotheses obtained using the 'tgsmall' language model. This evaluation helps us determine which topologies maintain competitive WER performance while also producing high-quality alignments based on their own predictions, crucial for practical applications where accuracy and alignment precision are essential.

Similar to the discussion in section 3.3.2, we can group all eight topologies into three categories based on their alignment accuracy performance. The alignment accuracy results exhibit a trend similar to that observed in Table 2. Specifically, when comparing alignment accuracy with different tolerances (denoted as τ), the S2-T1★ and S2-T1 topologies consistently outperform the CTC, especially with a low tolerance value.

Table 3: The alignment accuracy $\mathrm{ACC}(\tau)$ (%) of various topologies with different τ values (msec) on test-clean

	τ				
	10	20	30	40	50
S1-T1 (CTC)	78	86	91	95	96
S2-T1	88	94	96	97	97
S2-T1★	89	95	97	98	98
S2-T2★	62	68	74	79	83
S2-T2	64	71	76	81	85
S3-T2	61	68	73	78	82
S3-T2★	65	71	77	82	86
S3-T2★★	63	69	75	80	84

Table 4: The alignment accuracy $\mathrm{ACC}(\tau)$ (%) of different topologies with different τ values (msec) on test-other

	τ				
	10	20	30	40	50
S1-T1 (CTC)	75	82	90	91	93
S2-T1	82	89	92	93	94
S2-T1★	83	91	93	93	94
S2-T2★	54	61	67	72	76
S2-T2	56	63	69	74	78
S3-T2	53	59	65	71	75
S3-T2★	57	64	69	75	79
S3-T2★★	54	61	67	73	77

For instance, with τ = 10, the S2-T1★ topology achieves a 10.6% relative improvement in alignment accuracy over the baseline S1-T1 (CTC). This improvement highlights the limitations of the CTC topology, which employs a single state with a self-loop for each modelling unit, significantly restricting its modelling capabilities as discussed in section 2.3.

Previous research [3] has demonstrated that incorporating an additional state without a self-loop, as in the S2-T1 topology, can enhance WER performance and convergence speed in models that have not been pre-trained. However, our experiments with pre-trained models reveal minimal differences in WER performance among the different topologies. Despite this, the S2-T1 topology still achieves the second-best alignment quality among the eight topologies.

The consistent performance of the S2-T1 and S2-T1★ topologies suggests that adding an extra state can improve alignment accuracy, even in pre-trained models. This finding is crucial for applications requiring precise temporal alignment, where the baseline CTC topology may fall short. Thus, our results indicate that topologies with an additional state, such as S2-T1 and S2-T1★, offer a better balance between maintaining WER performance and enhancing alignment quality.

On the other hand, the remaining five topologies (S2-T2, S2-T2★, S3-T2, S3-T2★, and S3-T2★) show worse alignment accuracy performance than the baseline CTC. These topologies, which all have a minimum traversal frame of two, seem less effective in learning proper alignments between output frame sequences and the target character sequences. This could be due to the reduced flexibility in adapting to the alignment constraints imposed by the longer minimum traversal frame. Despite some of these topologies achieving competitive WER performance, such as S3-T2★, their alignment quality remains subpar.

The consistent underperformance of the Sx-T2 topologies in alignment accuracy suggests that the increased traversal frame introduces challenges that outweigh the benefits of additional states. This is particularly evident in our settings, where the average ratio of output frames to modelling units per utterance is around 3.6. The reduced number of frames available for each modelling unit likely hampers the ability of these topologies to achieve high-quality alignments.

3.3.4. Blank Ratio

Table 5 presents the blank ratios observed in different topologies when evaluated on the test-clean and test-other datasets. It is important to note that the 3-gram language model was not applied when generating the alignments for calculating the blank ratio. This approach allows us to more closely evaluate the intrinsic behaviour of the topologies without the influence of the language model.

Table 5: The Blank Ratio (%) of different topologies on test-clean and test-other.

	test-clean	test-other
S1-T1 (CTC)	45.8	48.5
S2-T1	6.6	7.7
S2-T1★	6.4	7.6
S2-T2★	5.5	6.6
S2-T2	5.1	6.2
S3-T2	4.6	5.7
S3-T2★	4.9	6.0
S3-T2★★	5.0	6.0

This comparison highlights the persistent peaky issue associated with the CTC topology, where a significant majority of frames are classified as blank labels in the best alignment paths. This phenomenon underscores a fundamental challenge in CTC's approach to speech recognition: its simplicity, while advantageous in certain contexts, may lead to excessive caution in frame classification.

In contrast, the S2-T1 and S2-T1★ topologies, which introduce an additional state, demonstrate a tendency to label fewer frames as blanks. This adjustment suggests an enhanced capability to discern non-blank elements within speech, thereby addressing one of the key limitations of the CTC topology.

An interesting observation arises when comparing the blank ratios between the test-clean and test-other datasets across all topologies. There is a noticeable increase in blank ratios for the more challenging test-other dataset. This increase likely reflects the models' struggle with the complexity and variability inherent in test-other, leading to a higher degree of uncertainty in frame classification. The difficulty of test-other is particularly pronounced for the CTC topology, which relies on a single state for each modelling unit. In scenarios where a frame's classification is not immediately clear, CTC's default is to opt for a blank label. While this choice reduces the risk of misclassification, it also limits the model's ability to capture detailed timing information.

Conversely, the S2-T1 and S2-T1★ topologies benefit from their enriched structure, featuring an additional state but maintaining the same minimum traversal frame of one. This provides them with more options for assigning frames to non-blank states. This capability not only mitigates the "peaky" issue but also enhances the model's reliability and accuracy in speech alignment. By affording more flexibility in frame classification, S2-T1 and S2-T1★ demonstrate a significant advancement in addressing the limitations of traditional CTC, offering a promising avenue for improving alignment quality in speech recognition systems.

Additionally, we observe that the S3-T2* topologies, which include three states, tend to have a lower blank ratio compared to the S2-T2* topologies. Furthermore, topologies with a larger minimum traversal frame tend to have a lower blank ratio. This is reasonable, as they need to absorb more frames and classify them as non-blank labels to align output sequences to target sequences effectively. Overall, these findings underscore the importance of topology design in speech recognition systems, highlighting how certain modifications can significantly enhance both alignment quality and model robustness.

4. CONCLUSION

Our study reveals that altering the topology can significantly enhance alignment quality without compromising WER performance. By adding an additional state to the CTC topology, as seen in the S2-T1 and S2-T1★ models, we improved alignment quality and reduced the peaky issue common in CTC. However, modifying the minimum traversal frame resulted in poorer alignment quality compared to CTC under our experimental conditions. These findings highlight the critical role of topology design in balancing alignment accuracy and recognition performance in speech recognition systems.

5. REFERENCES

[1] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, “Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks,” in *Proceedings of the 23rd International Conference on Machine Learning*, 2006, pp. 369–376.

[2] A. Graves and N. Jaitly, “Towards End-To-End Speech Recognition with Recurrent Neural Networks,” in *Proceedings of the 31st International Conference on Machine Learning*. PMLR, Jun. 2014, pp. 1764–1772.

[3] Z. Zhao and P. Bell, “Investigating Sequence-Level Normalisation For CTC-Like End-to-End ASR,” in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 7792–7796.

[4] T. Raissi, W. Zhou, S. Berger, R. Schlüter, and H. Ney, “HMM vs. CTC for Automatic Speech Recognition: Comparison Based on Full-Sum Training from Scratch,” in *2022 IEEE Spoken Language Technology Workshop (SLT)*, Jan. 2023, pp. 287–294.

[5] A. Laptev, S. Majumdar, and B. Ginsburg, “CTC Variations Through New WFST Topologies,” in *Interspeech 2022*. ISCA, Sep. 2022, pp. 1041–1045.

[6] E. Variani, K. Wu, D. Rybach, C. Allauzen, and M. Riley, “Alignment Entropy Regularization,” in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Jun. 2023, pp. 1–5.

[7] A. Zeyer, R. Schlüter, and H. Ney, “Why does CTC result in peaky behavior?” *CoRR*, vol. abs/2105.14849, 2021. [Online]. Available: https://arxiv.org/abs/2105.14849

[8] B. Lin and L. Wang, “Learning Acoustic Frame Labeling for Phoneme Segmentation with Regularized Attention Mechanism,” in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 7882–7886.

[9] Z. Zhao and P. Bell, “Regarding Topology and Variant Frame Rates for Differentiable WFST-based End-to-End ASR,” in *INTERSPEECH 2023*. ISCA, Aug. 2023, pp. 4903–4907.

[10] A. Hannun, V. Pratap, J. Kahn, and W.-N. Hsu, “Differentiable Weighted Finite-State Transducers,” *arXiv:2010.01003 [cs, stat]*, Oct. 2020.

[11] M. Mohri, F. Pereira, and M. Riley, “Speech Recognition with Weighted Finite-State Transducers,” in *Springer Handbook of Speech Processing*, J. Benesty, M. M. Sondhi, and Y. A. Huang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 559–584.

[12] O. Chang, D. Hwang, and O. Siohan, “Revisiting the Entropy Semiring for Neural Speech Recognition,” in *The Eleventh International Conference on Learning Representations*, Feb. 2023.

[13] Y. Miao, M. Gowayyed, and F. Metze, “EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding,” in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Dec. 2015, pp. 167–174.

[14] D. Povey, M. Hannemann, G. Boulianne, L. Burget, A. Ghoshal, M. Janda, M. Karafiát, S. Kombrink, P. Motlíček, Y. Qian, K. Riedhammer, K. Veselý, and N. T. Vu, “Generating exact lattices in the WFST framework,” in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2012, pp. 4213–4216.

[15] L. R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition,” *PROCEEDINGS OF THE IEEE*, vol. 77, no. 2, p. 30, 1989.

[16] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “Wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations,” in *Advances in Neural Information Processing Systems*, vol. 33. Curran Associates, Inc., 2020, pp. 12 449–12 460.

[17] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, and P. Schwarz, “The Kaldi speech recognition toolkit,” in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, no. CONF. IEEE Signal Processing Society, 2011.

[18] A. Paszke, S. Gross *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.

[19] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, “Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi,” in *Proc. Interspeech 2017*, 2017, pp. 498–502.

[20] X. Zhang, V. Manohar, D. Zhang, F. Zhang, Y. Shi, N. Singhal, J. Chan, F. Peng, Y. Saraf, and M. Seltzer, “On Lattice-Free Boosted MMI Training of HMM and CTC-Based Full-Context ASR Models,” in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. Cartagena, Colombia: IEEE, Dec. 2021, pp. 1026–1033.