Supplementary Materials for Self-Attention of SATs

Shu-wen Yang[1], *Andy T. Liu*[12], *Hung-yi Lee*[12]

[1]College of Electrical Engineering and Computer Science, National Taiwan University
[2]Graduate Institute of Communication Engineering, National Taiwan University
{r08944041, r07942089, hungyilee}@ntu.edu.tw

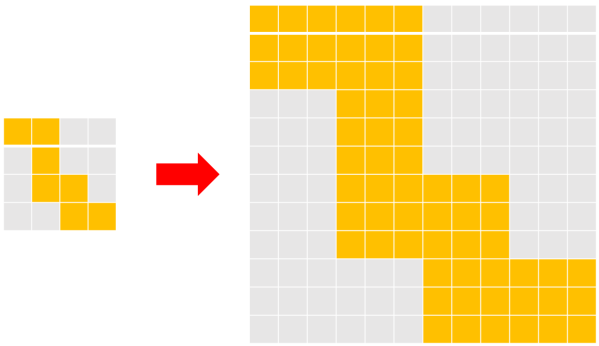

Figure 1: *Up-sample an attention map by $R_{factor} = 3$. The left image is an attention map on down-sampled representations; the right image is the map after up-sampled.*

1. Notations for Up-sampling

We redefine some notations for better clarification. The redefined notations are compatible with the original paper directly.

Raw features are first down-sampled by R_{factor} and then fed into the SAT. We denote $T = \widetilde{T} \cdot R_{factor}$ as the sequence length before down-sampling and $\widetilde{T}$ is the one after down-sampling. We use $\sim$ to denote representations or attentions inside the SAT, which is in length of $\widetilde{T}$. Given a sequence of vectors $\widetilde{\boldsymbol{x}} = \widetilde{x}_1, ..., \widetilde{x}_{\widetilde{T}} \in \mathbb{R}^d$, we denote $\widetilde{A}_u^h \in \mathbb{R}^{\widetilde{T} \times \widetilde{T}}$ as attention weights for all query-key pairs of a head h when propagating an utterance u. Hence, $\widetilde{A}_u^h[q, k] \in \mathbb{R}$ is the attention weight of $\widetilde{x}_q$ attending to $\widetilde{x}_k$. We use q for timestamp of query; k for timestamp of key, where $1 \leq q, k \leq \widetilde{T}$. As a result, $\widetilde{A}_u^h[q] \in \mathbb{R}^{\widetilde{T}}$ is the attention distribution formed by $\widetilde{x}_q$, which is a row if we view $\widetilde{A}_u^h$ as a map. When mentioning the representations of a L-layer SAT, we denote $\widetilde{\boldsymbol{x}}^{\boldsymbol{l}} = \widetilde{x}_1^l, ..., \widetilde{x}_{\widetilde{T}}^l \in \mathbb{R}^d$ as the representations of a given layer $1 \leq l \leq L$.

Since we study attentions by aligning them to phoneme labels originally marked on raw features in length of T, instead of down-sampling phoneme labels we up-sample representations and attention maps. For representations in layer $1 \leq l \leq L$, each $\widetilde{x}_t^l$ is duplicated by R_{factor}, forming a new sequence $\boldsymbol{x}^{\boldsymbol{l}} = x_1^l, ..., x_T^l$. Since raw features are originally in length of T, there is no need for duplication and we use $\boldsymbol{x}^{\boldsymbol{0}}$ to denote raw features. For attention maps, we duplicate attention weights as illustrated in Fig 1. Since the duplication makes the up-sampled map un-normalized at each row, which should be an attention distribution, we re-normalize each row in the map. The up-sampled and re-normalized attention map is denoted as $A_u^h \in \mathbb{R}^{T \times T}$. The A_u^h used in the original paper in fact refers to the up-sampled and re-normalized map, except for section 4.

2. Phoneme Segmentation

2.1. Algorithm

We provide the detailed segmentation algorithm used in our experiments, which is composed of two parts: (1) similarity matrix (2) boundary extraction. To generalize to different features, we defined $\boldsymbol{v} = v_1, ..., v_T \in \mathbb{R}^d$ as a sequence of frames prepared for segmentation, where d is the frame dimension. Features can be either raw features (MFCC or Mel-scale spectrogram), up-sampled representations or up-sampled attention maps of a head h. They take x_t^0, x_t^l and $A_u^h[t]$ as v_t, respectively, where $1 \leq t \leq T$.

2.1.1. Similarity matrices

We compute the kernel-gram-matrix (KGM) [1] on $\boldsymbol{v}$:

$$K[i, j] = exp(-\frac{|| v_i - v_j ||}{\alpha}) \tag{1}$$

where $1 \leq i, j \leq T$ and $||.||$ is 2-norm. $K \in \mathbb{R}^{T \times T}$ and $K[0, 0]$ starts from the upper-left corner. The larger α brings similar frames closer, and makes KGMs more visually block diagonal after fine-tuned. Fig 2 show examples of KGM on different features. Since different features have different distance between frames inside the same phoneme interval, α is not comparable across features.

2.1.2. Boundary extraction

We filter out activation far away from main diagonal and binarize the KGM to extract a phoneme neighborhood for each row, like the yellow blocks in Fig 2(c), which we call filtered KGM $K_f \in \mathbb{R}^{T \times T}$:

$$\epsilon_i = \beta \cdot \frac{1}{T} \sum_{j=1}^{T} K[i, j] \tag{2}$$

$$K_f[i, j] = \begin{cases} \prod_{k=i}^{j} \mathbb{I}_{K[i,k] > \epsilon_i} & \text{if } i \leq j \\ \prod_{k=j}^{i} \mathbb{I}_{K[i,k] > \epsilon_i} & \text{otherwise} \end{cases} \tag{3}$$

where β is a parameter for adjusting the neighborhood threshold. We plot a scoring curve $S \in \mathbb{R}^n$ from K_f, the white curve in Fig 2(d), where a local maximum represents the center of a phoneme interval; a local minimum represents a phoneme boundary, by summing up binary values of K_f in the direction of *anti*-diagonal for each timestamp t:

$$S[t] = \sum_{k=-T}^{T} K_f[t - k, t + k] \tag{4}$$

Finally, by detecting local minimums on S we extract phoneme boundaries, as shown by blue, red and purple (overlay from blue and red) lines in Fig 2(c). The algorithm differs from

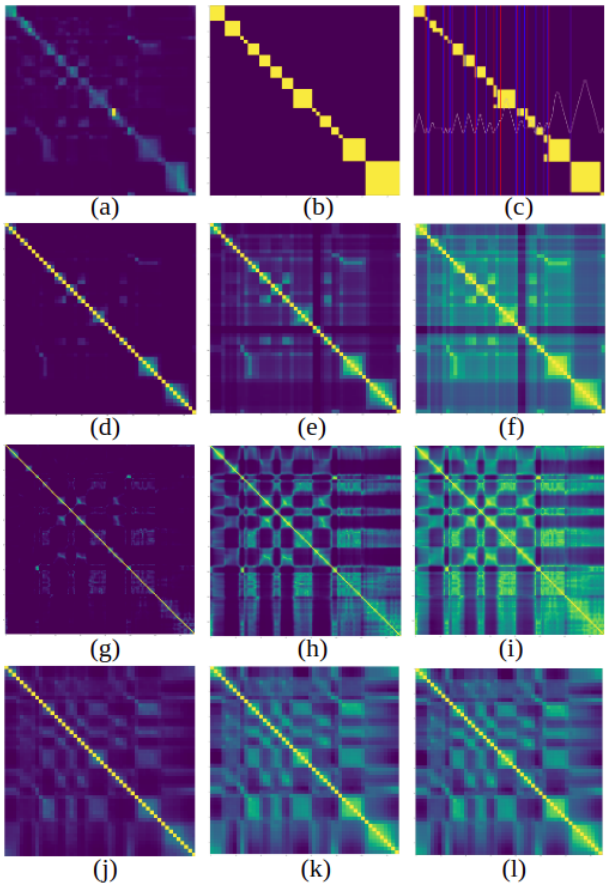

Figure 2: *All images are plotted with the same utterance. (a) is a block diagonal attention map of M3 (head ID 125). (b) is the block diagonal map from true boundaries. (c) is the result of boundary extraction algorithm. Yellow: filtered KGM from (a); White: scoring curve S; Blue: true boundaries; Red: predicted boundaries; Purple: precise alignments. (d)(e)(f) are KGMs on (a) with* $\alpha = 0.03, 0.08, 0.6$*, and (e) is acquired by fine-tuning* α *and* β *with R-value 79.99. (g)(h)(i) are KGMs on MFCC with* $\alpha = 0.5, 2, 5$*, and (h) is acquired by fine-tuning with R-value 76.68. (j)(k)(l) are KGMs on SAT representation (*$\boldsymbol{x}^{\mathbf{11}}$ *of M3) with* $\alpha = 10, 500, 2000$*, and (k) is acquired by fine-tuning with R-value 77.53.*

[1] in boundary-extraction part, which is composed of equations 2, 3, 4, and we use only one adjustable parameter β in contrast to two in [1].

2.2. Evaluation

2.2.1. Hit counting

[2] pointed out one can reuse predicted boundaries for several hits and changes in performance as large as 5%, since typically there are lots of true boundaries gathering together closely. It suggested to constrain the search region for hits as the solution. Since this subtle implementation detail is typically not mentioned in literature [1, 3, 4, 5, 6], we do not directly compare with previous works, but compare with baseline feature MFCC. We follow the hit suggestion in [2] for both R-value and precision-recall curve.

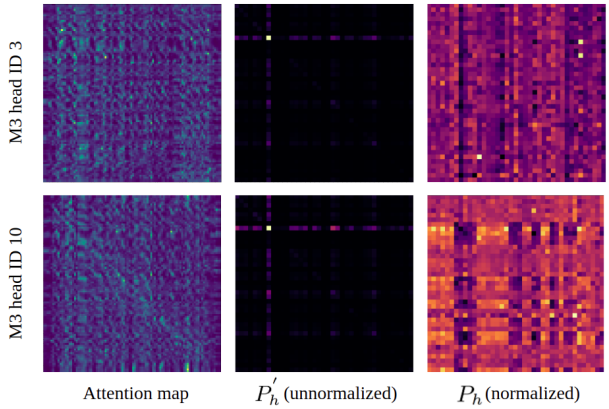

Figure 3: *Differences between normalized/unnormalized PRMs.*

2.2.2. Precision-recall

α and β are first fine-tuned to achieve the best R-value performance, and then we slightly adjust β to observe the trade-off. The larger β represents the stricter neighborhood threshold and results in more predicted boundaries, while the smaller β represents the opposite.

3. Phoneme Relation Map

3.1. Baseline relation distribution

To normalize the effect of dominating phoneme relations, we define the relation distribution of a speech corpus U as:

$$P_U[m,n] = \mathop{\mathbb{E}}_{u\sim U}\left[\sum_{q=1}^{T}\sum_{k=1}^{T}\mathbb{I}_{y_q=Y_m}\cdot\mathbb{I}_{y_k=Y_n}\cdot\frac{1}{T^2}\right] \quad (5)$$

which is equivalent to the relation distribution of uniform attention. Fig 3 show normalized/unnormalized PRMs of two heads. Without normalization, phoneme relations involving silence dominate and hurt the interpretability.

4. References

[1] S. Bhati, S. Nayak, and K. Murty, *Unsupervised Segmentation of Speech Signals Using Kernel-Gram Matrices*, 04 2018, pp. 139–149.

[2] O. Räsänen, U. Laine, and T. Altosaar, "An improved speech segmentation quality measure: The r-value," 01 2009, pp. 1851–1854.

[3] S. Bhati, S. Nayak, K. Murty, and N. Dehak, "Unsupervised acoustic segmentation and clustering using siamese network embeddings," 09 2019, pp. 2668–2672.

[4] A. Stan, C. Valentini-Botinhao, B. Orza, and M. Giurgiu, "Blind speech segmentation using spectrogram image-based features and mel cepstral coefficients," in *2016 IEEE Spoken Language Technology Workshop (SLT)*, 2016, pp. 597–602.

[5] I. Mporas, T. Ganchev, and N. Fakotakis, "Phonetic segmentation using multiple speech features," *International Journal of Speech Technology*, vol. 11, pp. 73–85, 06 2009.

[6] Y.-H. Wang, C.-T. Chung, and H. yi Lee, "Gate activation signal analysis for gated recurrent neural networks and its correlation with phoneme boundaries," in *INTERSPEECH*, 2017.