INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Contrastive Disentangled Learning for Memory-Augmented Transformer

Jen-Tzung Chien, Shang-En Li

Inst. of Electrical and Computer Engineering, National Yang Ming Chiao Tung University, Taiwan
jtchien@nycu.edu.tw, sean.ee08@nycu.edu.tw

Abstract

This paper developed a new memory-augmented sequential learning based on a contrastive disentangled transformer. Conventionally, transformer is insufficient to characterize long sequences since the sequence length is restricted to avoid the requirement of overlarge memory. A direct solution to handle this issue is to divide long sequence into short segments, but the context fragmentation will happen. In this paper, the contrastive disentangled memory is exploited to deal with the increasing computation cost as well as the overlarge memory requirement due to long sequences. In particular, an informative selection over the disentangled memory slots is proposed for iterative updating in a large-span sequence representation. This paper maximizes the semantic diversity of memory slots and captures the contextual semantics via contrastive learning. The experiments on language understanding show that the context fragmentation is mitigated by the proposed method with reduced computation.

Index Terms: Sequential learning, contrastive learning, disentangled memory, transformer, language understanding

1. Introduction

It is important to characterize long-term dependencies for sequential learning. Traditionally, recurrent neural network (RNN) or long short-term memory (LSTM) has been developed for sequence representation where the history information is recursively updated and condensed as a state vector. However, it is still insufficient to preserve large-span information in long sequences via recurrent networks. In [1], LSTM was analyzed to preserve the temporal information with a limited length in previous time steps. More recently, transformer [2, 3, 4, 5] has been explored for a variety of sequence-to-sequence learning tasks based on encoder-decoder framework. The key to a success using transformer is the self attention over sequence data via the transformer layer. This layer is learned to capture individual tokens as well as semantic contexts along a whole sequence. But, the shortcoming of transformer is caused by high computation due to full attention over a long string. The valid sequence length and contextual information in a real application is bounded. Such a shortcoming affects the deployment of a practical transformer in real world.

To handle the computation overhead and length limitation, there are two categories of solutions which are feasible to strengthen self attention in transformer. Firstly, the solutions based on sparse attention were proposed to reduce the computation and memory costs in full attention [6, 7]. In [8], a random window attention was exploited to alleviate the computation complexity of self attention along a long string sample. The second category of works was devoted to handle the length limitation. A direct solution was proposed by dividing an input string signal or sentence into segments to carry out transformer for sequence-to-sequence learning. Such a segment-wise attention is feasible to implement attention over long sequences. This block-wise attention served as a simple solution to run the attention on long sequences. An obvious weakness was that the sequence tokens could not access the context outside the segments which restricted the expressiveness of long-term dependencies beyond the predefined context length. To handle the *context fragmentation*, the transformer-XL [9] and compressive transformer [10] were proposed to design a first-in-first-out (FIFO) memory strategy to store the temporal information in history sentences in a fixed size of memory blocks. However, FIFO memory is likely redundant when saving large-span dependencies in a long sequence with the limited memory size.

This paper presents the contrastive disentanglement for sequence representation in a memory-augmented transformer where long-term features of input sequence is captured under the restriction of the predefined context length. To deal with the weakness of memory representation, there are twofold novelties proposed in this study. First, the information-theoretic disentanglement of memory slots is presented to enhance the semantic diversity of memory while reducing the memory redundancy. Second, a distinctive updating mechanism for memory slots instead of using FIFO method is developed to preserve the informative tokens and throw away the redundant tokens via a Gumbel-softmax function. Given the length of input sequence T, the proposed model architecture is feasible to reduce the time and memory complexities from $O(T^2)$ to $O(T)$ in inference stage while even increasing the classification performance in sequence-to-sequence learning tasks.

2. Learning for Transformer

2.1. Information-theoretic learning

This study presents a new disentangled memory representation using transformer based on the information-theoretic features which are surveyed as follows. Let X denote the input sequence and Y denote the output sequence which should be sufficiently preserved by informative mapping from X. Given their joint distribution $p(X, Y)$, a statistical dependency between X and Y can be measured by mutual information $I(X; Y)$. Y implicitly determines the relevant and irrelevant features in X. Information bottleneck method is feasible to find the informative representation of X which captures the relevant features and disregards the irrelevant features. In latent representation using transformer, we are finding the features Z which are most relevant to Y and simultaneously most compressed and irrelevant representation to X. The optimal representation Z is estimated by maximizing $I(Z; Y)$ under a constraint on $I(X; Z)$. The learning objective for Z is given by $I(Y; Z) - \beta I(X; Z)$

10.21437/Interspeech.2023-1652

where β is tuned to balance a tradeoff between the complexity of representation $I(X;Z)$ and the amount of the preserved relevant information $I(Y;Z)$. The learned features Z preserves the information in X which is useful to predict Y. In [11], the interpretable disentangled representation [12] was proposed with an annealing β in variational inference.

2.2. Sparse transformer

Vanilla transformer is built by self-attention layers with multiple heads where the individual pairs of tokens are mutually attended. Learning sparse attention has recently attracted considerable interests. Compared with full attention, the sparse transformer [8] presented the factorization of attention computation into local and stride operations which handled different heads to extract various sparse patterns through random attention and local attention. Sparse transformers were generally categorized into three groups. The first category of methods reduced the computation by making the attention maps sparse in a predetermined way. Each token in the sequence only attended a fixed and small set of the other tokens rather than the whole sequence. In [8, 13], the auxiliary tokens were added to improve the connectivity of the observed tokens while maintaining the sparsity. This method was weak because the sparse model was independent of input sequence and therefore was not adjustable to new data. The second category of methods imposed the sparsity in the calculation of attention map. The third category of methods attempted to boost the advantages of those in the first two categories. These works learned the sparse patterns from sequence data using the k-means clustering [14] and locality-sensitive hashing to predict the connection between tokens [15, 16]. The sparse patterns were determined before computing the attention matrix. The drawback was that the extra computation was required to train the additional module. This study pursues the disentangled memory for interpretable attention.

2.3. Memory-augmented transformer

In [9], transformer-XL was proposed to continuously store and update the past temporal information in a first-in-first-out (FIFO) memory. This memory-augmented transformer divided the input sequence into short segments and stored their features in a memory block [17, 18, 19, 20] which recorded a history of previous context. This method analyzed the sequence data in an identical manner similar to vanilla transformer except that, for each segment, the features calculated by the transformer were saved and connected with those of the next segment in a gradient-free manner. This method was able to preserve a large span of context in a memory which helped understanding from long sequence data. Let the t-th segment of length L [21] be denoted as $S_t = \{\mathbf{x}_m^t\}_{m=1}^L, \mathbf{x}_m^t \in \mathbb{R}^D$ and the hidden states or attended features H_t be produced by S_t based on the query Q_t, key K_t and value V_t via

$$\begin{aligned} Q_t = W_q S_t, \quad & K_t = W_k[f_{\text{sg}}(H_{t-1}) \oplus S_t] \\ V_t = W_v[f_{\text{sg}}(H_{t-1}) \oplus S_t], \quad & H_t = \text{Attn}(Q_t, K_t, V_t) \end{aligned} \tag{1}$$

where $\oplus$ denotes the concatenation, $f_{\text{sg}}(\cdot)$ denotes the stop-gradient function and $\{W_q, W_k, W_v\}$ denotes the model parameters. This calculation allows the model to access across previous segments from $t-1$ to t which provide a larger amount of input information and therefore assure a richer expressiveness. When the model scans to the next segment, the last past features in the memory are discarded, the remaining features were shifted, and the new features from a new segment are inserted in a FIFO manner. This paper focuses on the disentangled contextual representation from the tokens $X \leftarrow S_t$ and preserves the informative features of past segments in the memory $Y \leftarrow M_t$ as detailed in what follows.

3. Contrastive Disentangled Learning

In this study, the redundancy in memory-augmented representation is handled via the information-theoretic disentanglement [22]. Long sequences are represented by segment-based processing where large-span dependencies between segments are learned. The limitation of FIFO strategy in memory updating is tackled by an adaptive updating method using the Gumbel-softmax operation [23, 24] as a binary selector. The importance of tokens is emphasized via a self updating over memory slots.

3.1. Self adaptation of memory slots

Let M_t denote the memory at time t and X denote the input sequence which is split or segmented into different segments S_t with a fixed length L. The adaptive memory updating is illustrated in Figure 1. The first part of the representation is to learn how to compress the representation which contains contextual information of L vectors in $S_t = \{\mathbf{x}_m^t\}_{m=1}^L$. These vectors are compressed as the features $Z_t = \{\mathbf{z}_m^t\}_{m=1}^L$ via an encoder with parameter θ and the outputs of mean $\mu(\mathbf{x}_m^t)$ and variance $\Sigma(\mathbf{x}_m^t))$ of a Gaussian distribution $\mathbf{z}_m^t \sim \mathcal{N}(\mu(\mathbf{x}_m^t), \Sigma(\mathbf{x}_m^t))$. These compressed features are selected and stored in the memory M_t. Similar to the previous variational networks [25], the compressed feature $\mathbf{z}_m^t$ corresponding to the observed token $\mathbf{x}_m^t$ is sampled from the variational distribution $q_\theta(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mu(\mathbf{x}), \Sigma(\mathbf{x}))$. The second part is to update the memory M_t via an adaptive method based on the Gumbel-softmax function $g_\phi(\cdot) = \text{Gumbel-Softmax}(\cdot)$ with parameter ϕ [23]. A Bernoulli indicator or a binary mask K is carried out over the concatenation of the latest memory M_{t-1} and the segment of individual compressed vectors $Z_t = \{\mathbf{z}_m^t\}_{m=1}^L$. Those vectors which are not selected by the trainable mask are disregarded. FIFO updating is replaced accordingly. Such a memory updating function $M_t = f_{\text{upd}}(M_{t-1}, S_t)$ is yielded by

$$\begin{aligned} & Z_t = [\mathbf{z}_1^t, \ldots, \mathbf{z}_L^t], \;\text{ where }\; \mathbf{z}_m^t \sim \mathcal{N}(\mu(\mathbf{x}_m^t), \Sigma(\mathbf{x}_m^t)) \\ & \widetilde{M} = M_{t-1} \oplus Z_t, \;\; K = g_\phi(\text{Attn}(\widetilde{M})W_g), \;\; M_t = K\widetilde{M}. \end{aligned}$$

Here, $\text{Attn}(\cdot)$ is a self-attention function by using the concatenated memoy $\widetilde{M}$ and W_g is the parameter of a two-layer feed-forward network with the outputs calculating the unnormalized log probability distributions for different memory slots. The distribution measures the probability of each memory slot remaining in the updated memory M_t. Notably, the probability generated by the masking layer is implemented by a Gumbel-softmax function where hard sampling of a binary mask is approximated. Different from FIFO strategy, a distinctive and adaptive scheme to memory updating is implemented. This updating consolidates the most relevant composition of memory slots for prediction of future context S_{t+1}.

3.2. Contrastive disentangled learning

Importantly, the redundant information in memory-augmented transformer is reduced via an informative disentanglement [26]. The learning objective is constructed to pursue the optimally compressed or disentangled memory M_t corresponding to input segment S_t by maximally preserving the information to predict

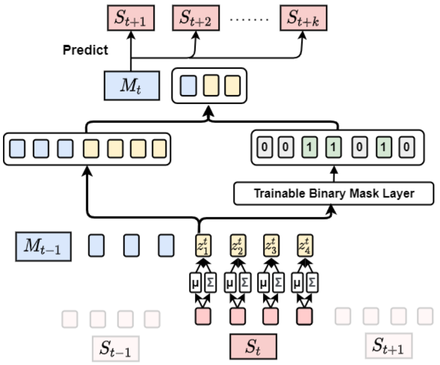

Figure 1: *Sequential memory updating via a binary mask layer.*

future segments $\{S_{t+1}, \ldots, S_{t+k}\}$ and increasing the semantic diversity $D(\cdot)$ of current memory by maximizing

$$I(S_{t+k}; M_t) - \beta(I(S_t; M_t) - D(M_t)) \tag{2}$$

where the second term is seen as a constraint. With $\beta > 1$, the memory M_t is pushed to learn a compact latent representation of input tokens, which is disentangled if the tokens in S_t contain the underlying variations that are independent of M_t. At the same time, the memory is diverse by maximizing $D(\cdot)$ and reflecting the semantic meaning of future information S_{t+k} as much as possible. The contrastive loss or InfoNCE loss [27] is used to estimate the mutual information between memory M_t and future segments S_{t+k}. Considering a set of N_s segments $S = \{S_t\}_{t=1}^{N_s}$ containing one positive sample from $p(S_{t+k}|M_t)$ and $N_s - 1$ negative samples from S which is randomly picked from the segments in the same batch, we maximize

$$I(S_{t+k}; M_t) \approx -\mathcal{L}_{\text{InfoNCE}} = \mathbb{E}_S \left[\log \frac{f(S_{t+k}, M_t)}{\sum_{S_i \in S} f(S_i, M_t)} \right] \tag{3}$$

where $f(S_{t+k}, M_t) \propto \frac{p(S_{t+k}|M_t)}{p(S_{t+k})}$ is an estimate of density ratio. This study practically uses the average pooling to calculate the segment-level representation of memory M_t and segment S_{t+k}, and then adopts the cosine similarity to calculate the density ratio between them. The objective $\mathcal{L}_{\text{InfoNCE}}$ is seen as the categorical cross-entropy of classifying the positive sample correctly through the form $\frac{f}{\sum_S f}$ for representing the prediction model. This prediction model corresponds to calculate the conditional probability of detecting a positive segment $S_i \in S$. The probability $p(S_i|S, M_t)$ of a segment S_i drawn from the conditional distribution $p(S_i|M_t)$ rather than the proposal distribution $p(S_i)$ is calculated by

$$\frac{p(S_i|M_t) \prod_{l \neq i} p(S_l)}{\sum_{j=1}^{N_s} p(S_j|M_t) \prod_{l \neq j} p(S_l)} = \frac{\frac{p(S_i|M_t)}{p(S_i)}}{\sum_{j=1}^{N_s} \frac{p(S_j|M_t)}{p(S_j)}}. \tag{4}$$

Owing to high dimensionality of input features in S_t, it is challenging to compute $I(S_t; M_t)$ when the memory size increases. The efficient calculations for the diversity of memory and the mutual information between memory M_t and input segment S_t are required. In the implementation, the localized information [28] between input token $\mathbf{x}$ and compressed feature $\mathbf{z}$ is calculated. To enhance semantic diversity in the memory, the binary mask is estimated to drop the tokens that are redundant with similar meaning. Accordingly, the constraint loss $\mathcal{L}_{\text{constr}}$ in the second term of Eq. (2) is calculated for individual samples $\{\mathbf{x}_m^t, \mathbf{z}_m\}$. Mutual information and diversity measure are calculated through Kullback-Leiblier (KL) divergence by

$$\sum_{\mathbf{x}_m^t \in S_t} I(\mathbf{x}_m^t; \mathbf{z}_m) - D(M_t) = \sum_{\mathbf{x}_m^t \in S_t} \text{KL}[p_\theta(\mathbf{z}|\mathbf{x}_m^t) \| p(\mathbf{z})]$$
$$- \sum_{\mathbf{x}_m \in B_t} \sum_{\mathbf{x}_j \in B_t} \text{KL}(p_\theta(\mathbf{z}|\mathbf{x}_m) \| p_\theta(\mathbf{z}|\mathbf{x}_j)) \triangleq \mathcal{L}_{\text{constr}} \tag{5}$$

where B_t is the buffer to store all original tokens at time t. This constraint loss is minimized to encourage the memory learning a disentangled representation [11] as well as enhance the semantic diversity of the updated memory. The overall loss for contrastive disentangled memory is formed by $\mathcal{L}_{\text{mem}} = \mathcal{L}_{\text{InfoNCE}} + \beta \mathcal{L}_{\text{constr}}$. Minimizing the loss $\mathcal{L}_{\text{mem}}$ is equivalent to maximizing the bound of mutual information in Eq. (3). Notably, the training procedure for the contrastive disentangled memory is performed in a way of *self-supervised* sequential learning where an efficient memory updating is learned without any labeled data.

3.3. Contrastive disentangled transformer

The contrastive disentangled memory is then combined in transformer to construct the contrastive disentangled transformer (CDT) for supervised sequence-to-sequence learning. CDT is seen as a new transformer powered by the contrastive disentangled memory which is utilized to map from input sequence $X = \{\mathbf{x}_m\}_{m=1}^{T_i}$ to output sequence $Y = \{\mathbf{y}_n\}_{n=1}^{T_o}$ with different lengths. A sliding window with a fixed size L is used to split X into individual segments $S = \{S_t\}_{t=1}^{N_s}$ with $N_s = \lceil \frac{T_i}{L} \rceil$. To enhance the dependencies between different segments, this new transformer carries out the memory-augmented multi-head attention in the encoder with parameter $\theta_e = \{W_q, W_k, W_v\}$

$$Q_t = W_q S_t, \quad K_t = W_k[M_{t-1} \oplus S_t]$$
$$V_t = W_v[M_{t-1} \oplus S_t], \quad H_t = \text{Attn}(Q_t, K_t, V_t).$$

This encoder is operated by finding the memory-driven input features $H = \{H_t\}_{t=1}^{N_s}$ where $H_t = \text{Encoder}(S_t, M_{t-1}; \theta_e)$. Importantly, the contrastive disentangled memory, updated by $M_t = f_{\text{upd}}(M_{t-1}, S_t)$ in Sec. 3.1 and trained by minimizing memory loss $\mathcal{L}_{\text{mem}}$ in Eqs. (3)(5), is configured to build CDT. Importantly, a decoder with parameter θ_d is incorporated to predict y_n of a target sequence $\mathbf{y}_{1:T_o}$ or $\{y_n\}_{n=1}^{T_o}$ sequentially by given the previous tokens $\mathbf{y}_{1:n-1}$. Segmentation is run for learning the self-attention layers of decoder or the cross-attention layers between encoder and decoder. The conditional likelihood for prediction of an output sample y_n is calculated via the decoder or classifier $p(\mathbf{y}_n|\mathbf{y}_{1:n-1}, X) = \text{Decoder}(\mathbf{y}_{1:n-1}, H; \theta_d)$. This CDT is trained by minimizing the memory loss for disentangled updates as well as the classification loss for sequence generation

$$\mathcal{L}_{\text{cdt}} = \mathcal{L}_{\text{men}} + \mathbb{E}_{X,Y \sim p(X,Y)}[-\log p(Y|X)]. \tag{6}$$

4. Experiments

4.1. Experimental setup

WMT dataset was used to evaluate the proposed method for machine translation where 4M, 6K, and 7K pairs of sentences were used for training, validation and testing, respectively. Byte pair encoding [29] with 32K merge operations were employed to segment the words into subword units. All attention models were built with 6 blocks. Using WMT, the dimensions of word embeddings, hidden states and heads were 512, 1024 and 8, respectively. The evaluations were done by using FAIRSEQ framework [30]. All of the models were composed by an additional convolution module for subsampling, consisting of two layers of 2D-convolution with ReLU activation, stride size 2, 256 channels and kernel size 3. The dimensions of blocks, hidden states and heads in all attention models were 6, 1024, and 8, respectively. In WMT, all models were trained from scratch with Adam optimizer with initial learning rate 1e-3, hyperparameter β and memory slot number were set to 1.1 and 10, respectively. The dropout rate for attention and feed-forward layers was set as 0.1. The proposed CDT (also denoted as the disentangled transformer) was compared with four strong baselines including transformer [2], sparse transformer [8], transformer-XL [9] and compressive transformer [10, 31].

4.2. Experimental results

Table 1 reports the results using WMT dataset. Inference time relative to transformer is shown. In WMT En-De and English-to-French (En-Fr), CDT obtains the highest BLEU in comparison with other efficient transformers, and also attains the BLEU close to transformer but with one-third inference time.

Model	En→De	En→Fr	Cost
Transformer	27.3	38.1	1x
Sparse Transformer	23.2	34.8	0.39x
Transformer-XL	24.1	35.2	**0.31x**
Compressive Transformer	24.6	37.1	0.36x
Disentangled Transformer	**27.4**	**38.4**	0.33x

Table 1: *Results of BLEUs and costs on WMT En-De and En-Fr.*

A number of analyses are performed. Diversity analysis is first investigated. The embedding vectors in a group of texts can be treated as a cluster in high-dimensional space. Specifically, the shape of an isocontour of a cluster is seen as an axis-aligned ellipsoid in $\mathbb{R}^D$ where D is the embedding dimension. A diversity metric [32] is measured to estimate the cluster's dispersion or spreadness via a generalized sense of the radius. The diversity metric is defined as the geometric mean of radius across all axes by $M_{\text{diversity}} = (r_1 r_2 \cdots r_D)^{\frac{1}{D}}$ where r_i is the radius or the standard deviation of the cluster along the i-th axis. Table 2 shows that the semantic diversity using CDT memory is much larger than those memories in compressive transformer and transformer-XL. With a memory containing richer diversity, CDT receives better disentangled representation than the other transformers using FIFO memory in machine translation.

Model	Diversity
Transformer-XL	2.314
Compressive Transformer	2.661
Disentangled Transformer	**5.253**

Table 2: *Results of the diversity of memory on WMT En-Fr.*

The updating algorithm in a memory system is crucial. Using the FIFO strategy, the memory blocks were deleted, shifted and added in a regular order without considering how often they were accessed before. However, the optimal memory updating algorithm needs to make sure that the system can swap out unnecessary slot when a slot needs to be swapped in, which is very hard to be achieved in practice. To address this issue, contrastive disentangled memory is designed to mimic the optimal algorithm by utilizing the contrastive learning. As shown in Table 3, CDT outperforms FIFO memory-augmented transformers even with a smaller number of memory slots.

Model	Memory Slots	BLEU
Transformer-XL	5	35.1
Transformer-XL	10	35.2
Transformer-XL	20	36.1
Disentangled Transformer	5	37.2
Disentangled Transformer	10	38.4
Disentangled Transformer	20	**38.8**

Table 3: *BLEUs under different size of memory slots on WMT.*

Model	#Params	Perplexity
Transformer	128M	31.2
Sparse Transformer	156M	27.3
Transformer-XL	151M	24.0
Disentangled Transformer	153M	**21.9**

Table 4: *Model sizes and perplexities on WikiText-103.*

Different methods are further investigated by language modeling in presence of long sequences where the document-level context using WikiText-103 is adopted to evaluate the long-term dependencies. This dataset contains 103M training tokens from 28K articles wtih an averaged length of 3.6K tokens per article. In the implementation, the attention length, memory size, layer size, head size and other hyperparameters are referred to those of transformer-XL [9]. Table 4 shows that the perplexity using word-level CDT is lower than those of character-level transformer [9] and word-level sparse transformer and transformer-XL. The increase of parameter size using CDT relative to transformer-XL is limited. The proposed CDT is seen as a general approach to different lengths of training and test sequences for language understanding.

5. Conclusions

This paper has presented a distinctive and adaptive memory updating to preserve informative elements and discard the redundant elements. Latent disentanglement and memory updating were performed by maximizing the information-theoretic objective with Gumbel-softmax operation. With the enriched memory, transformer-based model could handle long sequences with a linear computational complexity and a constant memory complexity. This improvement tackled the issue of memory limitation and redundant information in transformers. From the experiments on three sequential learning tasks, the disentangled memory was demonstrated to be effective to memorize the past information with a small size of memory space. The experimental results showed that this transformer outperformed the other transformers and even achieved comparable performance with vanilla transformer but under a relatively low computation time.

6. References

[1] J. Zhao, F. Huang, J. Lv, Y. Duan, Z. Qin, G. Li, and G. Tian, “Do RNN and LSTM have long memory?” in *Proc. of International Conference of Machine Learning*, 2020, pp. 11 365–11 375.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[3] J.-T. Chien and W.-H. Chang, “Dualformer: a unified bidirectional sequence-to-sequence learning,” in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2021, pp. 7718–7722.

[4] H. Lio, S.-E. Li, and J.-T. Chien, “Adversarial mask transformer for sequential learning,” in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022, pp. 4178–4182.

[5] J.-T. Chien and C.-W. Tsao, “Graph evolving and embedding in transformer,” in *Proc. of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, 2022, pp. 538–545.

[6] Q. Wu, Z. Lan, K. Qian, J. Gu, A. Geramifard, and Z. Yu, “Memformer: A memory-augmented transformer for sequence modeling,” in *Proc. of International Joint Conference on Natural Language Processing*, 2022, pp. 308–318.

[7] C.-Y. Wu, Y. Li, K. Mangalam, H. Fan, B. Xiong, J. Malik, and C. Feichtenhofer, “MeMViT: Memory-augmented multiscale vision transformer for efficient long-term video recognition,” in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, June 2022, pp. 13 587–13 597.

[8] M. Zaheer, G. Guruganesh, K. A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, and A. Ahmed, “Big bird: Transformers for longer sequences,” in *Advances in Neural Information Processing Systems*, 2020, pp. 17 283–17 297.

[9] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Quoc, and R. Salakhutdinov, “Transformer-xl: Attentive language models beyond a fixed-length context,” in *Proc. of Annual Meeting of Association for Computational Linguistics*, 2019, pp. 2978–2988.

[10] J. W. Rae, A. Potapenko, S. M. Jayakumar, C. Hillier, and T. P. Lillicrap, “Compressive transformers for long-range sequence modelling,” in *Proc. of International Conference on Learning Representations*, 2020.

[11] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, “Understanding disentangling in β-VAE,” in *arXiv preprint arXiv:1804.03599*, 2017.

[12] J.-T. Chien and Y.-H. Huang, “Bayesian transformer using disentangled mask attention,” in *Proc. of Annual Conference of International Speech Communication Association*, 2022, pp. 1761–1765.

[13] I. Beltagy, M. E. Peters, and A. Cohan, “Longformer: The long-document transformer,” *arXiv preprint arXiv:2004.05150*, 2020.

[14] J.-T. Chien and C.-H. Chueh, “Topic-based hierarchical segmentation,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 55–66, 2012.

[15] N. Kitaev, Lukasz Kaiser, and A. Levskaya, “Reformer: The efficient transformer,” in *Proc. of International Conference on Learning Representations*, 2020.

[16] A. Vyas, A. Katharopoulos, and F. Fleuret, “Fast transformers with clustered attention,” in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 21 665–21 674.

[17] J.-T. Chien and T.-A. Lin, “Supportive attention in end-to-end memory networks,” in *Proc. of IEEE International Workshop on Machine Learning for Signal Processing*, 2018, pp. 1–6.

[18] K.-W. Tsou and J.-T. Chien, “Memory augmented neural network for source separation,” in *Proc. of IEEE International Workshop on Machine Learning for Signal Processing*, 2017, pp. 1–6.

[19] R. Qiu, Z. Huang, and H. Yin, “Memory augmented multi-instance contrastive predictive coding for sequential recommendation,” in *Proc. of IEEE International Conference on Data Mining*, 2021, pp. 519–528.

[20] Z. Lai, E. Lu, and W. Xie, “MAST: A memory-augmented self-supervised tracker,” in *Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6479–6488.

[21] J.-T. Chien and C.-H. Huang, “Bayesian learning of speech duration models,” *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 558–567, 2003.

[22] W. Wei, C. Huang, L. Xia, Y. Xu, J. Zhao, and D. Yin, “Contrastive meta learning with behavior multiplicity for recommendation,” in *Proc. of ACM international conference on web search and data mining*, 2022, pp. 1120–1128.

[23] E. Jang, S. Gu, and B. Poole, “Categorical reparameterization with Gumbel-Softmax,” in *Proc. of International Conference on Learning Representations*, 2017.

[24] J.-T. Chien and C.-Y. Kuo, “Markov recurrent neural network language model,” in *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop*, 2019, pp. 807–813.

[25] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, “Deep variational information bottleneck,” in *Proc. of International Conference on Learning Representations*, 2017.

[26] J.-T. Chien and S.-J. Huang, “Learning flow-based disentanglement,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2022.

[27] A. van den Oord, Y. Li, and O. Vinyals, “Representation learning with contrastive predictive coding,” in *arXiv preprint arXiv:1807.03748*, 2018.

[28] B. Wang, S. Wang, Y. Cheng, Z. Gan, R. Jia, B. Li, and J. Liu, “InfoBERT: Improving robustness of language models from an information theoretic perspective,” in *Proc. of International Conference on Learning Representations*, 2021.

[29] R. Sennrich, B. Haddow, and A. Birch, “Neural machine translation of rare words with subword units,” in *Proc. of Annual Meeting of Association for Computational Linguistics*, 2016.

[30] 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 *Proc. of Conference of North American Chapter of Association for Computational Linguistics*, 2019, pp. 48–53.

[31] C.-H. Leong, Y.-H. Huang, and J.-T. Chien, “Online compressive transformer for end-to-end speech recognition,” *Proc. of Annual Conference of International Speech Communication Association*, pp. 2082–2086, 2021.

[32] Y.-A. Lai, X. Zhu, , Y. Zhang, and M. Diab, “Diversity, density, and homogeneity: Quantitative characteristic metrics for text collection,” in *Proc. of Language Resources and Evaluation Conference*, 2020, pp. 1739–1746.