Efficient Long Sequence Encoding via Synchronization

Anonymous ACL submission

Abstract

Pre-trained Transformer models have achieved successes in a wide range of NLP tasks, but are inefficient when dealing with long input sequences. Existing studies try to overcome this challenge via segmenting the long sequence followed by hierarchical encoding or post-hoc aggregation. We propose a synchronization mechanism for hierarchical encoding. Our approach first identifies anchor tokens across segments and groups them by their roles in the original input sequence. Then inside Transformer layer, anchor embeddings are synchronized within their group via a self-attention module. Our approach is a general framework with sufficient flexibility – when adapted to a new task, it is easy to be enhanced with the task-specific anchor definitions. Experiments on two representative tasks with different types of long input texts, NarrativeQA summary setting and wild multi-hop reasoning from HotpotQA, demonstrate that our approach is able to improve the global information exchange among segments while maintaining efficiency.

1 Introduction

Transformer-based encoders (Vaswani et al., 2017) have been widely used in natural language processing with successes. The pre-trained language models based on Transformer, such as BERT (Devlin et al., 2019), GPT-3 (Brown et al., 2020), T5 (Raffel et al., 2019) and BART (Lewis et al., 2019), further make it a dominating architecture in NLP.

Despite its successes, the Transformer models suffer from a major challenge in encoding long sequences. It is due to the fact that the self-attention mechanism used in each Transformer layer requires to compute attention for each pair of input words. Such computations lead to $O(l^2)$ complexity in time and space in each Transformer layer, where l is the sequence length. This limits Transformer's roles in the increasingly important long sequence encoding for two common scenarios: (1) *encoding of a single long document* with lengths exceeding the input limitation, and (2) *joint-encoding of multiple related documents* for tasks that require synthesizing scattered pieces of evidence, e.g., multi-hop reasoning and multi-document summarization. Figure 1 gives an example of the necessary information exchange in multi-hop QA. Each paragraph provides a partial clue to solve the task (shown as the connected entities). Intuitively, an effective global encoding should allow the entities (e.g., *Dutch-Belgian*) appearing in multiple paragraphs to share information across all their occurrences. In this way, the embedding of *Dutch-Belgian* in the second paragraph can be aware of partial evidence from the first one and resolve the required information of *House of Anubis*.

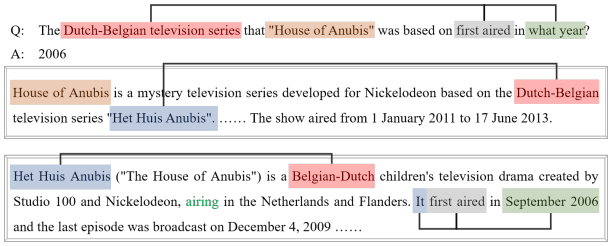

Figure 1: An example from HotpotQA. Different entities are color-coded. The solid lines indicate the correct partial evidence chain toward the true answer. *Dutch-Belgian* is an example of anchors that can pass the partial evidence to the others for collecting full evidence.

To overcome this difficulty, many techniques have been proposed. The existing solutions can be categorized into two classes. The first class is **hierarchical encoding**. The idea is to either explicitly split the input into multiple short segments for fast encoding of each segment and then exchange information on top of their embeddings following sample-agnostic strategies (Ainslie et al., 2020; Wang et al., 2020); or implicitly constrain the information exchange among tokens with a sparse attention map (Beltagy et al., 2020; Zaheer et al., 2020). The essential of these methods is to find efficient ways to pass information among

segments and compensate for the loss of important global context across segments. One solution is introducing a pseudo token for each segment and encouraging the pseudo tokens to attend one another during encoding (Ainslie et al., 2020) for inter-segment interactions. The second class is **post-hoc aggregation** in the generative framework, such as Fusion-in-Decoder (FiD, Izacard and Grave 2020) with BART model. The input is split into segments that are encoded independently by the encoder. Then the decoder casts global attention over all the segments and generates the prediction. This approach allows for only shallow information exchange because the encoding is purely localized; yet is proved empirically very powerful in many NLP tasks.

We propose an orthogonal direction towards an efficient encoding of long sequences. Our method starts with local segments in the post-hoc aggregation approaches and relies on our proposed **synchronization mechanisms** to swap useful information from the other relevant segments during encoding, so as to maintain global information. Formally, our approach first identifies a set of anchors in the segments and puts them into different groups based on the similarity in their semantic units or the roles they play in the original input sequence. The identified anchors and groups connect different segments logically and naturally. Our synchronization is applied only to the encoding stage where inside each Transformer layer of the encoder, we perform an additional embedding update for each anchor using other anchor embeddings in the same group after a normal local encoding. The local encoding and anchor synchronization happen iteratively so that the global information is propagated deeply among segments with anchors as bridges.

Compared to previous hierarchical encoding approaches with fixed communication designs, our approach is more powerful and flexible. First, our approach provides a finer-grained information exchange mechanism. Second, our approach is a general framework that reduces the problem of global encoding to synchronization schema design. For any new applications or tasks, it is easy to infuse human prior knowledge to the model by identifying task-specific anchors and anchor grouping.

We evaluate our approach on two different tasks that require encoding of long sequences: NarrativeQA (Kočiskỳ et al., 2018), where each input is a single long story summary; and a wild multi-hop QA task adapted from HotpotQA (Yang et al., 2018), where the evidence annotation is assumed unavailable and the input documents are treated more independently from each other. The two settings correspond to the representative examples of the aforementioned long sequence encoding scenarios (1) & (2). Results show that our approach significantly improves the performance while remaining efficient. Moreover, building on top of FiD, the state-of-the-art hierarchical method ETC (Ainslie et al., 2020) does not bring further improvement as we observe but our approach improves consistently.

2 The Transformer with Synchronization (TRANSYNC) Framework

In this section, we propose our TRANSYNC framework which extends Transformer layer with an embedding synchronization module attached to the end. Given a long context sequence C, we divide it into segments, i.e. $C = [s_1; s_2; ...; s_n]$ where s_i is the i-th segment of C and n is the number of segments. A segment can represent a natural sentence or a sequence in a certain length. Together with the question q, we re-organize the input to the Transformer and form a set of question-prefixed segments $\{s_i^q\}_{i=1}^n$, s.t.

$$s_i^q = [q; \text{<SEP>}; s_i] \tag{1}$$

where <SEP> is a special token. An embedding layer converts the text segments $\{s_i^q\}_{i=1}^n$ into their corresponding question-aware embeddings $\{\mathbf{e}_i^q\}_{i=1}^n$, s.t.

$$\mathbf{e}_i^q = [\mathbf{t}_i^1; \mathbf{t}_i^2; ...; \mathbf{t}_i^{l_i}] \in \mathbb{R}^{l_i \times d} \tag{2}$$

where l_i is the length of s_i^q's token sequence, d is the dimension of the feature vector and $\mathbf{t}_i^j \in \mathbb{R}^d$ is the embedding for the j-th token in the i-th segment.

For genericity, our synchronization is performed between the target anchor and the incoming anchors, following the idea of message passing. The values of the target anchor embedding $\mathbf{a}_t$ are updated with the weighted sum of the incoming anchor embeddings and itself, i.e.,

$$\mathbf{a}_t' = \sum_k \alpha_k \mathbf{a}^k, \quad s.t. \sum_k \alpha_k = 1 \tag{3}$$

where $\{\mathbf{a}^k\}$ are the embedding spans[1] of the same length within the same anchor group; α_k is the

[1] Some words may correspond to multiple tokens due to the byte pair encoding (BPE) algorithm.

normalized weight. In this work, for each anchor group, we form a new sequence from the selected anchor embeddings, i.e., $[\mathbf{a}^1; \mathbf{a}^2; ...; \mathbf{a}^k]$, and use a self-attention module to compute the weights and update the embedding values.

Our TRANSYNC framework is embedded in Transformer layer. The synchronization is performed between the local self-attention and normalization steps to achieve deep information exchanging. At the end of the last Transformer layer, the synchronized segment embeddings are fused into one by concatenating one another as follows:

$$[\mathbf{e}_1^q; \mathbf{e}_2^q; ...; \mathbf{e}_n^q] \in \mathbb{R}^{\sum_1^n l_i \times d} \quad (4)$$

The flexibility of our TRANSYNC framework is granted by the manifold strategies of identifying anchors in the segments and the heterogeneous message passing directions. The schema will be detailed in Section 3.

3 Evaluating Tasks

In this section, we introduce two experiments performed to verify the feasibility and flexibility of our TRANSYNC framework.

3.1 NarrativeQA

Task Description NarrativeQA dataset has a collection of 783 books and 789 movie scripts. Each book and script is annotated with a long summary and 30 question-answer pairs on average. NarrativeQA provides two different settings, the summary setting and the full-story setting. In this work, we follow the summary setting by answering questions from the summaries, and formulate it as a generative QA task due to the free-form annotated answers. NarrativeQA is a representative example of the first type of long sequence encoding scenarios: a single document with length exceeding the input limitation.

Synchronization Schema We split each summary into natural sentences as the segments $\{s_i\}$ and prefix them with the question q following Section 2. This breakup of the continuous summary sentences drops the global context across segments during encoding. As a compensation, we apply a segment-level synchronization, which takes the preceding question sequence as the anchor. Practically, we simply use the special token <SEP> that connects to the question in each segment as the representatives, which significantly reduce the synchronization cost. The segment-level synchronization happens only among the closest neighbouring segments, inspired by their natural order in the summary text. Intuitively, it provides each segment compressed contextual information from its neighbors; and makes the question embedding be aware of its matched contents across multiple segments. Therefore, we expect it can better deal with questions that require multiple sentences to answer.

3.2 Wild Multi-hop Reasoning

Task Description We construct a wild multi-hop reasoning task from the HotpotQA dataset which provided two evidence documents and eight distractor documents for each question. We adopt the realistic assumption with no evidence annotation provided, to investigate the models' ability to sort out the reasoning chains from multiple documents. We designed two settings on the HotpotQA dataset intending to verify the effect of various context lengths on different models. The **MultiHop-10** uses all the 8 distractors in the dataset, the concatenation of the documents is thus beyond the length limit of BART. The **MultiHop-6** uses only 4, which is on average within BART's limit.

With the wild multi-hop reasoning, we hope to justify if our TRANSYNC can effectively pass important messages across segments. For consistency, we also formulate it as a generative QA task with the goal of predicting a free-form or YES/NO answer given the question and the context.

Synchronization Schema We split each concatenated document into segments in similar lengths containing various numbers of natural sentences. We have two ways of synchronization according to the task's unique properties. Firstly, a similar segment-level synchronization schema is applied. However, due to the different segment splitting strategies, there is no continuation guaranteed between the neighboring segments. Therefore, we synchronize across all segments rather than only among the neighbors. Secondly, we take the titles of the original documents as word-level anchors. For simplicity, the titles are added to the input[2], immediately following the question sequence. Similarly, we perform synchronization among all the title-associated special tokens to cut down compu-

[2]The title words already appear in the document, hence adding them to the input does not introduce new information and is regarded as a fair comparison.

tational costs. Due to the multi-hop nature of the samples, we expect the token-level synchronization to help build latent connections among the evidence.

System	NarQA	MultiHop-10		MultiHop-6	
	Rouge-L	EM	F1	EM	F1
BART	64.78	41.63	54.85	55.62	69.96
FiD	66.57	55.65	69.35	57.42	71.30
FiD+ETC	65.89	55.46	69.31	57.52	71.66
TRANSYNC	**67.58**	**56.49**	**70.32**	**58.30**	**72.61**

Table 1: Overall results on the NarrativeQA and the two multi-hop setting tasks (%).

4 Experiments

Baseline Our backbone model is the pre-trained BART-large model[3]. We compare with three baselines: (1) the original **BART**, which directly takes the concatenation of the question and the raw sequence without splitting. The sequence is truncated with a maximum of 1,024 tokens. (2) **FiD** (Izacard and Grave, 2020), the state-of-the-art hierarchical encoding algorithms for generative Transformer models. (3) **FiD+ETC**, a FiD variant enhanced by our implementation of ETC (Ainslie et al., 2020) in the encoder.

Metrics Because of the generative nature of the NarrativeQA task, following previous works (Kočiskỳ et al., 2018; Tay et al., 2019; Mou et al., 2020), we evaluate the QA performance with Rouge-L (Lin, 2004).[4] On HotpotQA dataset, the Exact Match (EM) and F1 scores[5] are reported that are commonly used in open-domain QA evaluation. Both hypothesis and reference are lowercased with the punctuation removed before evaluation.

Overall Results Table 1 shows the overall results on all three tasks. Our proposed TRANSYNC achieves the best results on both NarrativeQA and our new wild multi-hop QA tasks.

To our surprise, splitting the long sentences into question-aware segments alone (FiD) gives strong results against the BART baseline. This indicates the post-hoc aggregation of local embeddings can handle a significant portion of testing cases, reflecting the absence of global reasoning in many existing datasets. Our synchronization mechanism compensates for the loss of global context resulting from the sequence splitting and brings a consistent 1% improvement over FiD across all three tasks. ETC does not provide a further improvement over FiD as our approach does. This empirically shows that ETC's synchronization mechanism does not provide complementary global information to the post-hoc aggregation approach.

Finally, aside from Table 1, we also experiment with different segment lengths and find that the split context length should be at least 2 times longer than the prefixing question for effective encoding; otherwise, the question would dominate in the segment and it would lead to a significant drop in performance. Together with the observations that FiD with short segments outperforms BART with long sequences in both settings, we conclude that the splitting length is a hyper-parameter worth tuning.

Efficiency Table 2 provides an analysis of the efficiency of our TRANSYNC framework. The complexity comparison shows that the TRANSYNC is more memory efficient than the BART baseline in theory and becomes more superior when $l_q \ll \frac{l_c}{n}$. We also compare the runtime speed empirically by measuring the average time used for encoding per token. Though the synchronization introduces extra complexities to the encoding procedure, our experiments on the NarraitveQA dataset verify that the overall speed of our methods remains doubled to the BART baseline.

System	$\mathcal{O}(f)$	Time/Token
BART Baseline	$(l_q + l_c)^2$	292 μs
TRANSYNC	$(l_q + \frac{l_c}{n})^2 \cdot n$	134 μs

Table 2: Efficiency comparison. l_q and l_c are the length of the question sequence and the context sequence; n is the number of split segments. The encoding time per token is averaged over 100 QA samples.

5 Conclusion

In this work, we propose TRANSYNC framework with flexible synchronization mechanisms for encoding long sequences. We demonstrate the feasibility of our method in reasoning tasks with long context, and also show its high adaptability to different scenarios. We consider our work to be valuable as an easy solution to address the long context issue in QA, and to be potentially applicable to other long sequence modeling tasks.

[3]Implementation from https://huggingface.co/

[4]We use an open-source evaluation library (Sharma et al., 2017): https://github.com/Maluuba/nlg-eval.

[5]The squad/evaluate-v1.1.py script is used.

References

Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. 2020. Etc: Encoding long and structured inputs in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 268–284.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT 2019*, pages 4171–4186.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering. *arXiv preprint arXiv:2007.01282*.

Tomáš Kočiskỳ, Jonathan Schwarz, Phil Blunsom, Chris Dyer, Karl Moritz Hermann, Gábor Melis, and Edward Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. page 10.

Xiangyang Mou, Mo Yu, Bingsheng Yao, Chenghao Yang, Xiaoxiao Guo, Saloni Potdar, and Hui Su. 2020. Frustratingly hard evidence retrieval for qa over books. *ACL 2020*, page 108.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Shikhar Sharma, Layla El Asri, Hannes Schulz, and Jeremie Zumer. 2017. Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *arXiv preprint arXiv:1706.09799*.

Yi Tay, Shuohang Wang, Anh Tuan Luu, Jie Fu, Minh C Phan, Xingdi Yuan, Jinfeng Rao, Siu Cheung Hui, and Aston Zhang. 2019. Simple and effective curriculum pointer-generator networks for reading comprehension over long narratives. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4922–4931.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Shuohang Wang, Luowei Zhou, Zhe Gan, Yen-Chun Chen, Yuwei Fang, Siqi Sun, Yu Cheng, and Jingjing Liu. 2020. Cluster-former: Clustering-based sparse transformer for long-range dependency encoding. *arXiv preprint arXiv:2009.06097*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of EMNLP 2018*, pages 2369–2380.

Manzil Zaheer, Guru Guruganesh, Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *arXiv preprint arXiv:2007.14062*.