BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension

논문 리뷰 세미나
백형렬

BART
차례

1. 구조
    - seq2seq (transformer enc-dec)

2. Pretrain
    - Arbirary nosing function으로 Corrupting text
        - Token Masking
        - Token Deletion
        - Text Infilling
        - Sentence Permutation
        - Document Rotation
    - Corrupted text를 다시 Original text로 복원하면서 Pretrain

3. Finetuning
    - Sequence Classification Task
    - Token Classification Task
    - Sequence Generation Task
    - Machine Translation

4. 실험 및 결과
    - 효과적 Pretrain은 Task에 따라 다름
    - Token masking 부류의 Pretrain은 두루 효과적
    - 특히 Text-infilling이 text generation & comprehension 모두에 효과적

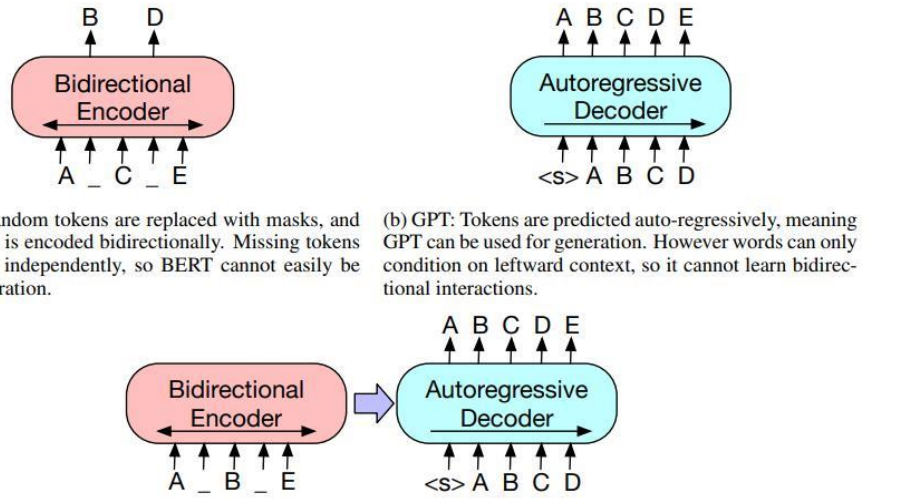
BART 구조

1. 구조

- Base Model: 6 Transformer Enc./Dec. Blocks
- Large Model: 12 Transformer Enc./Dec. Blocks

(- *BERT base/GPT2: 12 Transformer Enc./Dec. Blocks)

- Input: Enc./Dec. 각각 corrupted text, Original text
- Cross attn.: Enc.의 아웃풋을 key, value vector로 활용
- Dec. Masked attn.: Left To Right으로 참조



(a) BERT: Random tokens are replaced with masks, and the document is encoded bidirectionally. Missing tokens are predicted independently, so BERT cannot easily be used for generation.

(b) GPT: Tokens are predicted auto-regressively, meaning GPT can be used for generation. However words can only condition on leftward context, so it cannot learn bidirectional interactions.

(c) BART: Inputs to the encoder need not be aligned with decoder outputs, allowing arbitary noise transformations. Here, a document has been corrupted by replacing spans of text with mask symbols. The corrupted document (left) is encoded with a bidirectional model, and then the likelihood of the original document (right) is calculated with an autoregressive decoder. For fine-tuning, an uncorrupted document is input to both the encoder and decoder, and we use representations from the final hidden state of the decoder.

Figure 1: A schematic comparison of BART with BERT (Devlin et al., 2019) and GPT (Radford et al., 2018).

<그림1> BERT, GPT, BART 비교

BART 구조

1. 구조
   - Base Model: 6 Transformer Enc./Dec. Blocks
   - Large Model: 12 Transformer Enc./Dec. Blocks
   (- *BERT base/GPT2: 12 Transformer Enc./Dec. Blocks)
   - Input: Enc./Dec. 각각 corrupted text, Original text
   - Cross attn.: Enc.의 아웃풋을 key, value vector로 활용
   - Dec. Masked attn.: Left To Right으로 참조

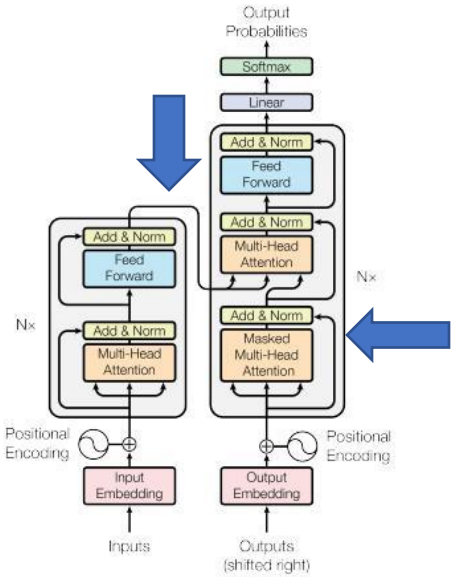
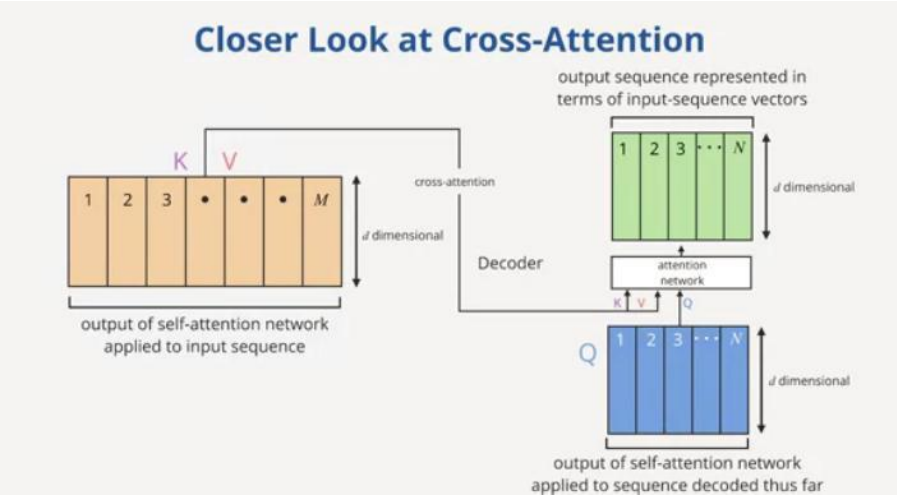


Figure 1: The Transformer - model architecture.

<그림2> Transformer



<그림3> Transformer Cross Attention
(출처:https://www.coursera.org/lecture/machine-learning-duke/cross-attention-in-the-sequence-to-sequence-model-oajUR)

Row: Query; Column: Key

	<s>	I	am	a	boy	<pad>
<s>	1.0					
I	0.1	0.9	Decoder Mask			
am	0.0	0.3	0.7			
a	0.1	0.2	0.2	0.6		
boy	0.0	0.1	0.1	0.1	0.7	
<pad>			Pad Mask			

<그림4> Transformer Decoder Masked Attention

BART
Pretrain

2. Pretrain
   - 거의 모든 방법으로 Noise생성 가능. Sentence 길이 달라져도 무관.
     - Token Masking: BERT착안. [Mask] token으로 대체.
     - Token Deletion: Random하게 token 삭제. 삭제 위치와 단어 맞춰야 함.
     - Text Infilling: SpanBERT착안. 다만 single [Mask] token으로 대체.
     - Sentence Permutation: Document input의 경우, sentence 순서 교체.
     - Document Rotation: Random하게 token선택 및 시작. 앞선 단어들은 맨 뒤로 보냄.

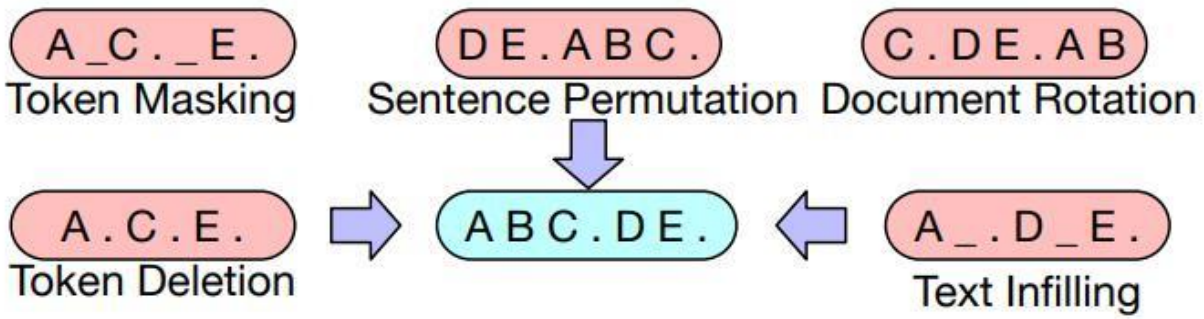
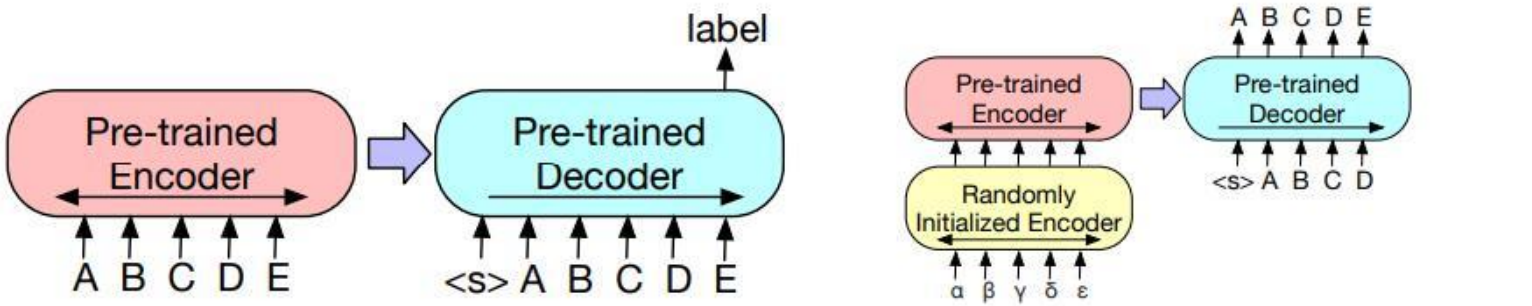


Figure 2: Transformations for noising the input that we experiment with. These transformations can be composed.

BART
Finetuning

3. Finetuning
   - Sequence Classification Task: BERT와 달리 Dec.의 last order output 사용. (eg. 감정분류)
   - Token Classification Task: Dec. last layer의 hidden state 사용 (eg. POS tagging)
   - Sequence Generation Task: QA의 경우, Enc.에 Question을 넣고, Dec.는 <s> 넣고 시작.
   - Machine Translation
     - Embedding만 다른 Encoder 추가(Embedding은 Random 초기화)
     - Unfrozen 1st layer: initialized embedding, positional embedding, self-attn input project 학습
     - Unfrozen all layer: 모든 parameter 학습



(a) To use BART for classification problems, the same input is fed into the encoder and decoder, and the representation from the final output is used.

(b) For machine translation, we learn a small additional encoder that replaces the word embeddings in BART. The new encoder can use a disjoint vocabulary.

Figure 3: Fine tuning BART for classification and translation.

BART

실험 및 결과

4.1. Pretrain Objectives 비교

1) 기존 Pretrain Objectives 적용한 BERT vs 새로 제안한 objectives 적용한 BART
   - LM, Permuted LM(XLNet), Masked LM(BERT), Multitask Masked LM(UniLM), Masked Seq2seq(MASS)

2) a. task에 맞는 pretrain방법이 따로 존재; b. token masking 중요; c. SQuAD는 Bidirection 중요; d.B ART Text Infilling 성능우수

Model	SQuAD 1.1 F1	MNLI Acc	ELI5 PPL	XSum PPL	ConvAI2 PPL	CNN/DM PPL
BERT Base (Devlin et al., 2019)	88.5	**84.3**	-	-	-	-
Masked Language Model	90.0	83.5	24.77	7.87	12.59	7.06
Masked Seq2seq	87.0	82.1	23.40	6.80	11.43	6.19
Language Model	76.7	80.1	**21.40**	7.00	11.51	6.56
Permuted Language Model	89.1	83.7	24.03	7.69	12.23	6.96
Multitask Masked Language Model	89.2	82.4	23.73	7.50	12.39	6.74
BART Base						
w/ Token Masking	90.4	84.1	25.05	7.08	11.73	6.10
w/ Token Deletion	90.4	84.1	24.61	6.90	11.46	5.87
w/ Text Infilling	**90.8**	84.0	24.26	**6.61**	**11.05**	5.83
w/ Document Rotation	77.2	75.3	53.69	17.14	19.87	10.59
w/ Sentence Shuffling	85.4	81.5	41.87	10.93	16.67	7.89
w/ Text Infilling + Sentence Shuffling	**90.8**	83.8	24.17	6.62	11.12	**5.41**

BART
실험 및 결과

4.2. 모델 간 성능 비교

- BART train setup: bs=8,000; iter=500,000; BPE; pretrain=text infilling, sentence shuffling; pre train data=corpus(160GB)

1) Discriminative Task
   - 비교모델: RoBERTa(bigger batch, longer training time, more data)
   - 결과: decoder있지만 classification task도 잘 해결

2) Genration task
   - 요약(abstractive 요약 포함), QA, 대화 등 탁월

3) Translation
   - back-translation aug.하면 더 좋아짐

	SQuAD 1.1 EM/F1	SQuAD 2.0 EM/F1	MNLI m/mm	SST Acc	QQP Acc	QNLI Acc	STS-B Acc	RTE Acc	MRPC Acc	CoLA Mcc
BERT	84.1/90.9	79.0/81.8	86.6/-	93.2	91.3	92.3	90.0	70.4	88.0	60.6
UniLM	-/-	80.5/83.4	87.0/85.9	94.5	-	92.7	-	70.9	-	61.1
XLNet	**89.0**/94.5	86.1/88.8	89.8/-	95.6	91.8	93.9	91.8	83.8	89.2	63.6
RoBERTa	88.9/**94.6**	**86.5/89.4**	**90.2/90.2**	96.4	92.2	94.7	**92.4**	86.6	**90.9**	**68.0**
BART	88.8/**94.6**	86.1/89.2	89.9/90.1	**96.6**	**92.5**	**94.9**	91.2	**87.0**	90.4	62.8

<Discriminative Task Score>

BART

실험 및 결과

4.2. 모델 간 성능 비교

- BART train setup: bs=8,000; iter=500,000; BPE; pretrain=text infilling, sentence shuffling; pre train data=corpus(160GB)

1) Discriminative Task
   - 비교모델: RoBERTa(bigger batch, longer training time, more data)
   - 결과: decoder있지만 classification task도 잘 해결
2) Genration task
   - 요약(abstractive 요약 포함), QA, 대화 등 탁월
3) Translation
   - back-translation aug하면 더 좋아짐

	CNN/DailyMail			XSum		
	R1	R2	RL	R1	R2	RL
Lead-3	40.42	17.62	36.67	16.30	1.60	11.95
PTGEN (See et al., 2017)	36.44	15.66	33.42	29.70	9.21	23.24
PTGEN+COV (See et al., 2017)	39.53	17.28	36.38	28.10	8.02	21.72
UniLM	43.33	20.21	40.51	-	-	-
BERTSUMABS (Liu & Lapata, 2019)	41.72	19.39	38.76	38.76	16.33	31.15
BERTSUMEXTABS (Liu & Lapata, 2019)	42.13	19.60	39.18	38.81	16.50	31.27
BART	**44.16**	**21.28**	**40.90**	**45.14**	**22.27**	**37.25**

Table 3: Results on two standard summarization datasets. BART outperforms previous work on summarization on two tasks and all metrics, with gains of roughly 6 points on the more abstractive dataset.

<Generative task score>

	RO-EN
Baseline	36.80
Fixed BART	36.29
Tuned BART	**37.96**

Table 6: The performance (BLEU) of baseline and BART on WMT'16 RO-EN augmented with back-translation data. BART improves over a strong back-translation (BT) baseline by using monolingual English pre-training.

<Translation>

END

논문 리뷰 세미나
백형렬

BART
참고

1. EM/F1 scoer
"F1 score, which captures the precision and recall that words chosen as being part of the answer are actually part of the answer, and an exact match EM score, which is the number of answers that are exactly correct (with the same start and end index)."

2. m/mm score
- matched acc./mismatched acc.

3. Dataset
- SQuAD: Extractive Question Answering task
- MNLI: Textual Entailment
- ELI5: Abstractive Question Answering
- XSum: Abstractive summaries
- ConvAI2: a dialogue response generation task
- CNN/DM: Extractive new summries

4. Pretrain Obj.
- Permuted LM: eg. [t1, t2, t3, t4, t5] -> t

BART
참고

4. Pretrain Obj.
- XLnet, Permuted LM

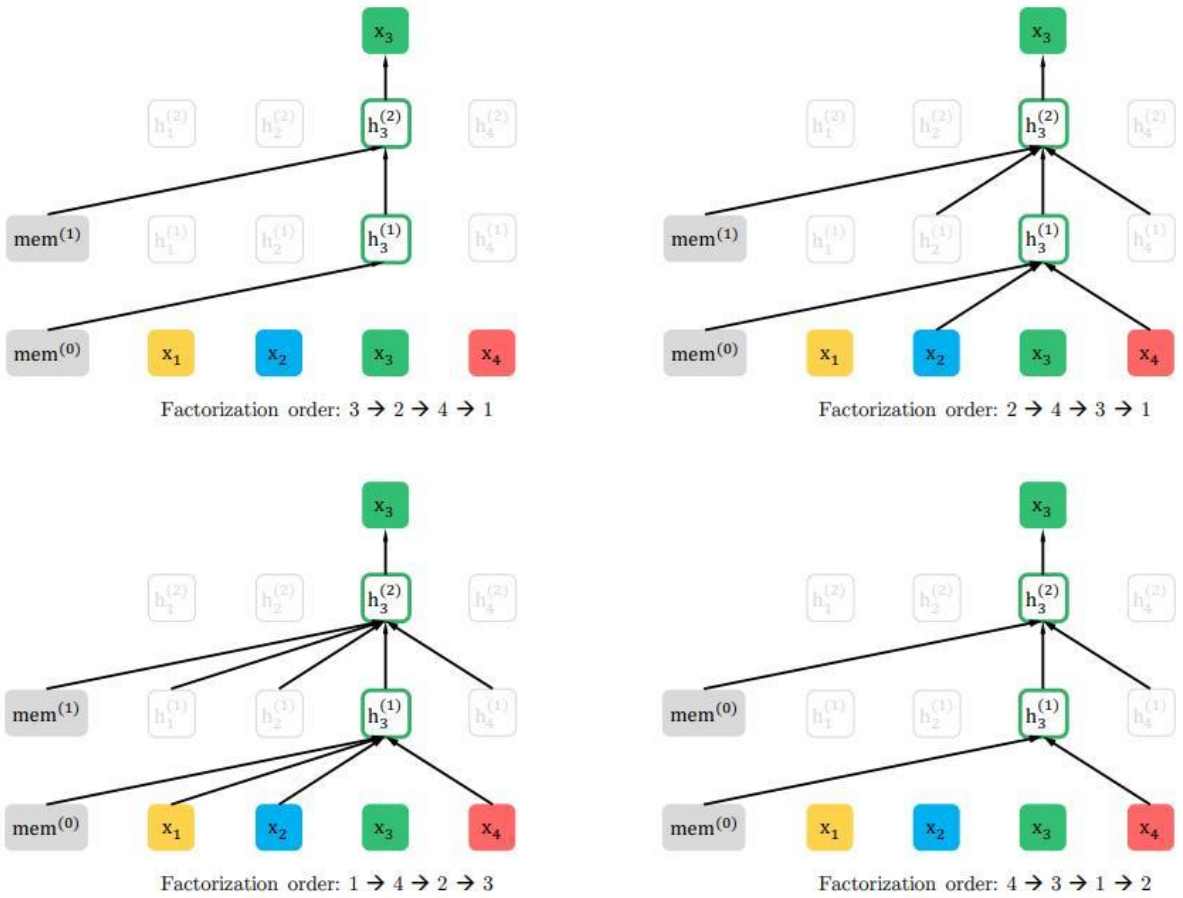


Figure 4: Illustration of the permutation language modeling objective for predicting x_3 given the same input sequence $\mathbf{x}$ but with different factorization orders.

BART
참고

4. Pretrain Obj.
- SpanBERT

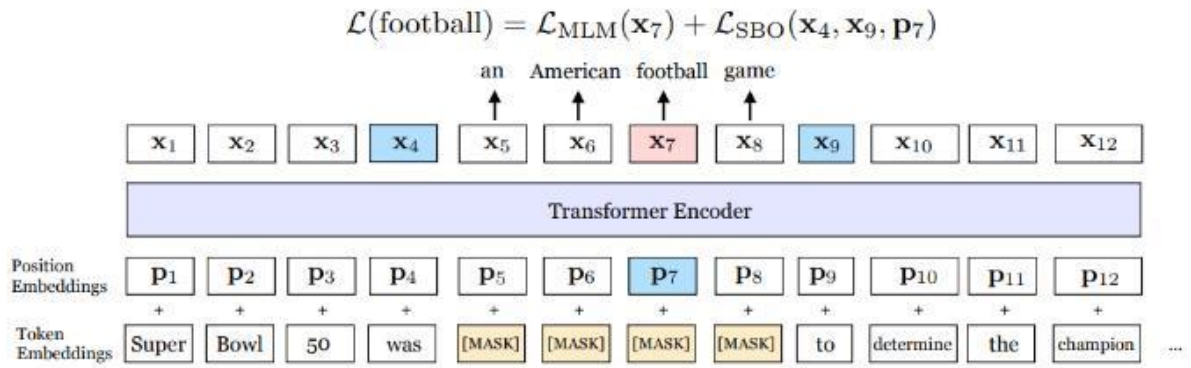


Figure 1: An illustration of SpanBERT training. In this example, the span *an American football game* is masked. The span boundary objective then uses the boundary tokens *was* and *to* to predict each token in the masked span.

- MASS, Masked seq2seq: Dec.에 AutoRegressive하게 input

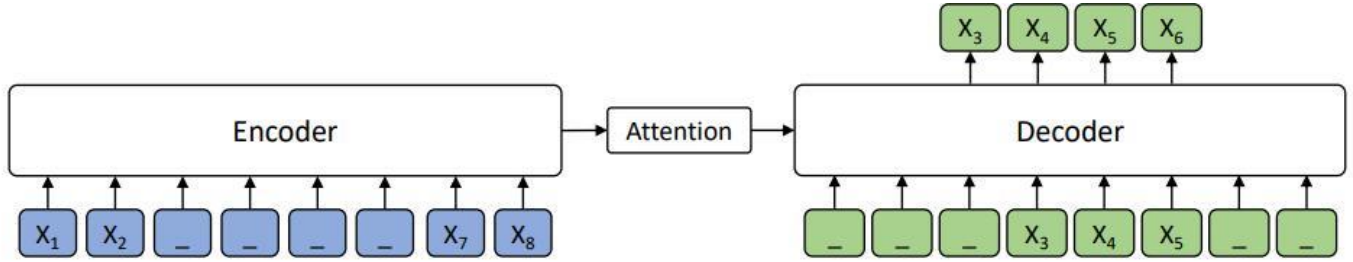


Figure 1. The encoder-decoder framework for our proposed MASS. The token "_" represents the mask symbol [M].

BART
참고

4. Pretrain Obj.

- UniLM, Multitask Masked LM: MLM에 self-attn.을 적용

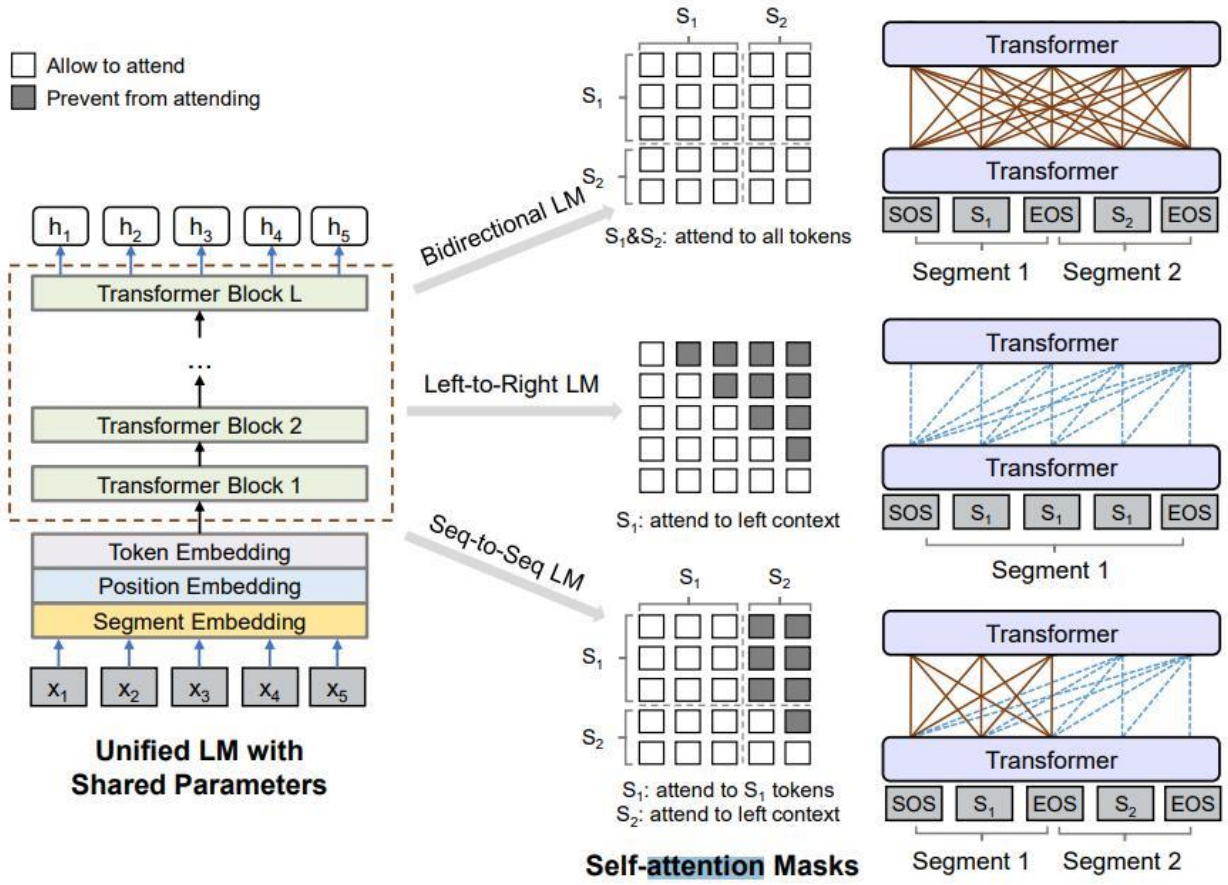


Figure 1: Overview of unified LM pre-training. The model parameters are shared across the LM objectives (i.e., bidirectional LM, unidirectional LM, and sequence-to-sequence LM). We use different self-attention masks to control the access to context for each word token. The right-to-left LM is similar to the left-to-right one, which is omitted in the figure for brevity.