2024-1 자연어 세미나

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

Facebook AI, ACL, 2019

인공지능 연구실 석사 과정 1기 김정현

CONTENTS

Introduction

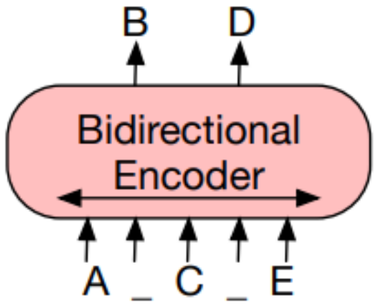
BART = Bidirectional and Auto-Regressive Transformers

- BART uses a standard Transformer-based neural machine translation architecture
- Generalizing BERT(bidirectional encoder), GPT(left-to-right decoder), and ...

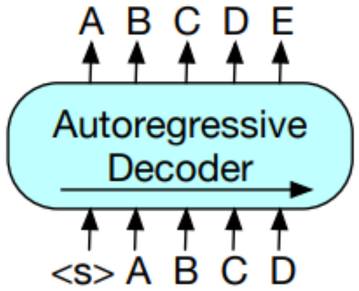
BART Contribution

- Finds a best noising approach (Text Infilling + Sentence Permutation)
- Effective when fine-tuned for text generation + comprehension tasks
- Present a new scheme for machine translation

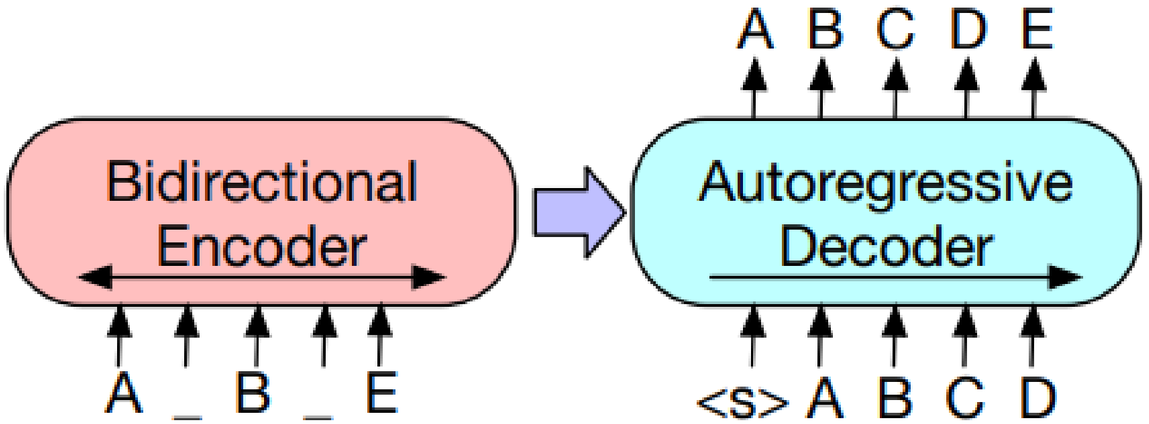
Introduction



BERT



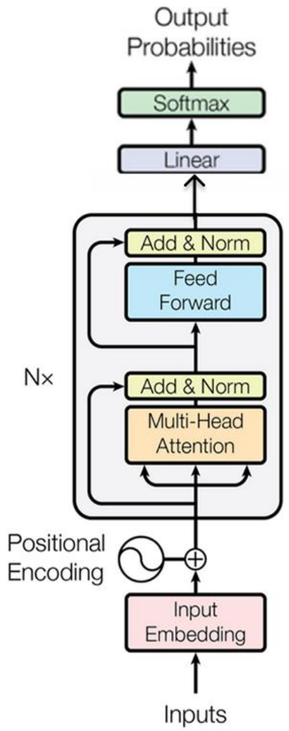
GPT



BART
(BERT + GPT)

Model

Base 기준



Hyperparameter

Layer : 12

Head : 12

d_model : 768

d_ff : 3072

V : 30522

BERT

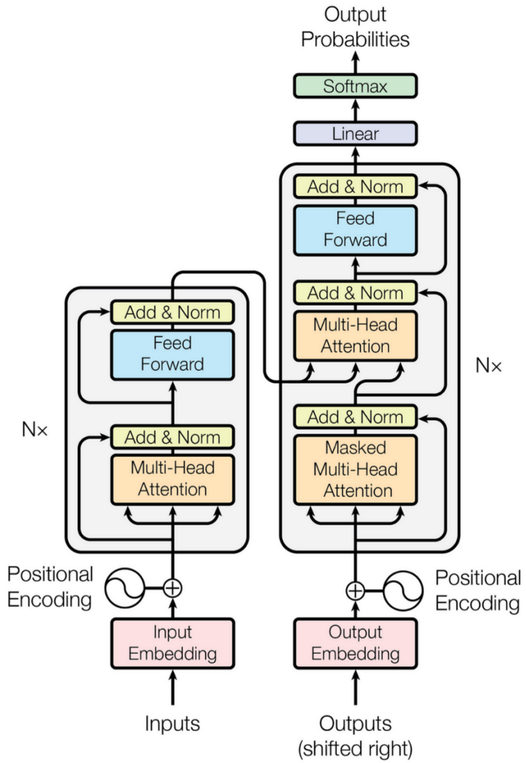


Figure 1: The Transformer - model architecture.

Hyperparameter

Layer :6 + 6

Head : 12

d_model : 768

d_ff : 3072

V : 50257 (GPT2)

50265 in Huggingface

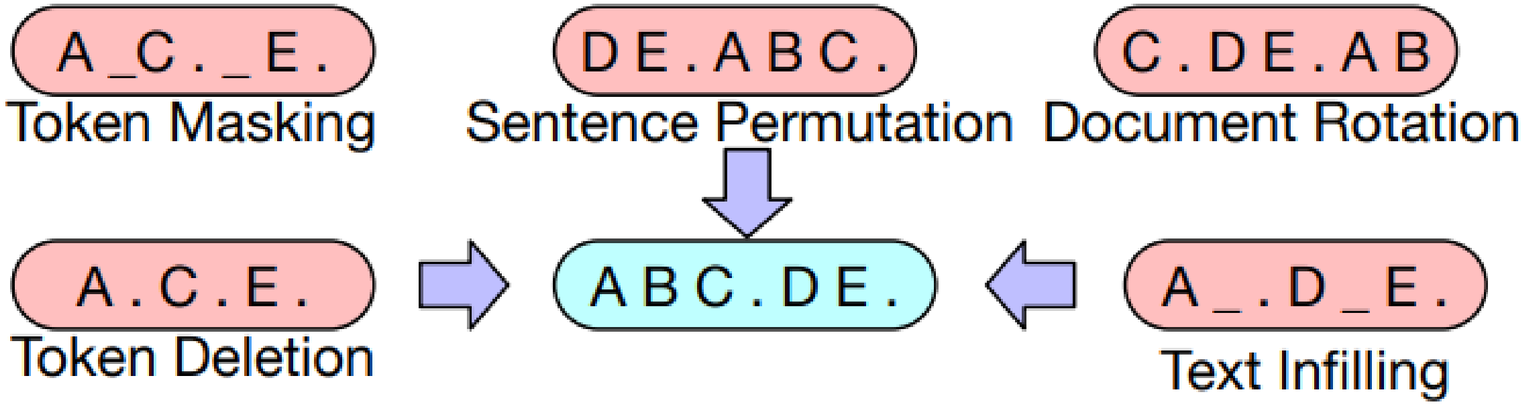
BART

그림 해석 주의

BERT와 BART는 모두 Transformer의 Positinoal Encoding이 아닌 Learnable Parameter를 사용한 Absolute Positional Embedding을 사용함.
또한 출력층은 Task에 따라 변할 수 있음.

Pre-training

Pretraining Objectives

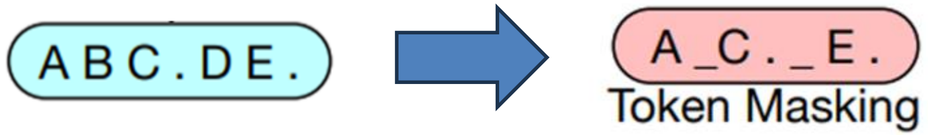


- BART is trained by corrupting documents and then optimizing a reconstruction loss (BERT)
- Pre-training Objective에 따라서 모델의 성능이 많이 달라진다는 사실이 최근 연구들에서 보고됨 (RoBERTa)
- BART는 총 5가지의 Noising 기법 사용, 성능 비교

Pre-training

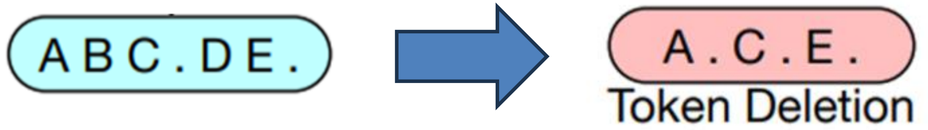
Token Masking

- Following BERT, random tokens are sampled and replaced with [MASK] elements
- 토큰 하나 마스킹



Token Deletion

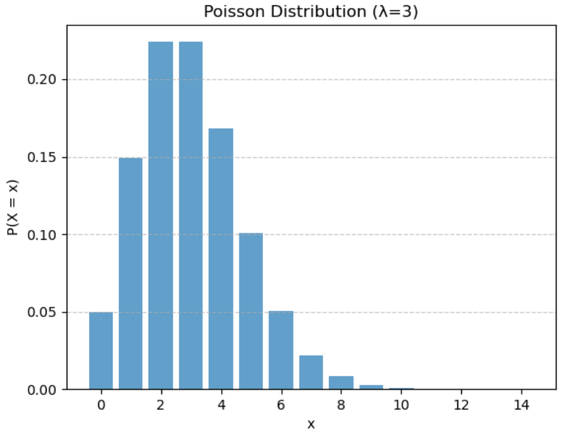
- Random tokens are deleted from the input
- 토큰 하나 삭제

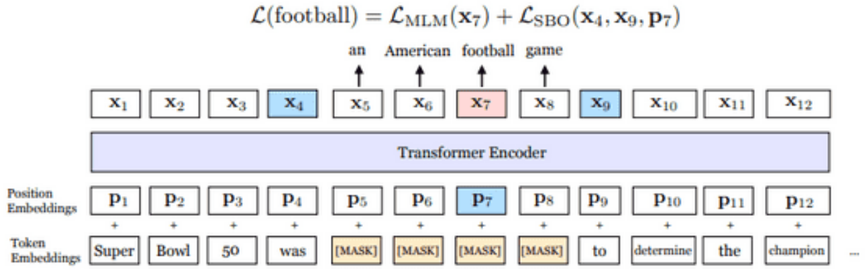


Pre-training

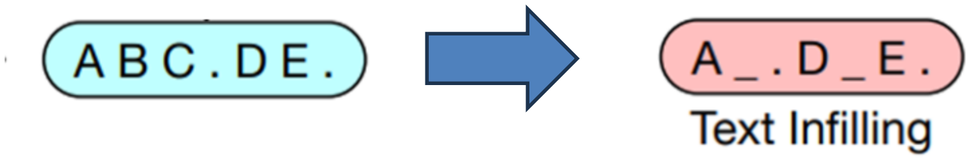
Text Infilling

- Text infilling is inspired by SpanBERT
- A number of text spans are sampled, with span lengths drawn from a Poisson distribution (λ = 3)
- Each span is replaced with a single [MASK] token
- 연속 토큰 여러 개(Span) -> Mask 하나





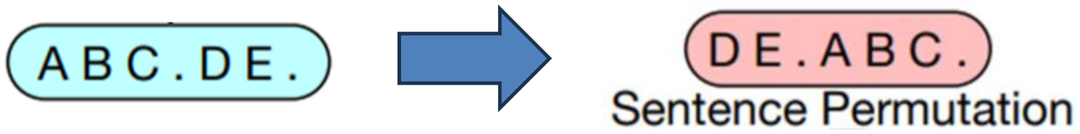
SpanBERT



Pre-training

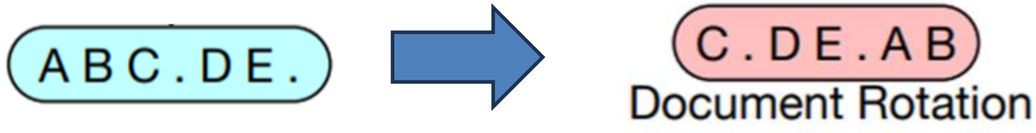
Sentence Permutation

- Sentences are shuffled in a random order.
- 문장 순서 바꾸기



Document Rotation

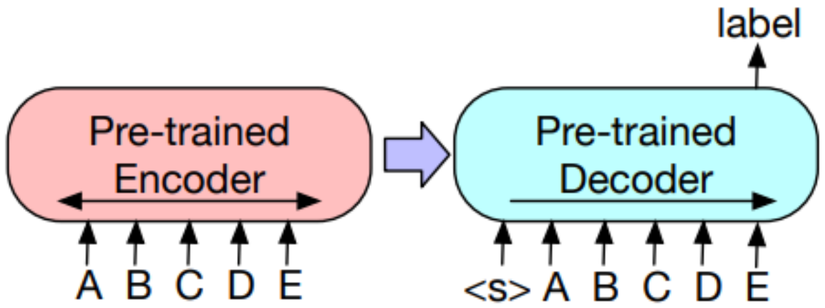
- A token is chosen uniformly at random.
- The document is rotated so that it begins with that token.
- 특정 토큰 기준으로 잘라서 순서 바꾸기



Fine-tuning

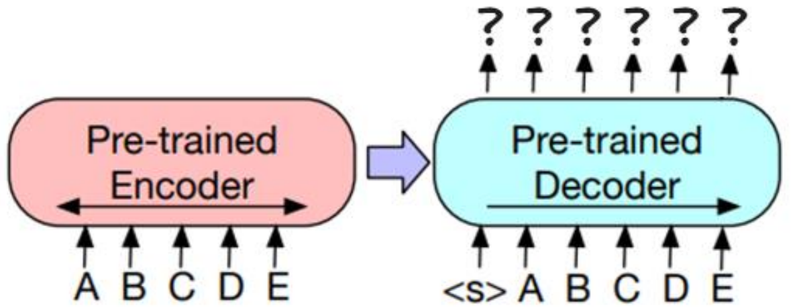
Sequence Classification

- BERT : 문장 처음의 CLS 토큰으로 Classification 진행
- BART : 문장 끝의 end 토큰으로 Classification 진행
- new multi-class linear classifier 사용



Token Classification

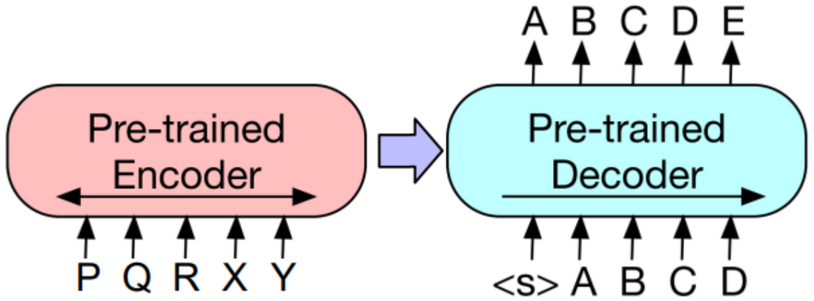
- 토큰 별로 분류가 필요할 땐 (e.g. SQuAD)
  BERT처럼 각 토큰의 last hidden state를 사용해
  분류 진행



Fine-tuning

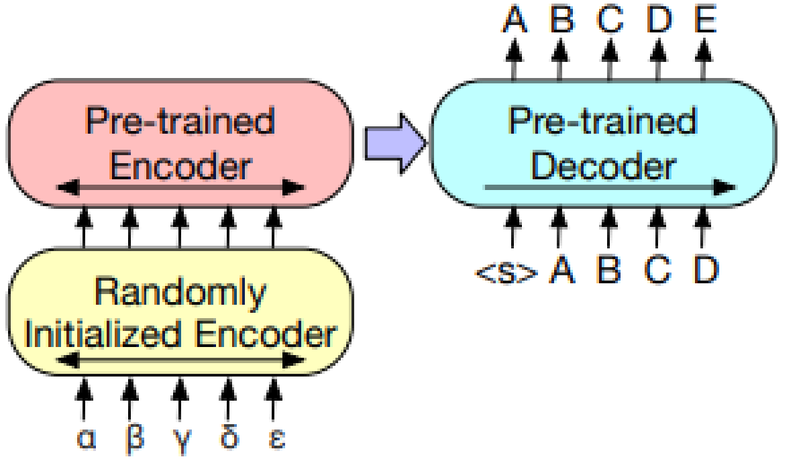
Sequence Generation

- Encoder Input으로 original Text, Decoder에서 generation



Machine Translation

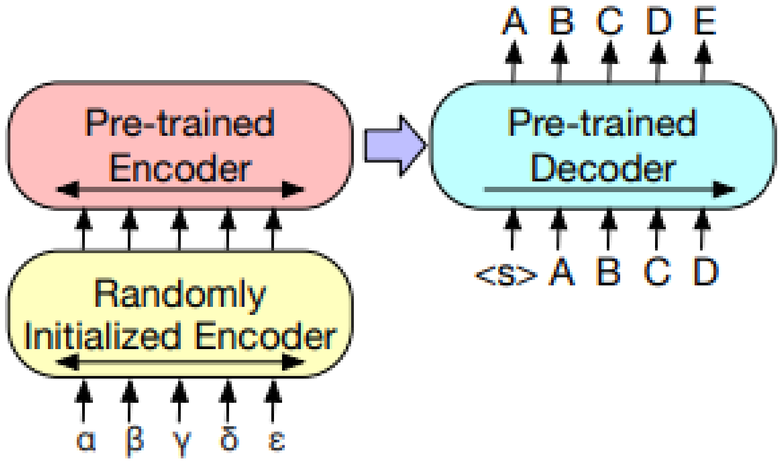
- 새로운 인코더를 추가하고 BART를 하나의 디코더로 사용



Fine-tuning

Machine Translation

- Pre-train된 BART 인코더의 임베딩 레이어를 삭제하고 Randomly Initialized Encoder를 추가
- 추가된 Encoder는 Non-English 언어의 단어 집합을 가짐
- 학습 단계 1. 새로운 인코더, Positional embeddings, BART인코더의 첫 레이어의 W_Q, W_V, W_K 만 학습시키고 나머지 파라미터는 모두 freeze
- 학습 단계 2. 적은 스텝으로 전체 파라미터 학습



In the first step, we freeze most of BART parameters and only update the randomly initialized source encoder, the BART positional embeddings, and the self-attention input projection matrix of BART's encoder first layer. In the second step, we train all model parameters for a small number of iterations.

Comparing Pre-training Objectives

Finding the Best Way for Pre-training

- 발표된 모델마다 사용한 학습 데이터, 학습 방법 등이 다 달라서 직접 비교하기 어려움
- 따라서 자체적으로 SOTA급 모델들을 재구성하여 최고의 Pre-training 기법을 찾고자 함
- Downstream Tasks에 Pre-train 모델들을 FineTuning 후 결과 비교

We aim, as much as possible, to control for differences unrelated to the pre-training objective

Comparing Pre-training Objectives

Model to Compare

- Language Model ~ GPT
- Permuted Language Model ~ XLNet
- Masked Language Model ~ BERT
- Multitask Masked Language Model ~ UniLM
- Masked Seq-to-Seq ~ MASS

Appendix 30~32 참고

Comparing Pre-training Objectives

Tasks to Compare

- SQuAD = an extractive question answering task
- MNLI = bitext classification task to predict whether one sentence entails another
- ELI5 = long-form abstractive question answering dataset
- XSum = a news summarization dataset with highly abstractive summaries
- ConvAI2 = a dialogue response generation task
- CNN/DM = a news summarization dataset, Almost extractive

Appendix 33 참고

Comparing Pre-training Objectives

	Model	SQuAD 1.1 F1	MNLI Acc	ELI5 PPL	XSum PPL	ConvAI2 PPL	CNN/DM PPL
BERT 논문	BERT Base (Devlin et al., 2019)	88.5	**84.3**	-	-	-	-
BERT	Masked Language Model	90.0	83.5	24.77	7.87	12.59	7.06
MASS	Masked Seq2seq	87.0	82.1	23.40	6.80	11.43	6.19
GPT	Language Model	76.7	80.1	**21.40**	7.00	11.51	6.56
XLNet	Permuted Language Model	89.1	83.7	24.03	7.69	12.23	6.96
UniLM	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
BEST	w/ Text Infilling + Sentence Shuffling	**90.8**	83.8	24.17	6.62	11.12	**5.41**

$$\mathrm{PPL}(X) = \exp\left\{-\frac{1}{t}\sum_{i}^{t}\log p_\theta(x_i|x_{<i})\right\}$$

Trained on Books and Wikipidea (BERT와 동일) for 1M steps

Trained on identical data using the same code-base, and fine-tuned with the same procedures -> Pre-train의 영향만 비교

Comparing Pre-training Objectives

Results

- Performance of pre-training methods varies significantly across tasks
- Token masking is crucial (GPT Limitation)
- Left-to-right pre-training improves generation (BERT Limitation)
- Bidirectional encoders are crucial for SQuAD
- The pre-training objective is not the only important factor (모델 구조, 훈련 방식 등도 중요)
- Pure language models perform best on ELI5
- BART achieves the most consistently strong performance

Large-scale Pre-training Objectives

- 최근 연구들은 Pre-train의 규모에 따라서도 성능 차이가 크다고 말함 (XLNet, RoBERTa)
- 따라서 저자들은 이전에 찾았던 최적의 Pre-train Objective와 함께 대규모의 Pre-train과 Fine-tuning을 한 번 더 진행해 BART의 성능을 비교함.
- 직접적인 비교를 위해 RoBERTa와 같은 대규모의 Pre-train 진행.

Large-scale Pre-training Objectives

Experimental Setup

- Use BART Large Model (Almost Same with BERT Large)
- Tokenizer : BPE from GPT2 -> Vocab size : 50257
- Batch size 8000, 500000 steps
- mask 30% of tokens in each document, and permute all sentences
- disabled dropout for the final 10% of training steps
- Training data from RoBERTa (BOOKCORPUS, CC-NEWS, OPENWEBTEXT, STORIES) = 160GB

Large-scale Pre-training Objectives

Discriminative Task

	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

Table 2: Results for large models on SQuAD and GLUE tasks. BART performs comparably to RoBERTa and XLNet, suggesting that BART's uni-directional decoder layers do not reduce performance on discriminative tasks.

- 동일 데이터로 학습한 RoBERTa와 거의 비슷한 성능 유지.
- BART는 디코더까지 결합됐음에도 Discriminative한 Task도 잘 수행하는 것을 확인.
- RoBERTa의 Pre-train Objective는 MLM, BART는 Text Infilling + Sentence Permutation이라 직접 비교 가능.

m/mm : matched/mismatched
Mcc : Matthews Correlation Coefficient

Large-scale Pre-training Objectives

Summarization

	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**

- CNN/DM : Extractive, Xsum : Abstractive
- BART는 두 요약에서 모두 SOTA 달성, 특히 Abstractive한 Task에서 훨씬 우월한 성능을 보임.

CNN/DM은 논문에선 Extractive라고 하지만 현재는 Extractive + Abstractive로 분류되고 있음

Lead-3 : 문서의 앞의 3문장. CNN/DM이나 Xsum 데이터셋 같은 경우 맨 앞의 3문장이 요약본인 성향이 강함.

Large-scale Pre-training Objectives

Dialogue & QA

	ConvAI2	
	Valid F1	Valid PPL
Seq2Seq + Attention	16.02	35.07
Best System	19.09	17.51
BART	**20.72**	**11.85**

	ELI5		
	R1	R2	RL
Best Extractive	23.5	3.1	17.5
Language Model	27.8	4.7	23.1
Seq2Seq	28.3	5.1	22.8
Seq2Seq Multitask	28.9	5.4	23.1
BART	**30.6**	**6.2**	**24.3**

- ConvAI2 : Extractive + Abstractive한 Dialogue Dataset
- ELI5 : Abstractive QA Dataset
- BART는 두 데이터셋에서 모두 SOTA 달성

Large-scale Pre-training Objectives

Translation

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

BLEU Score

학습 단계 1. 새로운 인코더, Positional embeddings, BART인코더의 첫 레이어의 W_Q, W_V, W_K 만 학습시키고 나머지 파라미터는 모두 freeze

학습 단계 2. 적은 스텝으로 전체 파라미터 학습

- Baseline : Transformer Large / Fixed BART : 학습단계1 / Tuned BART :학습단계2
- We use a 6-layer transformer source encoder to map Romanian into a representation that BART is able to de-noise into English
- We experiment on the original WMT16 Romanian-English augmented with back-translation data
- Preliminary results suggested that our approach was less effective without back-translation data, and prone to overfitting

Related Work

1. Bidirectional Encoder

- GPT – only leftward context -> 일부 task에 불리
- ELMo – left-only + right-only -> 단순히 Concat만 함 -> 양방향 의미 학습 X
- GPT2 – 모델이 아주 크면 Unsupervised Multi-task가 가능 / BART는 각 task에 Finetuning

Related Work

2. Auto-Regressive Decoder

- BERT – 양방향 의미 학습 / 문장 생성할 때 Auto-Regressive하지 않음 -> Less effective
- UniLM – Finetune BERT with an ensemble of masks
- MASS – Disjoint sets of tokens are fed into the encoder and decoder -> Less effective for Discriminative task
- XLNet – Extend BERT by predicting maksed tokens auto-regressively

Related Work

3. Machine Translation 등

- MASS, XLM - 번역하고자 하는 두 언어에 대해 Pretraining / 모든 언어에 대해 Pretrain은 불가능
- ELMo 등 사전 임베딩을 사용한 모델링 - decoder의 성능 향상에는 효과 적음

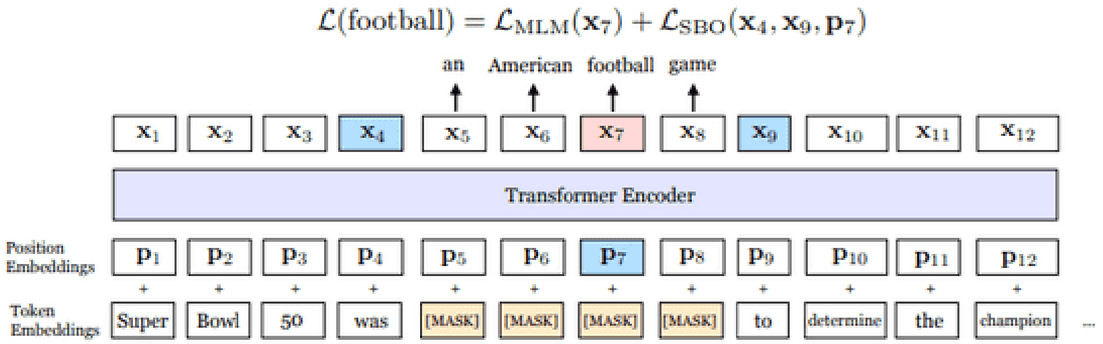
Conclusions

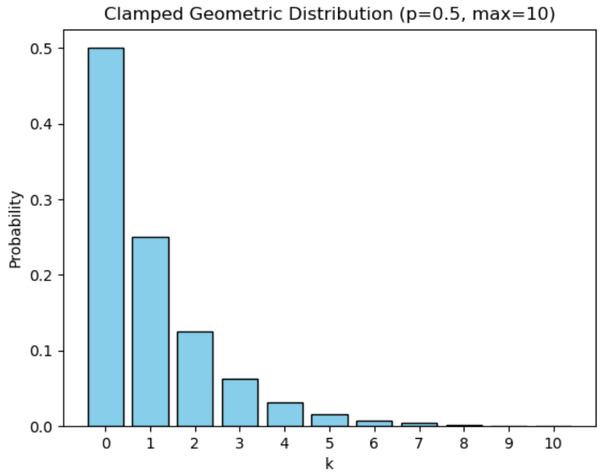
- We introduced BART, a pre-training approach that learns to map corrupted documents to the original.
- BART achieves similar performance to RoBERTa on discriminative tasks, while achieving new state-of-theart results on a number of text generation tasks (Abstractive QA, Summarization).
- Future work should explore new methods for corrupting documents for pre-training, perhaps tailoring them to specific end tasks.

감사합니다.

Appendix

SpanBERT





Text Infilling에서 BART와 SpanBERT의 차이점

1. BART는 포아송 분포 사용,
   SpanBERT는 Clamped Geometric Distribution 사용 -> Mask할 토큰 개수 정함

2. BART는 여러 토큰을 한 개의 [MASK]로
   SpanBERT는 토큰 각각을 [MASK]로

Appendix

UniLM (Multi-task Masked LM)

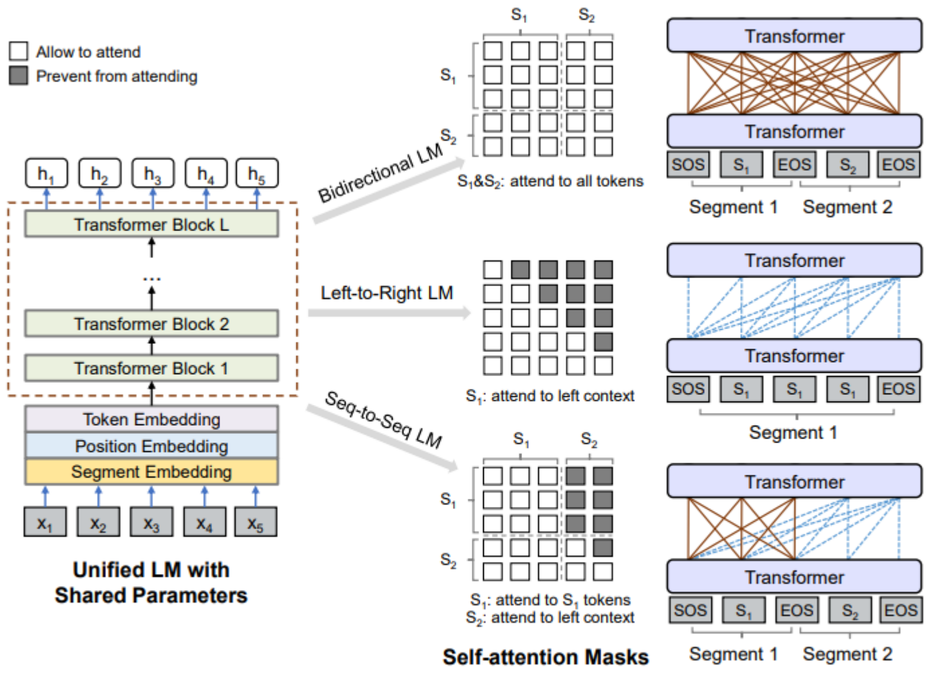


Figure 1: Overview of unified LM pre-training. The model parameters are shared across the LM objectives (i.e., bidirectional LM, unidirectional LM, and sequence-to-sequence LM). We use different self-attention masks to control the access to context for each word token. The right-to-left LM is similar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

- Unified Language Model
- Pretrained for Unidirectional LM, Bidirectional LM, Sequence-to-Sequence LM, Next Sentence Prediction

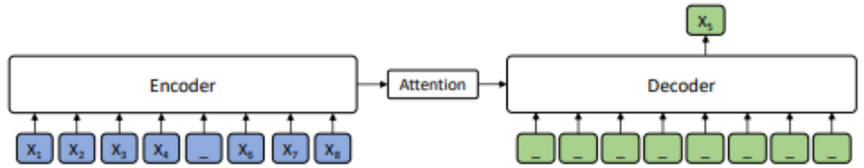
(With only Transformer Encoder)

- Why Multi-task?
- Pretrained for doing NLU + NLG in Single Architecture (Use different Attention Mask)
- vs GPT2

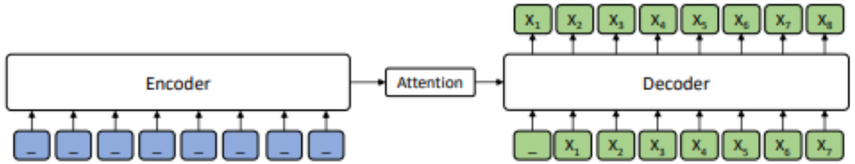
GPT2는 파인튜닝 안 해도 다양한 Task 수행할 수 있는 Zero-shot Multi-task (Zero-shot Learner). 하지만 UniLM은 Pre-train 후 파인튜닝까지 필요함. (추가적인 레이어는 필요 X)

Appendix

MASS (Masked Seq2Seq)



(a) Masked language modeling in BERT ($k = 1$)



(b) Standard language modeling ($k = m$)

- MAsked Sequence to Sequence pre-training
- Pretrained for Masked Spans, Good at Sequence Generation
- vs BERT MASK

BERT는 전체 토큰의 15%를 Mask, MASS는 전체 토큰의 50% 정도를 Mask. 그리고 한 시퀀스 안에서 Mask가 연속적임.

- SPANBERT나 BART의 Text Infilling이 이와 유사함
- MASK의 개수가 1개면 BERT, m개면 GPT의 언어 모델링과 비슷해짐. 따라서 MASS 방법론은 BERT와 GPT의 중간 개념.

Appendix

XLNet (Permuted LM)

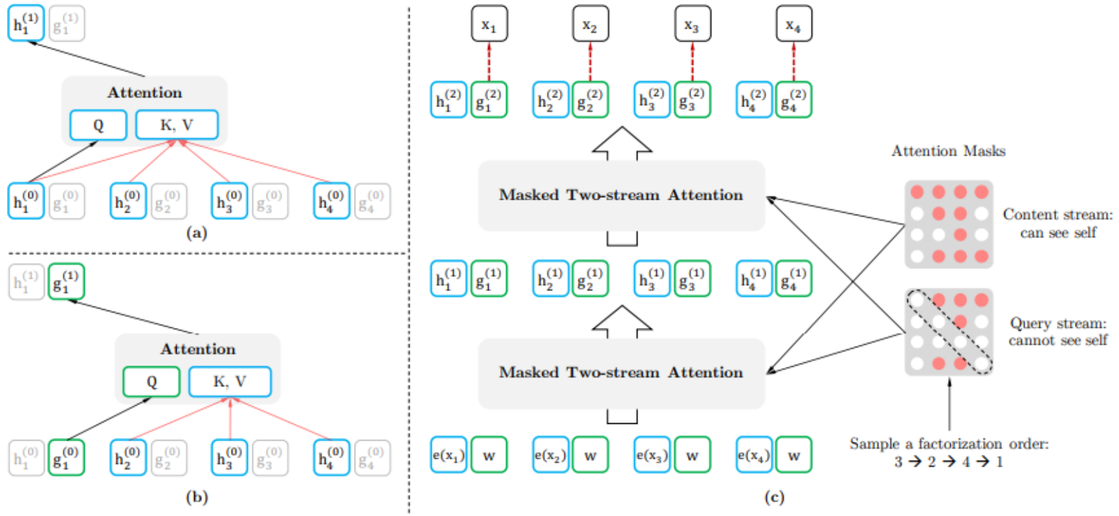


Figure 1: (a): Content stream attention, which is the same as the standard self-attention. (b): Query stream attention, which does not have access information about the content x_{z_t}. (c): Overview of the permutation language modeling training with two-stream attention.

- Permutation Language Modeling
- Generalized autoregressive pretraining method
- Use Two Way Self-Attention Stream (content, query)
- Transformer 디코더 아키텍처만 사용했으나 Sentence Permutation을 통해 Bidirectional한 문맥 학습
- BART의 Sentence Permutation과 유사함.
- Transformer-XL을 베이스로 발전.
- Relative Positional Embedding, segment-level recurrence 사용.

Appendix

Dataset Example

ELI5 (Explain Like I'm 5)

Input : *How do muscles grow?*

Output : *When you exercise, your muscles experience tiny tears. During recovery, the body repairs these tears by adding new muscle fibers, making your muscles larger and stronger over time*.

Xsum (Extreme Summarization)]

Input :*The city council has unveiled an ambitious plan to revitalize the downtown area by renovating old buildings, improving public transportation, and creating new public spaces. The initiative is expected to boost tourism and local economic growth amid rising unemployment.*

Output : *City council announces downtown revitalization plan to boost tourism and economy.*

https://huggingface.co/datasets/defunct-datasets/eli5

https://huggingface.co/datasets/EdinburghNLP/xsum?row=9

Appendix

Dataset Example

ConvAI2 (Conversation with AI)

Input : *User: Hi, how are you doing today?*
Bot: I'm doing well, thanks! What about you?
User: I'm a bit bored and looking for something fun to talk about.

Output : *Maybe we can chat about your hobbies. What do you usually enjoy doing in your free time?*

CNN/DM (CNN/Daily Mail)

Input : *News Article: powerful earthquake struck the coastal region early this morning, causing widespread damage to infrastructure and leaving hundreds injured. Emergency services have been mobilized, and residents are advised to stay alert as aftershocks are expected.*

Output : *Earthquake hits coastal region; emergency crews rescue hundreds amid aftershock warnings.*

https://huggingface.co/datasets/convai-challenge/conv_ai_2
https://huggingface.co/datasets/abisee/cnn_dailymail?row=6

Appendix

Evaluation Matrix

Rouge(Recall-Oriented Understudy for Gisting Evaluation)

$$F1 = \frac{2 \times Precision \times Recall}{(Precision + Recall)}$$

Precision = 정답 n-gram 수 / 생성 문장 길이
Recall = 정답 n-gram 수 / 정답 문장 길이

Rouge-n = n-gram 기준 F1 score
Rouge-L = n-gram 대신 LCS, *Longest Common Subsequence* 사용

https://medium.com/@eren9677/text-summarization-387836c9e178

BLEU

$$\text{BLEU} = BP \times \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

$$Brevity\ Penalty = \begin{cases} 1, & \text{if } c > r \\ e^{(1-r/c)}, & \text{if } c <= r \end{cases}$$

N = maximum n-gram size (보통 4)
P_n = precision in n-gram
w_n = weight for n-gram (보통 ¼)
C = 출력 문장 길이
R = 정답 문장 길이

https://towardsdatascience.com/foundations-of-nlp-explained-bleu-score-and-wer-metrics-1a5ba06d812b/

Appendix

- 굳이 Comparing Pre-training Objectives를 하면서 파인튜닝까지 진행한 이유?

Pre-training만 가지고는 각 Task에 적용된 실제 성능을 잘 보여주지 못함. 같은 시기에 나왔던 논문들 (특히 RoBERTa) 모두 파인튜닝 이후 성능을 비교하고 있지만, BART에서 강조하고 싶었던 것은 Pretrain 과정이었기 때문에 (주요 Contribution 중 하나) 제목은 Pretrain을 포함시키고 결과는 파인튜닝 이후 결과를 채택한 것.

- Bert base와 MLM의 성능 차이가 발생하는 이유?

본 논문에선 토큰의 15%를 MASK 처리하고 이를 예측하는 방식으로 MLM 모델 훈련을 진행했다고 언급함. (Original BERT는 전체 15% 선택 후 80% MASK, 10% 원래 토큰, 10% 랜덤 토큰) 구체적인 구조나 훈련 방식을 언급하지 않아서 Original BERT와 어떤 차이가 있는지 알 수 없음. 성능 향상을 위해 학습률과 정규화 등을 일부 조정했다는 언급은 있음. 정확한 이유는 알 수 없음.

- Sequence Generation Finetunig 추가 설명.

Abstractive summarization이나 QA에선 정답이 Encoder Input에 없기 때문에 Input을 조작해서 Decoder Output을 생성해야함. 이 과정은 Pre-train 에서 모델이 Noise된 Input을 Denoising하면서 학습하는 과정과 매우 유사하기 때문에 Generation도 잘 수행할 수 있다고 논문에서 언급함.

Appendix

- 4절에서 BART를 Pre-training 할 때 Token Masking이나 Token Deletion의 비율?

이 역시 논문에서 공개하질 않아서 정확한 수치는 알 수 없음. 하지만 Token Masking의 경우 BERT를 Follow했다는 언급으로 보아 15% 했을 가능성이 있음. Token Deletion은 아예 언급 없음. 이외에 Text Infiliing, Sentence Permutation, Document Rotation 등은 모든 Sequence에 대해서 진행한 것으로 보임.

- 5절에서 사용한 BART Large의 하이퍼파라미터

BERT base와 비슷하게 맞춘 BART base의 하이퍼파라미터로부터 BART Large의 하이퍼파라미터 또한 BERT Large와 비슷할 것이라고 추론할 수 있음. 논문에선 총 24개(12 + 12)의 레이어, 1024의 hidden size만 언급했음. 이는 BERT Large와 동일함. 따라서 어텐션 헤드와 d_ff 또한 BERT Large와 동일하게 16개, 4096이라고 추측가능. (transformers library의 BART로 확인)

- 4절과 5절에서 사용한 평가 지표가 다른 이유?

4절은 BART에게 좋은 Pre-train 기법을 찾는 느낌이 강하고, 5절은 4절을 토대로 BART의 진짜 성능을 확인하는 느낌이 강함.
4절에서 저자들은 To most directly compare our models on their ability to model their fine-tuning objective 라며 4절의 평가지표로 ppl을 사용한 이유를 설명함. 따라서 4절은 fine-tuning objective를 잘 학습시킬 수 있는 Pre-training 기법을 찾기 위해 ppl을, 5절은 잘 훈련된 BART가 SOTA급 성능을 보이는 성과를 달성했음을 보이기 위해 ACC 등의 지표를 선택한 것으로 보임.

Appendix

4절 실험 내용 보충

For the Permuted LM, Masked LM and Multitask Masked LM, we use two-stream attention (Yang et al., 2019) to efficiently compute likelihoods of the output part of the sequence (using a diagonal self-attention mask on the output to predict words left-to-right).

We experiment with (1) treating the task as a standard sequence-to-sequence problem, where the source input to the encoder and the target is the decoder output, or (2) adding the source as prefix to the target in the decoder, with a loss only on the target part of the sequence. We find the former works better for BART models, and the latter for other models. To most directly compare our models on their ability to model their fine-tuning objective (the log likelihood of the human text), we report perplexity in Table 1.

Appendix

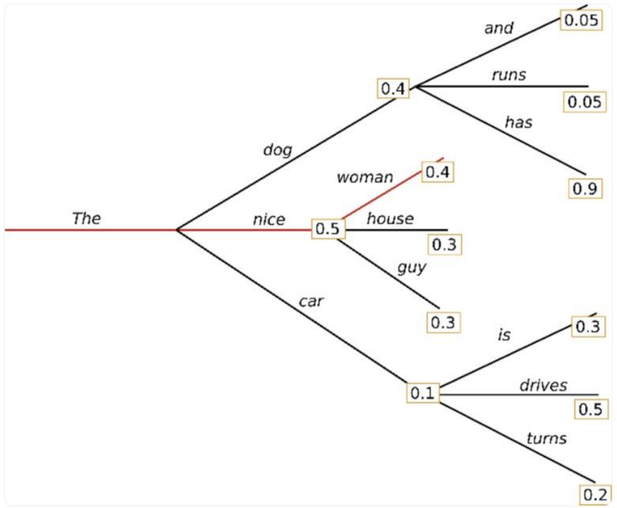
5절 실험 내용 보충

We also experiment with several text generation tasks. BART is fine-tuned as a standard sequence-to-sequence model from the input to the output text. During finetuning we use a label smoothed cross entropy loss (Pereyra et al., 2017), with the smoothing parameter set to 0.1. During generation, we set beam size as 5, remove duplicated trigrams in beam search, and tuned the model with min-len, max-len, length penalty on the validation set (Fan et al., 2017).

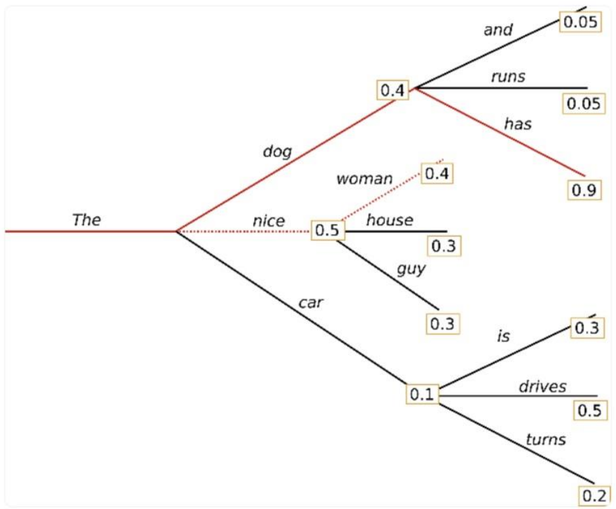
Appendix

Beam Search vs Greedy Search

Greedy



Beam



https://heidloff.net/article/greedy-beam-sampling/