Unified Language Model Pre-training for Natural Language Understanding and Generation

- 저자:
  - Li Dong∗ Nan Yang∗ Wenhui Wang∗ Furu Wei∗ † Xiaodong Liu Yu Wang Jianfeng Gao Ming Zhou Hsiao-Wuen Hon (**Microsoft Research**)
- 발표:
  - Presenter: 윤주성
  - Date: 191221

Who is an Author?

- 일단 쓴 논문들에 대한 기본 인용수가 높다
- 감성분석, MRC, Summarization 등 태스크를 가리지 않고, EMNLP, AAAI, ACL 등에 논문을 엄청 많이 냄.. 그냥 고수
- 이 논문은 NeurIPS 2019
- 191219 기준으로 인용수 26회

Li Dong

Microsoft Research
Verified email at microsoft.com - Homepage
Natural Language Processing

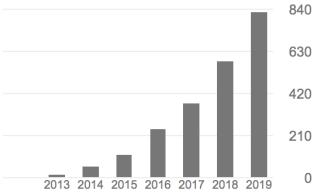
FOLLOW

GET MY OWN PROFILE

TITLE	CITED BY	YEAR
Long Short-Term Memory-Networks for Machine Reading J Cheng, L Dong, M Lapata Conference on Empirical Methods in Natural Language Processing (EMNLP)	372	2016
Adaptive Recursive Neural Network for Target-dependent Twitter Sentiment Classification L Dong, F Wei, C Tan, D Tang, M Zhou, K Xu The 52nd Annual Meeting of the Association for Computational Linguistics …	277	2014
Moodlens: an emoticon-based sentiment analysis system for chinese tweets J Zhao, L Dong, J Wu, K Xu Proceedings of the 18th ACM SIGKDD international conference on Knowledge …	259	2012
Language to Logical Form with Neural Attention L Dong, M Lapata The 54th Annual Meeting of the Association for Computational Linguistics …	232	2016
Question Answering over Freebase with Multi-Column Convolutional Neural Networks L Dong, F Wei, M Zhou, K Xu The 53rd Annual Meeting of the Association for Computational Linguistics and …	218	2015
Ranking with Recursive Neural Networks and Its Application to Multi-document Summarization Z Cao, F Wei, L Dong, S Li, M Zhou AAAI Conference on Artificial Intelligence	149	2015
Unraveling the origin of exponential law in intra-urban human mobility X Liang, J Zhao, L Dong, K Xu Scientific Reports	106	2013
Coarse-to-Fine Decoding for Neural Semantic Parsing L Dong, M Lapata Proceedings of the 56th Annual Meeting of the Association for Computational …	78	2018
Proactive Resource Management for LTE in Unlicensed Spectrum: A Deep Learning Perspective U Challita, L Dong, W Saad IEEE Transactions on Wireless Communications	75 *	2018
Unsupervised Word and Dependency Path Embeddings for Aspect Term Extraction Y Yin, F Wei, L Dong, K Xu, M Zhang, M Zhou Proceedings of the 25th International Joint Conference on Artificial …	70	2016
Adaptive multi-compositionality for recursive neural models with applications to sentiment analysis	62	2014

Cited by

	All	Since 2014
Citations	2239	2213
h-index	19	19
i10-index	21	21



Co-authors

- Furu Wei — Senior Principal Research Mana…
- Ming Zhou (周明) — Assistant Managing Director at …
- Mirella Lapata — School of Informatics, Edinburgh…
- Jianpeng Cheng — University of Edinburgh
- Wenhui Wang — Microsoft Research
- Ratish Puduppully — School of Informatics, University …
- Jonathan Mallinson — PhD candidate, The University of…
- Siva Reddy — Postdoc, Stanford University
- Jianfeng Gao — Microsoft Research, Redmond

느낀점

- NLG에서 SOTA를 꽤 찍었는데 방식이 좀 신기
- shared param (같은 모델)로 NLU와 NLG를 할 수 있다는게 가장 큰 장점
- masking으로 장난치면서(?) 모델을 발전시킨건 어쩌면 자연스러운 수순인듯
- 1st segment에서 passage와 answer를 concat하거나 conversation history를 concat 방식으로 집어넣는데, 잘되는게 좀 신기하긴함
- T5가 살아남을지 이 친구가 더 개량되서 살아남을지 궁금
- seq2seq LM을 fine-tuning하는 방법이 좀 신선했음 당연히 left-to-right 방식으로 teacher forcing할줄 알았는데.. ㅎㅎ

Abstract

- UNIfied pre-trained Language Model (UNILM) 이라는 모델을 제안함
- NLU와 NLG를 모두 할 수 있게 fine-tune이 가능한 모델임
- 3가지 LM task로 pretraining함
  - unidirectional
  - bidirectional
  - sequence-to-sequence prediction
- shared Transformer network와 specific self-attention masks(`to control what context the prediction conditions on`)를 통해서 unified modeling을 함
- UNILM은 GLUE, SQuAD 2.0, CoQA task도 좋은 성능을 낼 수 있고 NLG dataset에서도 5개 부분에서 SOTA를 기록함
  - CNN/DailyMail abstractive summarization ROUGE-L 값은 40.51을 기록함 (2.04 개선)
  - Gigaword abstractive summarization ROUGE-L은 35.75 기록함 (0.86 개선)
  - CoQA generative question answering F1 score는 82.5 기록함 (37.1 개선)
  - SQuAD question generation BLEU-4는 22.12 기록함 (3.75 개선)
  - DSTC7 document-grounded dialog response generation NIST-4는 2.67 기록함 (사람이 한 점수는 2.65)
- code & pretrained models: https://github.com/microsoft/unilm

1. Introduction

- LM pre-training은 다양한 NLP task에서 SOTA를 찍을 수 있게 해줌 (substantially advanced)
- Pre-trained LMs은 contextualizaed text representations을 단어 주변의 context를 활용해서 단어를 예측함으로써 학습하고 이때 대량의 text 데이터를 사용함
- Pre-trained LMs은 Downstream task에 대해서 fine-tune 해서 쓸수 있음
- pre-training LMs의 타입에 따라 다양한 prediction task와 training objectives가 사용되왔음
  - ELMo의 경우엔 2가지의 unidirectional LMs을 사용함
    - left-to-right와 right-to-left로 배우기 때문임
  - GPT의 경우엔 left-to-right 임
  - BERT의 경우는 bidrectional LM임

	ELMo	GPT	BERT	UNILM
Left-to-Right LM	✓	✓		✓
Right-to-Left LM	✓			✓
Bidirectional LM			✓	✓
Sequence-to-Sequence LM				✓

Table 1: Comparison between language model (LM) pre-training objectives.

1. Introduction

- BERT가 성능이 매우 좋은 모델이지만 특성상 NLG task에 적용이 어려움
- 본 연구에서는 UNIfied pre-trained Language Model (UNILM)을 제안하면서 모델을 NLU와 NLG task에 모두 적용하고자함
- UNILM은 multi-layer Transformer network이고 pre-train을 하면서 동시에 3가지 타입의 unsupervised language modeling objectives에 대해 학습함

1. Introduction

- 특별히 몇가지의 cloze tasks(빈칸 채우기)를 디자인했고 거기서 보는 context는 다음같음
  - unidirectional LM
    - left-to-right unidirectional LM
      - context는 왼쪽에 있는 모든 단어들이 됨
    - right-to-left unidirectional LM
      - 반대로 오른쪽에 있는 모든 단어들이 됨
  - bidirectional LM
    - context는 왼쪽 오른쪽 방향을 모두 포함하는 단어 주변의 모든 단어들
  - sequence-to-sequence LM
    - context는 encoder의 정보와 target sequence에서 예측해야되는 단어의 앞에 있는 모든 단어들
- BERT와 비슷하게 pre-trained UNILM은 fine-tuning이 가능하지만(`with additional task-specific layers if necessary`), NLU task가 메인인 BERT와 다르게 UNILM은 다른 종류의 LMs의 context를 결합하기 위해서 different self-attention masks를 사용하는 것으로 설계되었고 이는 NLU와 NLG task 모두를 가능하게 해줌

Backbone Network	LM Objectives of Unified Pre-training	What Unified LM Learns	Example Downstream Tasks
Transformer with shared parameters for all LM objectives	Bidirectional LM	Bidirectional encoding	GLUE benchmark Extractive question answering
	Unidirectional LM	Unidirectional decoding	Long text generation
	Sequence-to-Sequence LM	Unidirectional decoding conditioned on bidirectional encoding	Abstractive summarization Question generation Generative question answering

Table 2: The unified LM is jointly pre-trained by multiple language modeling objectives, sharing the same parameters. We fine-tune and evaluate the pre-trained unified LM on various datasets, including both language understanding and generation tasks.

1. Introduction

- 제안하는 UNILM은 3가지 장점이 있음
  - the unified pre-training procedure는 single Transformer LM이 다양한 타입의 LMs을 위한 모델의 parameters와 architecture를 공유할 수 있게 해줌 (`alleviating the need of separately training and hosting multiple LMs`)
  - context를 다르게 잡아내는 different LM objective를 학습하면 any sing LM task에서 발생할 수 있는 overfitting을 막아주기 때문에, 이러한 parameter sharing은 학습된 text representations을 더 general하게 해줌
  - UNILM은 sequence-to-sequence LM을 사용하는데, 이는 NLG를 위한 자연스러운 선택이됨 (`such as abstractive summarization and question generation`)
  - 실험결과를 보면, bidirectional encoder를 사용한 제안모델이 GLUE에서 BERT와 비교할만하고 two extractive QA task에서도 좋은 결과를 냄 (NLU, NLG 둘다 잘한다)

2. Unified Language Model Pre-training

- 주어진 input sequence $x = x_1 \cdots x_n$에 대해서 UNILM은 각 token에 대해서 contextualized vector representation을 얻음
- pre-training 단계에서 shared Transformer network를 `unidirectional LM, bidirectional LM, and sequence-to-sequence LM` 라는 LM objectives로 학습함
- 이를 위해서 self-attention에 대해 different masks 를 도입함 (`use masking to control how much context the token should attend`)
- pre-training 끝나면 downstream task를 위해 task-specific data로 fine-tuning해서 쓸 수 있음

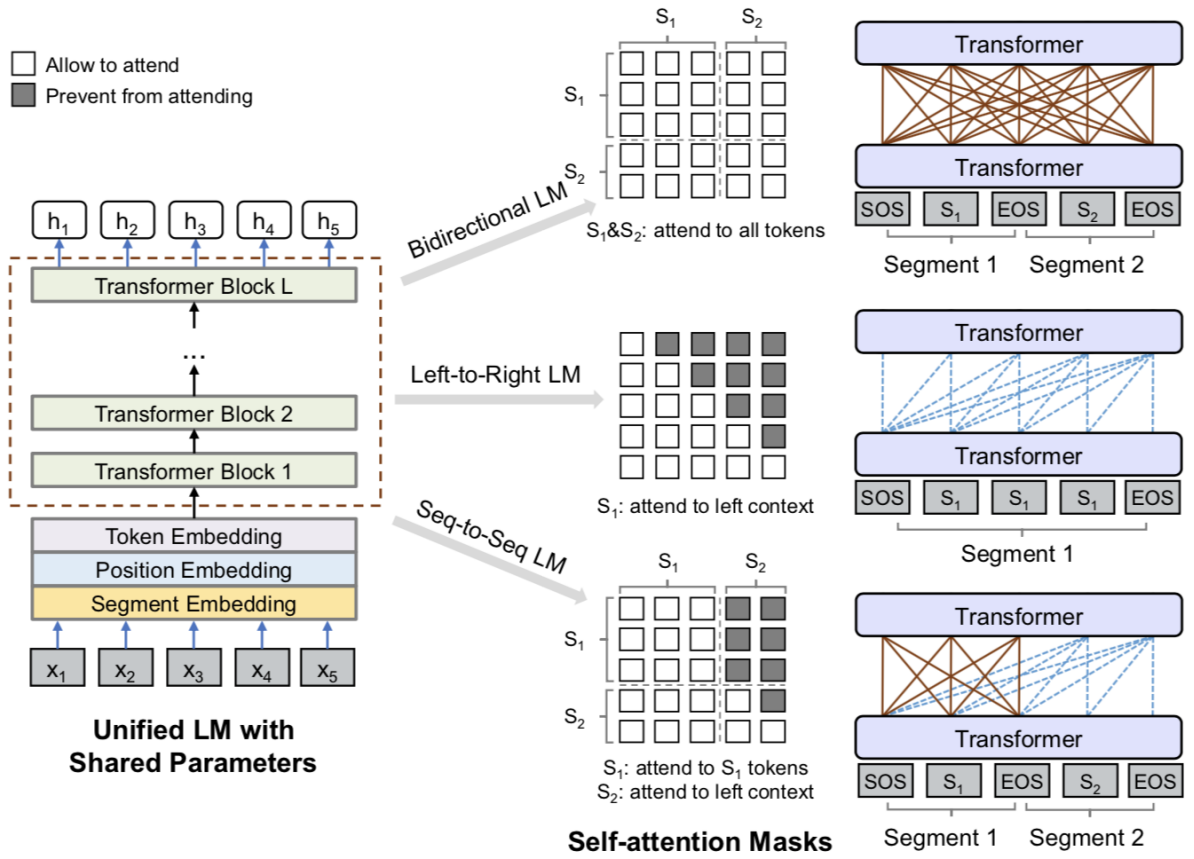


Figure 1: Overview of unified LM pre-training. The model parameters are shared across the LM objectives (i.e., bidirectional LM, unidirectional LM, and sequence-to-sequence LM). We use different self-attention masks to control the access to context for each word token. The right-to-left LM is similar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

2.1 Input Representation

- Special token 추가함
  - [SOS]: start-of-sequence
  - [EOS]: end-of-sequence
- input representation은 BERT 형식을 따름
- WordPiece로 토큰화됨
- LM 종류에 따라 segment가 달라짐 (Figure 1 참고)

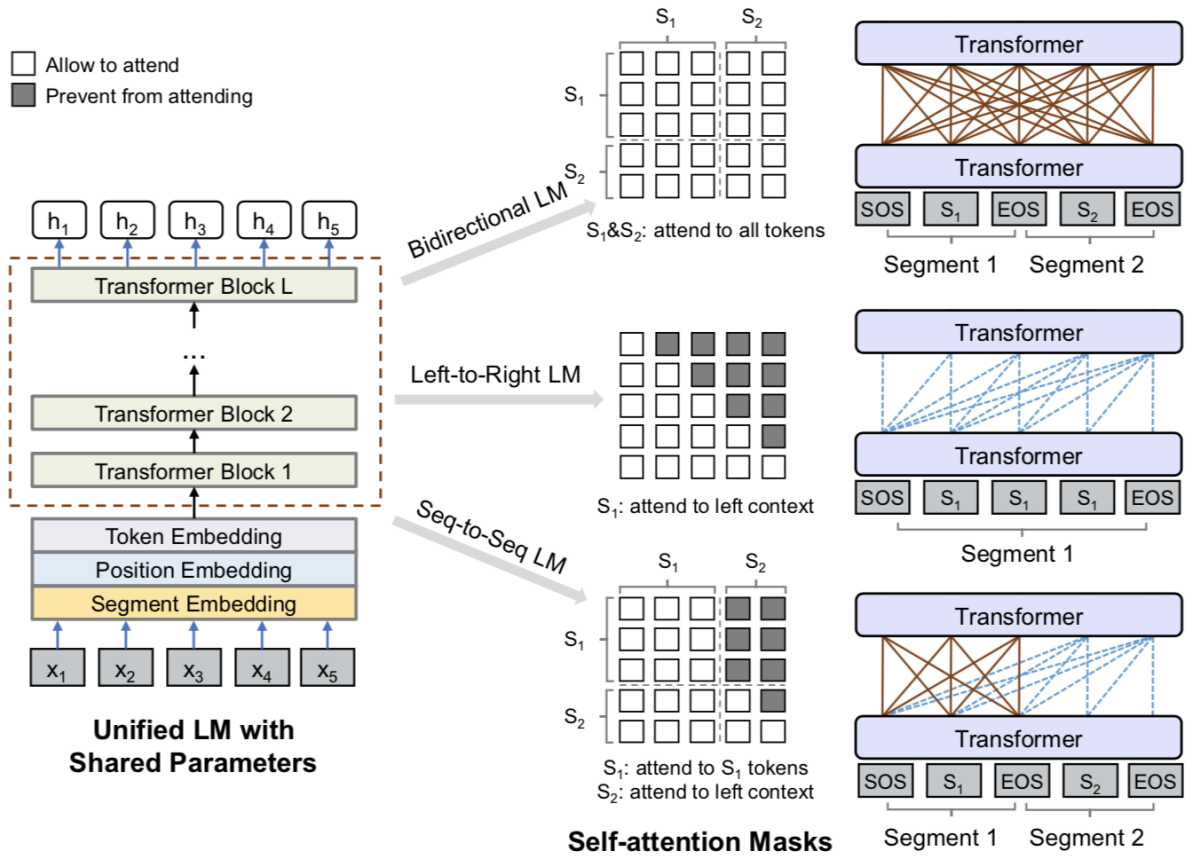


Figure 1: Overview of unified LM pre-training. The model parameters are shared across the LM objectives (i.e., bidirectional LM, unidirectional LM, and sequence-to-sequence LM). We use different self-attention masks to control the access to context for each word token. The right-to-left LM is similar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

2.2 Backbone Network: Multi-Layer Transformer

- input vectors를 $H^0 = [x_1, \cdots, x_n]$ 로 나타낼 수 있고 L-layer의 Transformer를 통해 different levels에서의 contextual representation으로 인코딩하면 $H^l = [h_1^l, \cdots, h_n^l]$ 으로 나타낼 수 있음
- $\mathbf{H}^l = \mathrm{Transformer}_l(\mathbf{H}^{l-1}), l \in [1, L]$ 로 표현 가능함
- l 번째 layer에서 self-attention Head $\mathbf{A}_l$ 의 output은 다음과 같이 계산됨

$$\mathbf{Q} = \mathbf{H}^{l-1}\mathbf{W}_l^Q, \quad \mathbf{K} = \mathbf{H}^{l-1}\mathbf{W}_l^K, \quad \mathbf{V} = \mathbf{H}^{l-1}\mathbf{W}_l^V$$

$$\mathbf{M}_{ij} = \begin{cases} 0, & \text{allow to attend} \\ -\infty, & \text{prevent from attending} \end{cases}$$

$$\mathbf{A}_l = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}} + \mathbf{M}\right)\mathbf{V}_l$$

- 이전 layer의 output인 $H^{l-1} \in R^{n \times d_h}$ 은 parameter matrices $W_l^Q, W_l^K, W_l^V \in R^{d_h \times d_k}$ 에 의해 queries, keys, vlaues로 linearly projected 됨
- mask matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ 는 token의 contextualized representation을 계산하기 위해 어떤 token들에 attention할지를 결정하기 위해 사용됨

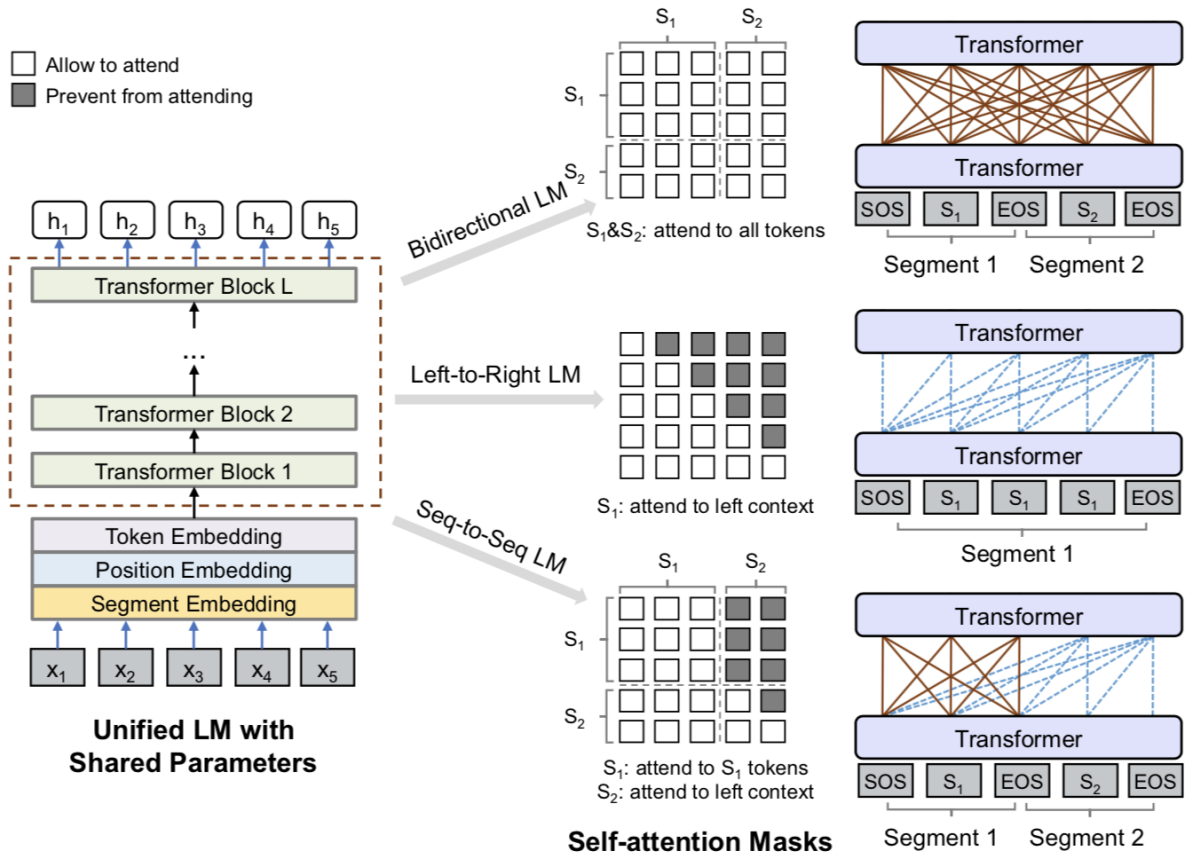


Figure 1: Overview of unified LM pre-training. The model parameters are shared across the LM objectives (i.e., bidirectional LM, unidirectional LM, and sequence-to-sequence LM). We use different self-attention masks to control the access to context for each word token. The right-to-left LM is similar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

2.3 Pre-training Objectives

- The parameters of UNILM are learned to minimize the cross-entropy loss computed using the predicted tokens and the original tokens
- LM 종류
  - Unidirectional LM:
    - use both left-to-right and right-to-left LM objectives
    - For instance, to predict the masked token of "$x_1 x_2$ [MASK] x_4", only tokens x_1, x_2 and itself can be used. This is done by using a triangular matrix for the self-attention mask M
  - Bidirectional LM:
    - the self-attention mask M is a zero matrix, so that every token is allowed to attend across all positions in the input sequence.
  - Sequence-to-Sequence LM:
    - the tokens in the first (source) segment can attend to each other from both directions within the segment, while the tokens of the second (target) segment can only attend to the leftward context in the target segment and itself, as well as all the tokens in the source segment
    - "[SOS] $t_1 t_2$ [EOS] $t_3 t_4 t_5$ [EOS]" into the model. While both t1 and t2 have access to the first four tokens, including [SOS] and [EOS], t4 can only attend to the first six tokens
    - sequence-to-sequence LM의 경우 bidirectional encoder와 unidirectional decoder를 학습한다고 보면 됨
- Next Sentence Prediction:
  - Bidirectional LM에 대해서는 NSP를 적용함

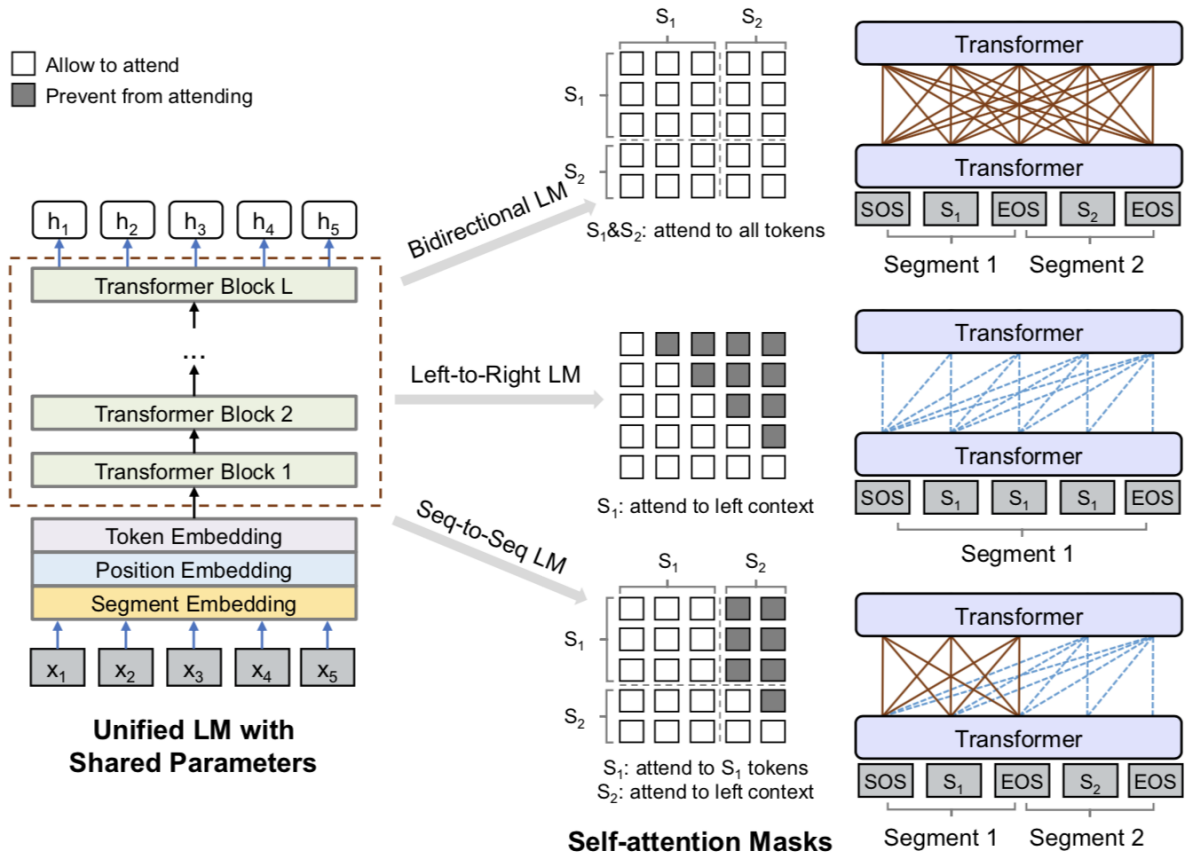


Figure 1: Overview of unified LM pre-training. The model parameters are shared across the LM objectives (i.e., bidirectional LM, unidirectional LM, and sequence-to-sequence LM). We use different self-attention masks to control the access to context for each word token. The right-to-left LM is similar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

2.4 Pre-training Setup

- one training batch당, 1/3은 bidirectional LM objective, 1/3은 seq2seq LM objective, 나머지 1/3은 unidirectional LM objective (left-to-right, right-to-left)를 사용함
- 모델의 구조는 $BERT_{LARGE}$와 같음
  - gelu activation
  - 24-layer transformer (340M params)
    - with 1,024 hidden size
    - 16 attention heads
  - `weight matrix of the softmax classifier is tied wtih token embeddings`
  - $BERT_{LARGE}$의 weight로 initialize함
- Corpus는 English Wikipedia와 BookCorpus 사용

2.4 Pre-training Setup

- Vocab size: 28,996
- Maximum lengths of input seq: 512
- Masking Prob: 15%
  - 80%: [MASK]
  - 10%: random token
  - 10%: original token
- 마스킹할때 80%는 one token으로 나머지 20%는 bigram or trigram으로 마스킹함
- Optimizer:
  - Adam: $\beta_1 = 0.9, \beta_2 = 0.999$
  - lr: 3e-5
  - warm up: first 40,000 steps (and linear decay)
  - weight decay: 0.01
- Dropout rate: 0.1
- Batch size: 330 (~~특이하네~~)
- pre-training procedure runs: 770,000 steps
- time: 7 hours for 10,000 steps
- GPUs: 8 Nvidia Telsa V100 32GB

2.5 Fine-tuning on Downstream NLU and NLG Tasks

- NLU task에 대해서는 BERT처럼 fine-tuning하면 됨
  - [SOS] 토큰에 대한 vector $\mathbf{h}_1^L$에 randomly initialized softmax classifier를 붙임
  - $softmax(h_1^L W^C),\ \text{where}\ W^C \in R^{d_h \times C}$ (C는 카테고리 개수(클래스 개수)임)
- NLG task에 대해서는 seq2seq task와 비슷함
  - Notation
    - S1: source sequence
    - S2: target sequence
  - 하나로 합침(pack)
    - "[SOS] S1 [EOS] S2 [EOS]"
  - fine-tuning 방법:
    - target sequence에 있는 토큰을 특정 비율로 랜덤하게 마스킹한 후에 맞추도록 학습함(`masking some percentage of tokens in the target sequence at random, and learning to recover the masked words.`)
    - The training objective is to maximize the likelihood of masked tokens given context
    - 생성을 끝내는 의미로도 사용되는 [EOS]에 대해서도 마스킹을 하는게 좋은데, 그 이유는 모델이 언제 generation process를 끝내야되는지도 학습할 수 있기 때문임(`It is worth noting that [EOS], which marks the end of the target sequence, can also be masked during fine-tuning, thus when this happens, the model learns when to emit [EOS] to terminate the generation process of the target sequence`)
    - (~~근데 이렇게 finetuning하면 fully generation하는게 아닌데 잘 되나..~~)

3. Experiments

- NLU는 GLUE, extractive question answering으로 평가
- NLG는 abstractive summarization, question generation, generative question answering, and dialog response generation등으로 평가

3.1 Abstractive Summarization

- Dataset:
  - non-anonymized version of the CNN/DailyMail dataset
  - Gigaword for model fine-tuning and evaluation
- Input representation:
  - by concatenating document (the first segment) and summary (the second segment)
- Finetune process:
  - fine-tune our model on the training set for 30 epochs
  - reuse most hyper-parameters from pre-training
    - Masking prob: 0.7 (~~되게 높아졌기 때문에 generation이 가능한거군..!~~)
    - label smoothing with rate of 0.1
  - For CNN/DailyMail:
    - batch size to 32, and maximum length to 768
  - For Gigaword:
    - batch size to 64, and maximum length to 256
- Decoding:
  - beam search with beam size of 5
  - remove duplicated trigrams in beam search
- The input document is truncated
  - first 640 for CNN/DailyMail
  - first 192 tokens for Gigaword

3.1 Abstractive Summarization

- Evaluation
  - Metric:
    - F1 version of ROUGE
  - Table 3는 CNN/DailyMail 에 대한 평가이고 Table 4는 Gigaword에 대한 평가임
  - Other Models
    - LEAD-3 (Baseline): 첫 3문장을 문서의 summary로 보는 것
    - PGNet: Pointer-generator network 기반의 seq2seq 모델 (copy mechanism)
    - S2S-ELMo: pre-trained ELMo representation을 통해 seq2seq 모델을 개량한 것
    - Bottom-Up: salient phrases를 선택하는 content selector를 사용

	RG-1	RG-2	RG-L
Extractive Summarization			
LEAD-3	40.42	17.62	36.67
Best Extractive [27]	43.25	**20.24**	39.63
Abstractive Summarization			
PGNet [37]	39.53	17.28	37.98
Bottom-Up [16]	41.22	18.68	38.34
S2S-ELMo [13]	41.56	18.94	38.47
UNILM	**43.33**	20.21	**40.51**

Table 3: Evaluation results on CNN/DailyMail summarization. Models in the first block are extractive systems listed here for reference, while the others are abstractive models. The results of the best reported extractive model are taken from [27]. RG is short for ROUGE.

	RG-1	RG-2	RG-L
10K Training Examples			
Transformer [43]	10.97	2.23	10.42
MASS [39]	25.03	9.48	23.48
UNILM	**32.96**	**14.68**	**30.56**
Full Training Set			
OpenNMT [23]	36.73	17.86	33.68
Re3Sum [4]	37.04	19.03	34.46
MASS [39]	37.66	18.53	34.89
UNILM	**38.45**	**19.45**	**35.75**

Table 4: Results on Gigaword abstractive summarization. Models in the first block only use 10K examples for training, while the others use 3.8M examples. Results of OpenNMT and Transformer are taken from [4, 39]. RG is short for ROUGE.

3.2 Question Answering (QA)

- Extractive QA: 답이 passage안의 text span라고 가정
  - bidirectional encoder를 사용해서 접근함
  - experiments
    - SQuAD 2.0 (Stanford Question Answering Dataset)
      - hyper params
        - epoch: 3
        - batch size: 24
        - max len: 384
    - CoQA (Conversational Question Answering)
      - SQuAD랑은 좀 다른데, 대화 내역에 기반한 답변을줘야 함
      - 답변은 free-form texts 형태임 (yes/no answer 포함)
      - concatenate the question-answer histories to the first segment
      - for yes/no questions, we use the final hidden vector of the [SOS] token to predict whether the input is a yes/no question, and whether the answer is yes or no
      - for other examples, we select a passage subspan
      - hyper params
        - epoch: 2 / batch size: 16 / max len: 512
  - 결과를 보면 EM (Exact Match)이나 F1 모두 UNILM이 젤 높음

	EM	F1
RMR+ELMo [20]	71.4	73.7
$\text{BERT}_{\text{LARGE}}$	78.9	81.8
UNILM	**80.5**	**83.4**

Table 5: Extractive QA results on the SQuAD development set.

	F1
DrQA+ELMo [35]	67.2
$\text{BERT}_{\text{LARGE}}$	82.7
UNILM	**84.9**

Table 6: Extractive QA results on the CoQA development set.

	F1
Seq2Seq [35]	27.5
PGNet [35]	45.4
UNILM	**82.5**

Table 7: Generative QA results on the CoQA development set.

3.2 Question Answering (QA)

- Generative QA: 답을 즉석으로 생성해야함
  - seq2seq model 방법 채택
  - 기존 vanilla seq2seq model은 extractive method 보다 성능이 낮았음 (Reddy et al. [2019])
  - 첫번째 segment에는 대화 이력을 concat해서 넣음(the input question and the passage)
  - 두번째 segment에서는 답변을 출력
  - experiments
    - CoQA 데이터셋에 대해서 fine-tuning
      - epoch: 10
      - batch size: 32
      - mask prob: 0.5
      - max len: 512
      - label smoothing: 0.1
    - decoding에 beam search 적용 (with 3 beam size)

	EM	F1
RMR+ELMo [20]	71.4	73.7
$\text{BERT}_{\text{LARGE}}$	78.9	81.8
UNILM	**80.5**	**83.4**

Table 5: Extractive QA results on the SQuAD development set.

	F1
DrQA+ELMo [35]	67.2
$\text{BERT}_{\text{LARGE}}$	82.7
UNILM	**84.9**

Table 6: Extractive QA results on the CoQA development set.

	F1
Seq2Seq [35]	27.5
PGNet [35]	45.4
UNILM	**82.5**

Table 7: Generative QA results on the CoQA development set.

3.3 Question Generation

- passage와 answer가 주어졌을 때, question을 생성하는 것
- seq2seq 문제로 보고 풀겠음
  - 1st seg: input passage + answer
  - 2nd seg: generated question
- SQuAD 1.1 dataset을 평가셋으로 사용함
- 선행 연구에서와 같이 original training set을 training과 test sets으로 쪼개서 사용하기로하고 original dev set은 그대로둠
- hyper params:
  - epoch: 10
  - batch size: 32
  - mask prob: 0.7
  - lr: 2e-5
  - label smoothing: 0.1

	BLEU-4	MTR	RG-L
CorefNQG [11]	15.16	19.12	-
SemQG [50]	18.37	22.65	46.68
UNILM	**22.12**	**25.06**	**51.07**
MP-GSN [51]	16.38	20.25	44.48
SemQG [50]	20.76	24.20	48.91
UNILM	**23.75**	**25.61**	**52.04**

Table 8: Question generation results on SQuAD. MTR is short for METEOR, and RG for ROUGE. Results in the groups use different data splits.

3.3 Question Generation

Generated Questions Improve QA

- Question generation model로 질문을 만들어서(data augmentation) 다시 학습시키면 기존의 question answering model의 성능이 올라감

	EM	F1
UNILM QA Model (Section 3.2)	80.5	83.4
+ UNILM Generated Questions	**84.7**	**87.6**

Table 9: Question generation based on UNILM improves question answering results on the SQuAD development set.

3.4 Response Generation

- document-grounded dialog response generation task로 UNILM을 평가해봄
- multi-turn conversation history와 a web document as the knowledge source가 주어진 상태에서 시스템은 대화에도 알맞고, web document contents도 반영하는 답변을 해야함
- UNILM을 seq2seq model로 사용함
  - 1st seg: web document + conversation history
  - 2nd seg: response
- dataset: DSTC7
- hyper params:
  - epoch: 20
  - batch size: 64
  - masking prob: 0.5
  - max len: 512
- decoding에 beam search 적용 (with 10 beam size)

	NIST-4	BLEU-4	METEOR	Entropy-4	Div-1	Div-2	Avg len
Best System in DSTC7 Shared Task	2.523	1.83	8.07	9.030	0.109	0.325	15.133
UNILM	**2.669**	**4.39**	**8.27**	**9.195**	**0.120**	**0.391**	14.807
Human Performance	2.650	3.13	8.31	10.445	0.167	0.670	18.76

Table 10: Response generation results. Div-1 and Div-2 indicate diversity of unigrams and bigrams, respectively.

3.5 GLUE Benchmark

- (~~버트보다 좋은 성능 가진 모델이 많이 나왔는데 버트랑만 비교하는건 좀 아쉽다~~)

Model	CoLA MCC	SST-2 Acc	MRPC F1	STS-B S Corr	QQP F1	MNLI-m/mm Acc	QNLI Acc	RTE Acc	WNLI Acc	AX Acc	**Score**
GPT	45.4	91.3	82.3	80.0	70.3	82.1/81.4	87.4	56.0	53.4	29.8	72.8
$\text{BERT}_{\text{LARGE}}$	60.5	**94.9**	89.3	86.5	**72.1**	86.7/**85.9**	**92.7**	70.1	65.1	39.6	80.5
UNILM	**61.1**	94.5	**90.0**	**87.7**	71.7	**87.0/85.9**	**92.7**	**70.9**	65.1	38.4	**80.8**

Table 11: GLUE test set results scored using the GLUE evaluation server.

4. Conclusion and Future Work

- several LM objectives를 shared parameters로 학습하는 unified pre-training model인 UNILM을 제안함
- NLU와 NLG 둘다 가능함
- BERT와 GLUE 벤치마크에서 비교할만했음
- 5가지 NLG dataset에서 SOTA를 달성함 (`CNN/DailyMail and Gigaword abstractive summarization, SQuAD question generation, CoQA generative question answering, and DSTC7 dialog response generation`)
- Future works:
  - training more epochs and larger models on web scale text corpora + ablation experiments
  - support cross-lingual tasks
  - multi-task fine-tuning on both NLU and NLG tasks (MT-DNN의 extension)

Github

- https://github.com/microsoft/unilm