Language Models are Unsupervised Multitask Learners

Alec Radford | Jeffrey Wu | Rewon Child | David Luan

Dario Amodei | Ilya Sutskever

Seungone Kim

Department of Computer Science

Yonsei University

louisdebroglie@yonsei.ac.kr

2021.07.06

연세대학교 YONSEI UNIVERSITY | SOFT COMPUTING LABORATORY

Outline

- 관련 연구
  - 논문 발표 시점 언어 모델 동향 분석
  - 기존 방법들의 한계점

- 제안하는 방법
  - Language modeling without supervision
  - Unsupervised multi-task learning
  - Byte-pair Encoding
  - Model Architecture

- 실험 결과
  - Visualization of how Attention, GPT works
  - hyperparameters
  - Zero-shot performance on various datasets

- 분석 및 요약
  - Generalization vs Memorization
  - Inspecting performance

- 결론

관련 연구

- 논문 발표 시점 : 2019.02

- Combination of pre-training and supervised fine-tuning
  - Best performing systems on language tasks

- History with a trend towards more flexible forms of transfer
  - Word Vectors (Mikolov et al., 2013)
  - Contextual representations of recurrent networks (Dai & Le, 2015) (Peters et al., 2018)
  - Transferring many self-attention blocks (Radford et al., 2018) (Devlin et al., 2018)

관련 연구 (기존 방법들의 한계점)

- Move towards more general systems which can perform many tasks
  - Multi-task Learning
  - Current systems are better characterized as narrow experts rather than generalists
  - Sensitive to data distribution (Recht et al., 2018)
  - Sensitive to task specification (Kirkpatrick et al, 2017)
  - Prevalence of single task training on single domain datasets is major reason

- Multitask training in NLP is still nascent
  - Adapting previously acquired knowledge about language (Yogatama et al., 2019)
  - Benchmark for measuring performance across ten tasks (McCann et al., 2018)
  - (dataset, objective)

- Current methods still require supervised training in order to perform a task
  - Connect two lines of work and continue trend of more general methods of transfer
  - Zero-shot setting : without any parameter or architecture modification

제안하는 방법

- Language modeling
  - Joint probabilities over symbols as the product of conditional probabilities (Bengio et al., 2003)

$$p(x) = \prod_{i=1}^{n} p(s_n | s_1, ..., s_{n-1})$$

- Expressing in a probabilistic framework
  - Learning to perform a single task : p(output | input)
  - Multitask and meta-learning setting : p(output | input, task)
  - e.g. (translate to french, english text, french text), (answer question, document, question, answer)

- Language modeling can learn target tasks without explicit supervision
  - Supervised objective is the same as the unsupervised objective
  - Supervised objective is only evaluated on a subset of the sequence
  - Global minimum of unsupervised objective is also the global minimum of supervised objective
  - Is is possible, in practice, to optimize the unsupervised objective to converge

제안하는 방법

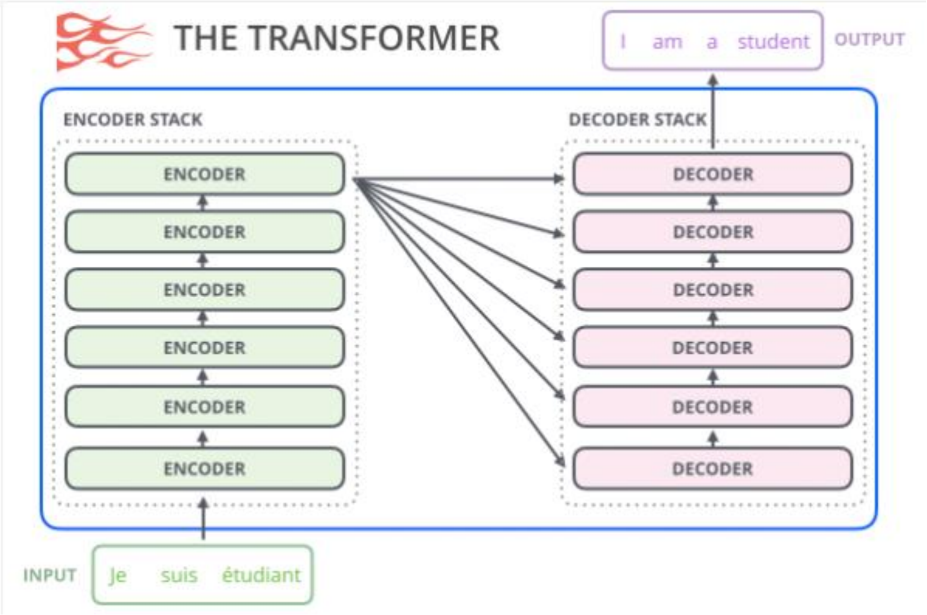
- Unsupervised Multitask Learning
  - A LM with sufficient capacity will begin to learn to infer and perform the tasks
  - Better prediction of language sequences, regardless of their procurement

- Byte Pair Encoding
  - Practical middle ground between character and word level language modeling
  - Despite its name, reference BPE often operate on Unicode, not byte sequences
  - Modeling on all Unicode strings result base vocabulary over 130,000
  - In contrast, byte-level version vocabulary only requires size 256
  - Allows to combine empirical benefits of word-level LMs with generality of byte-level approaches

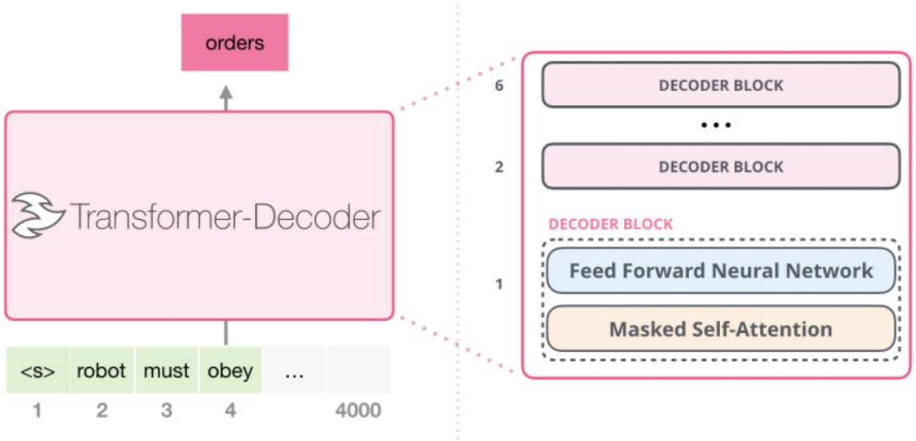
- Models
  - Follows detail of OpenAI GPT model (Radford et al., 2018)
  - Layer Normalization (Ba et al., 2016)
  - Initialization which accounts for the accumulation on residual path with model depth
  - Vocabulary expanded to 50,257
  - Context size from 512 to 1024 tokens and batch size of 512

Visualizing Model Architecture

- 출처 : https://jalammar.github.io/illustrated-gpt2/



Transformer (Vaswani et al., 2017)



GPT (Radford et al., 2018)

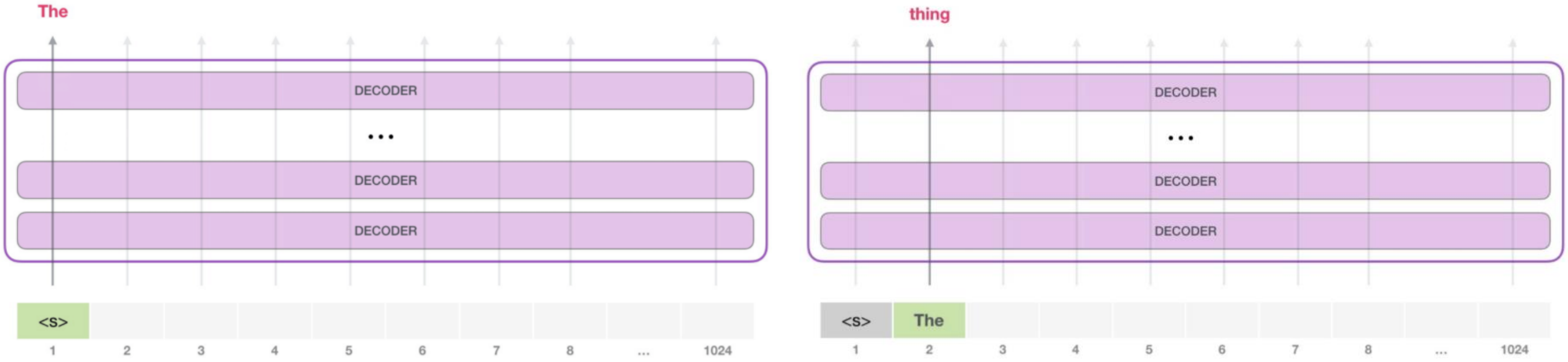
Visualizing how GPT works (Auto Regressive)

- 출처 : https://jalammar.github.io/illustrated-gpt2/
- 출처 : https://blog.pingpong.us/xlnet-review/

$$input\ sequence : x = (x_1, x_2, \ldots, x_T)$$

$$forward\ likelihood : p(x) = \Pi_{t=1}^{T} p(x_t \mid x_{<t})$$

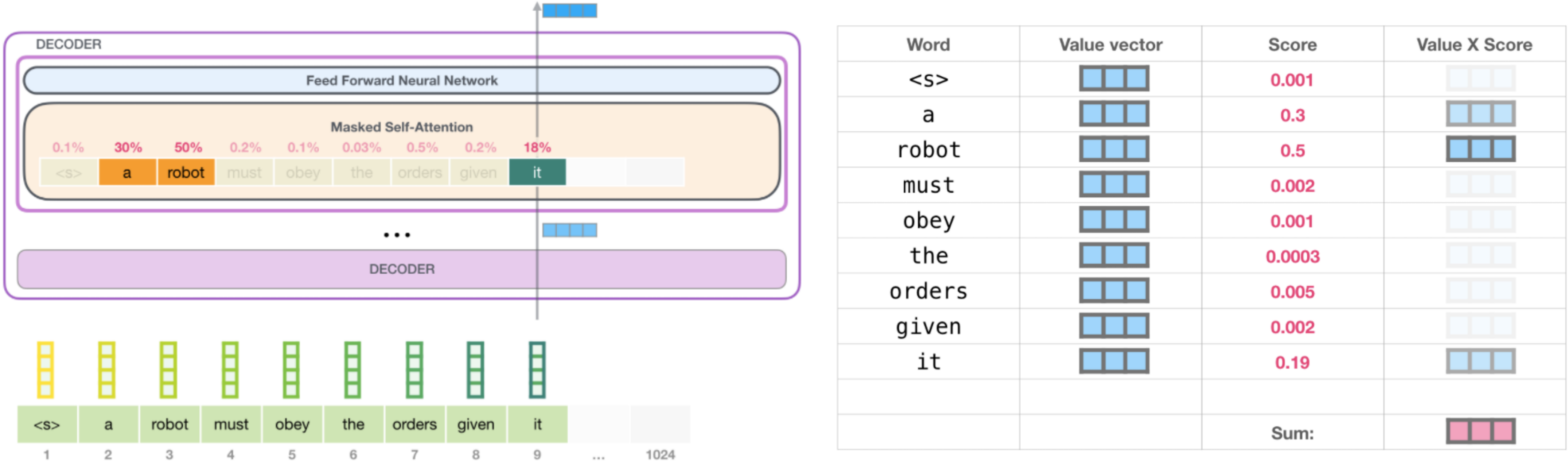
$$training\ objective(forward) : \max_{\theta}\ \log p_{\theta}(x) = \max_{\theta}\ \Sigma_{t=1}^{T} \log p(x_t \mid x_{<t})$$



Process of Decoding (requires O(N))

Visualization of Self-Attention

- 출처 : https://jalammar.github.io/illustrated-gpt2/

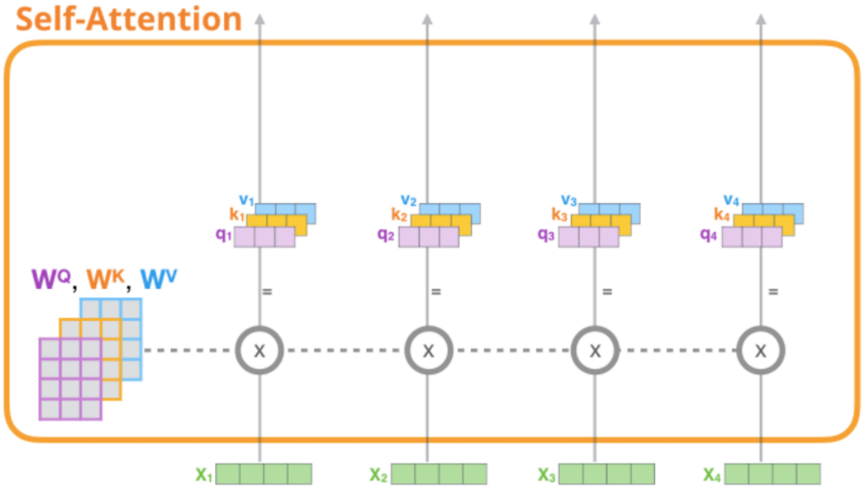


Word	Value vector	Score	Value X Score
<s>	🟦🟦🟦	0.001	⬜⬜⬜
a	🟦🟦🟦	0.3	🟦🟦🟦
robot	🟦🟦🟦	0.5	🟦🟦🟦
must	🟦🟦🟦	0.002	⬜⬜⬜
obey	🟦🟦🟦	0.001	⬜⬜⬜
the	🟦🟦🟦	0.0003	⬜⬜⬜
orders	🟦🟦🟦	0.005	⬜⬜⬜
given	🟦🟦🟦	0.002	⬜⬜⬜
it	🟦🟦🟦	0.19	🟦🟦🟦
		Sum:	🟥🟥🟥

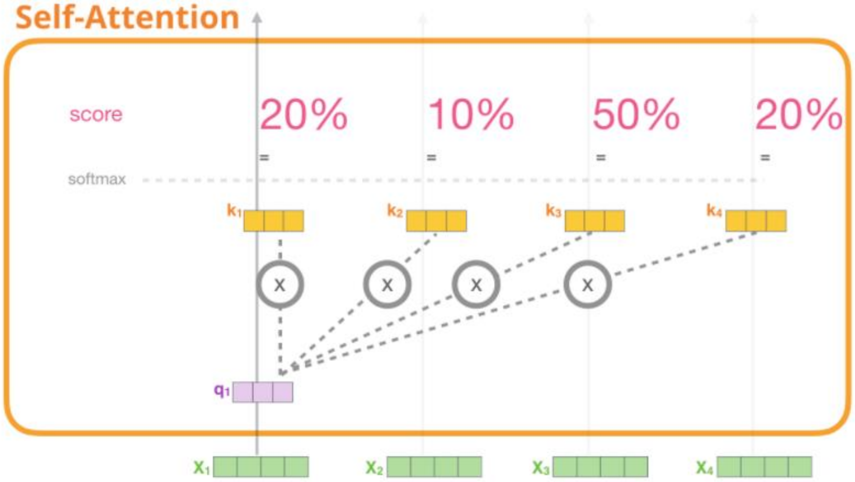
Process of achieving Decoder output

Visualization of Self-Attention

- 출처 : https://jalammar.github.io/illustrated-gpt2/



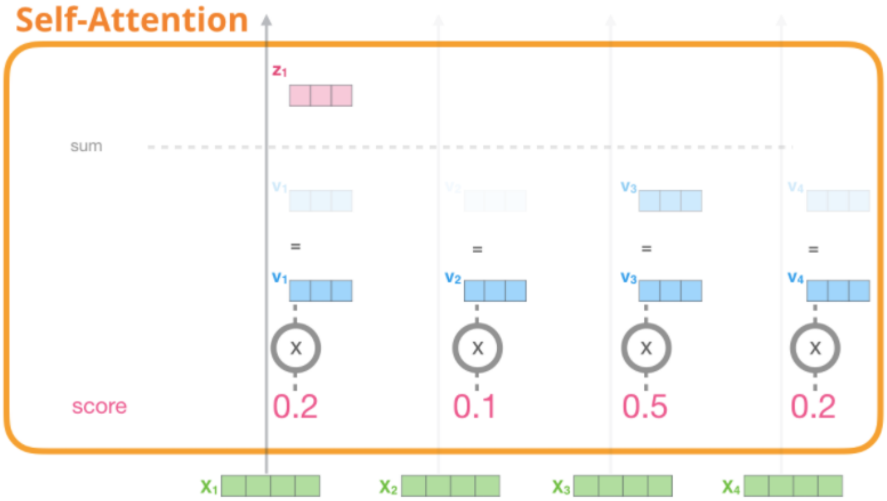
Calculating Q,K,V matrices with input



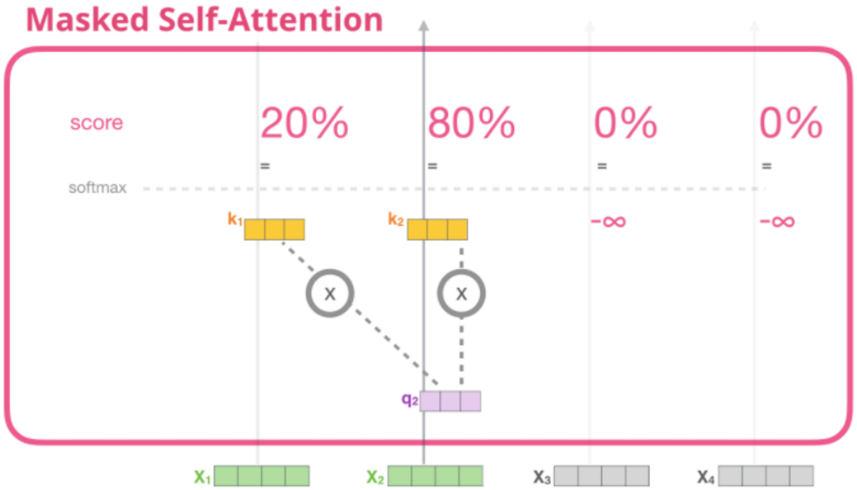
Calculating score with Q,K

Visualization of Masked Self-Attention

- Calculating Q,K,V matrices with input

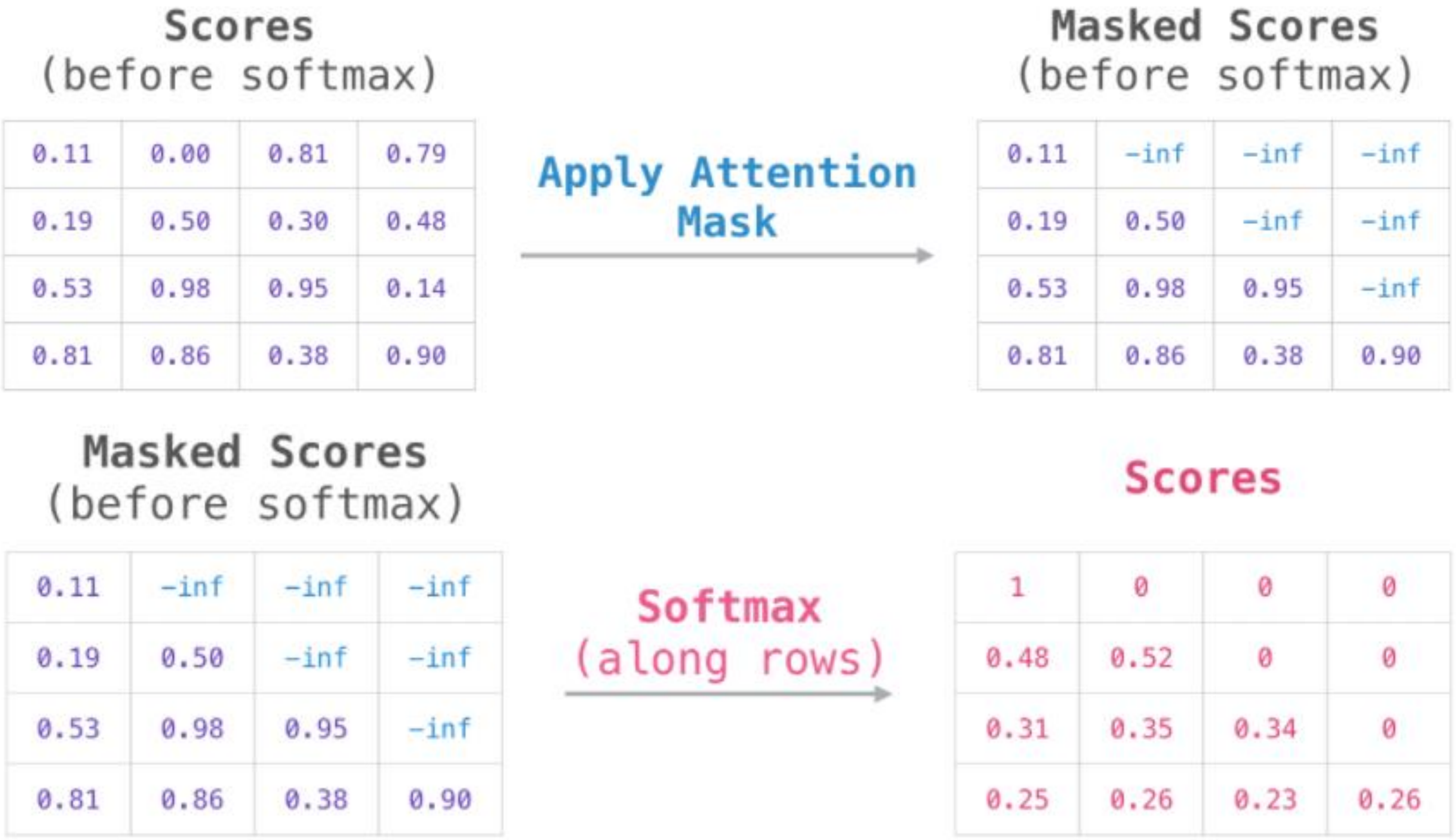


Calculating output with score, V



Calculating output with masked score, V

Visualization of applying mask



Scores
(before softmax)

0.11	0.00	0.81	0.79
0.19	0.50	0.30	0.48
0.53	0.98	0.95	0.14
0.81	0.86	0.38	0.90

Apply Attention Mask

Masked Scores
(before softmax)

0.11	-inf	-inf	-inf
0.19	0.50	-inf	-inf
0.53	0.98	0.95	-inf
0.81	0.86	0.38	0.90

Masked Scores
(before softmax)

0.11	-inf	-inf	-inf
0.19	0.50	-inf	-inf
0.53	0.98	0.95	-inf
0.81	0.86	0.38	0.90

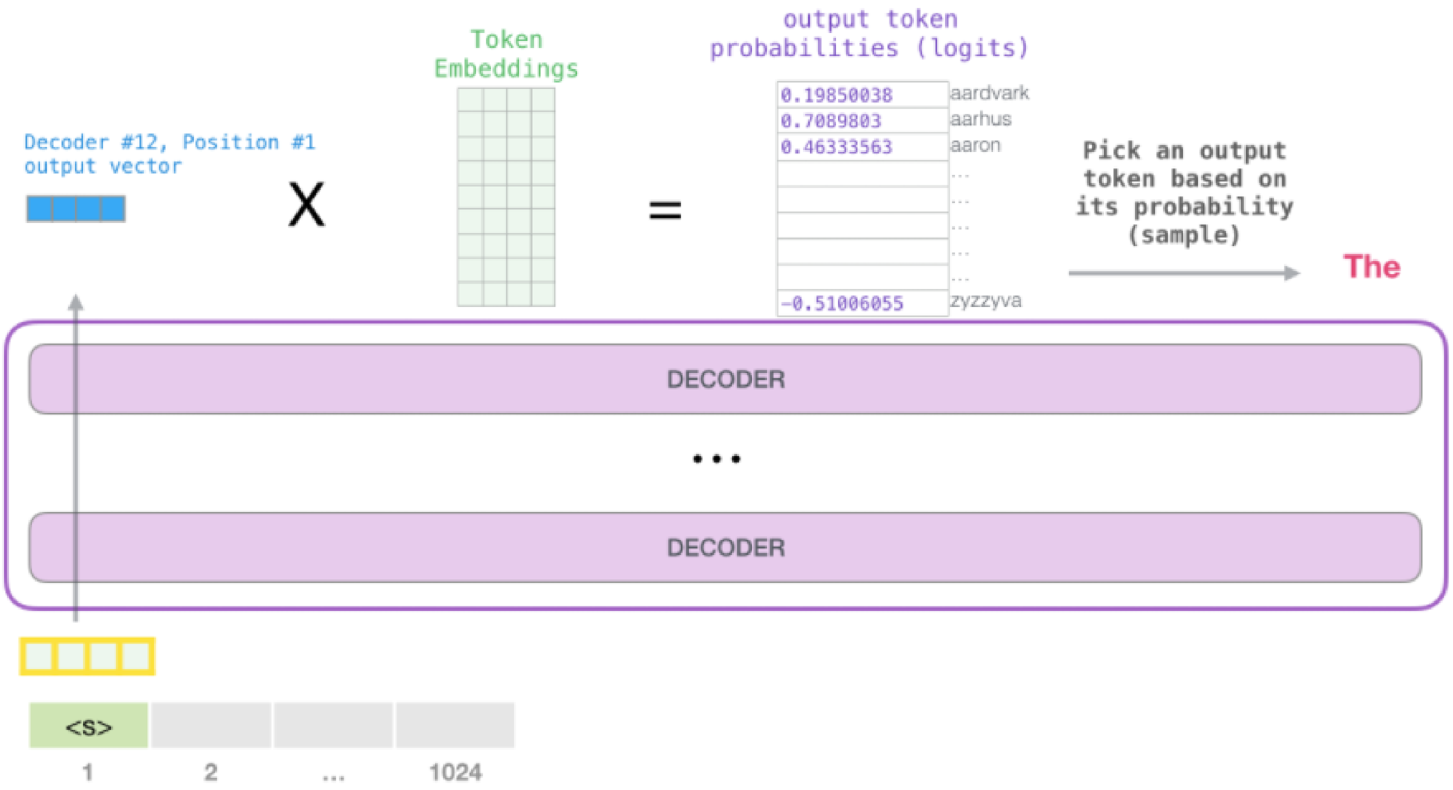
Softmax
(along rows)

Scores

1	0	0	0
0.48	0.52	0	0
0.31	0.35	0.34	0
0.25	0.26	0.23	0.26

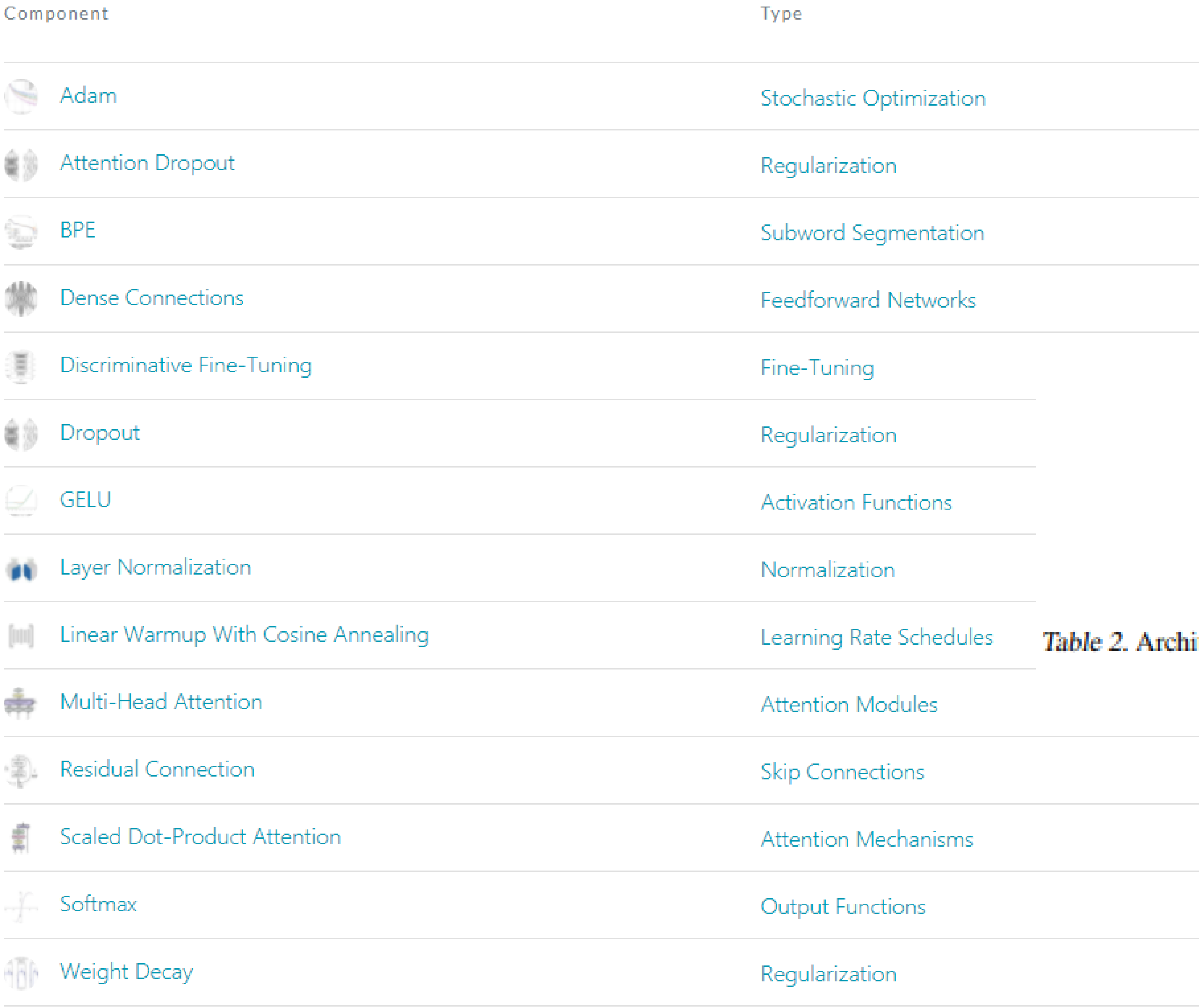
Comparison of vanilla process and applying attention mask

Visualization of how to generate text

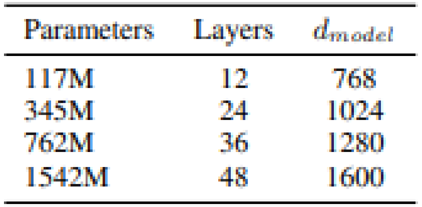


Decoding model output into output token

실험 결과



Component	Type
Adam	Stochastic Optimization
Attention Dropout	Regularization
BPE	Subword Segmentation
Dense Connections	Feedforward Networks
Discriminative Fine-Tuning	Fine-Tuning
Dropout	Regularization
GELU	Activation Functions
Layer Normalization	Normalization
Linear Warmup With Cosine Annealing	Learning Rate Schedules
Multi-Head Attention	Attention Modules
Residual Connection	Skip Connections
Scaled Dot-Product Attention	Attention Mechanisms
Softmax	Output Functions
Weight Decay	Regularization



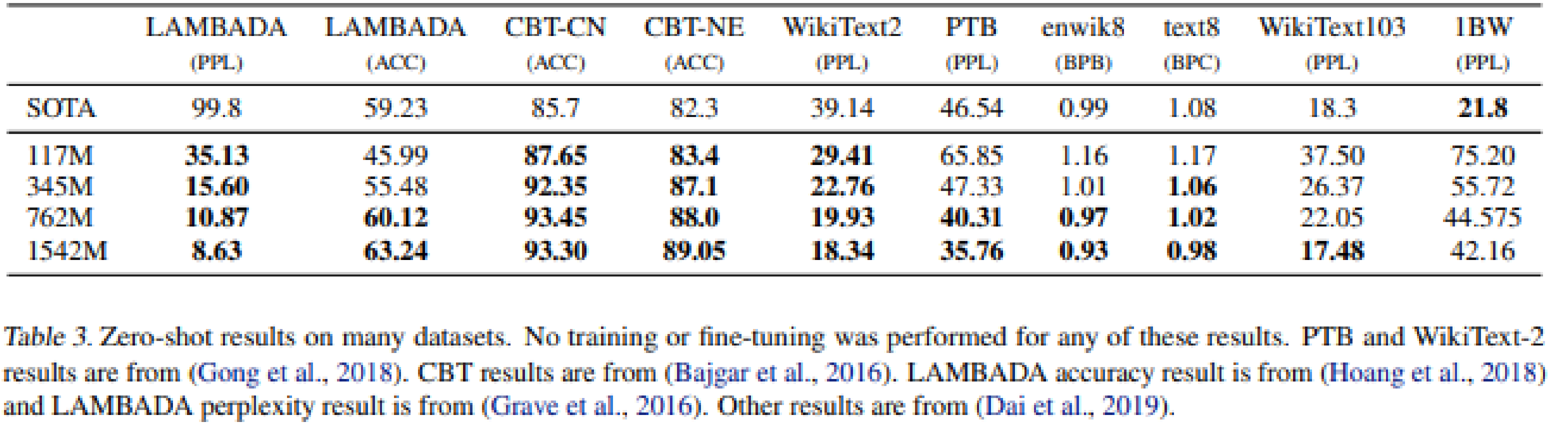
Parameters	Layers	d_{model}
117M	12	768
345M	24	1024
762M	36	1280
1542M	48	1600

Table 2. Architecture hyperparameters for the 4 model sizes.

출처 : https://paperswithcode.com/method/gpt-2

실험 결과

- Result of zero-shot performance with only pretraining on WebText LM

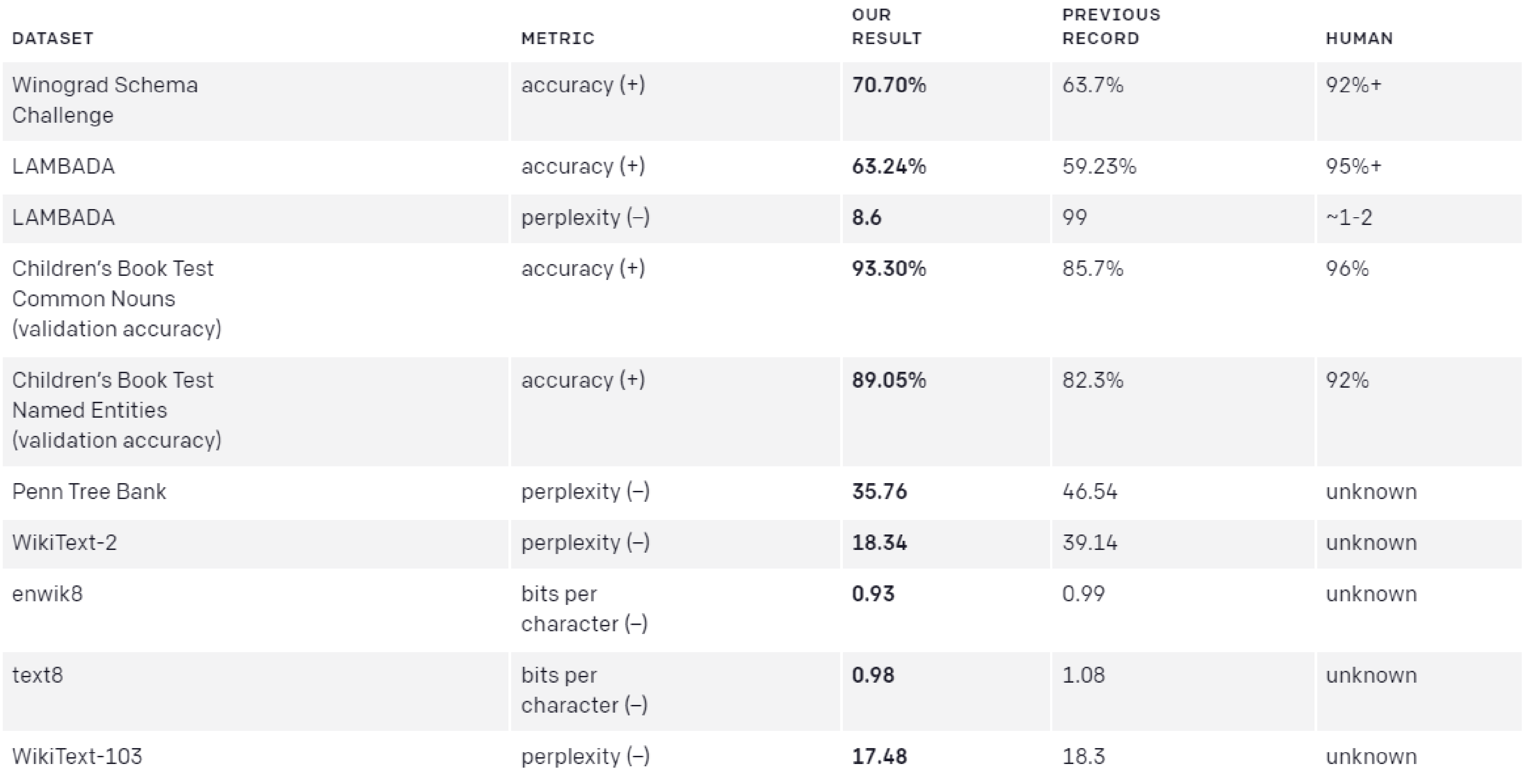


	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)	PTB (PPL)	enwik8 (BPB)	text8 (BPC)	WikiText103 (PPL)	1BW (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14	46.54	0.99	1.08	18.3	**21.8**
117M	**35.13**	45.99	**87.65**	**83.4**	**29.41**	65.85	1.16	1.17	37.50	75.20
345M	**15.60**	55.48	**92.35**	**87.1**	**22.76**	47.33	1.01	**1.06**	26.37	55.72
762M	**10.87**	**60.12**	**93.45**	**88.0**	**19.93**	**40.31**	**0.97**	**1.02**	22.05	44.575
1542M	**8.63**	**63.24**	**93.30**	**89.05**	**18.34**	**35.76**	**0.93**	**0.98**	**17.48**	42.16

Table 3. Zero-shot results on many datasets. No training or fine-tuning was performed for any of these results. PTB and WikiText-2 results are from (Gong et al., 2018). CBT results are from (Bajgar et al., 2016). LAMBADA accuracy result is from (Hoang et al., 2018) and LAMBADA perplexity result is from (Grave et al., 2016). Other results are from (Dai et al., 2019).

실험 결과

- Result of zero-shot performance with only pretraining on WebText LM
  - Achieves SOTA on a variety of datasets in a zero-shot setting
  - On other language tasks like QA, Summarization, Translation still fall short for SOTA

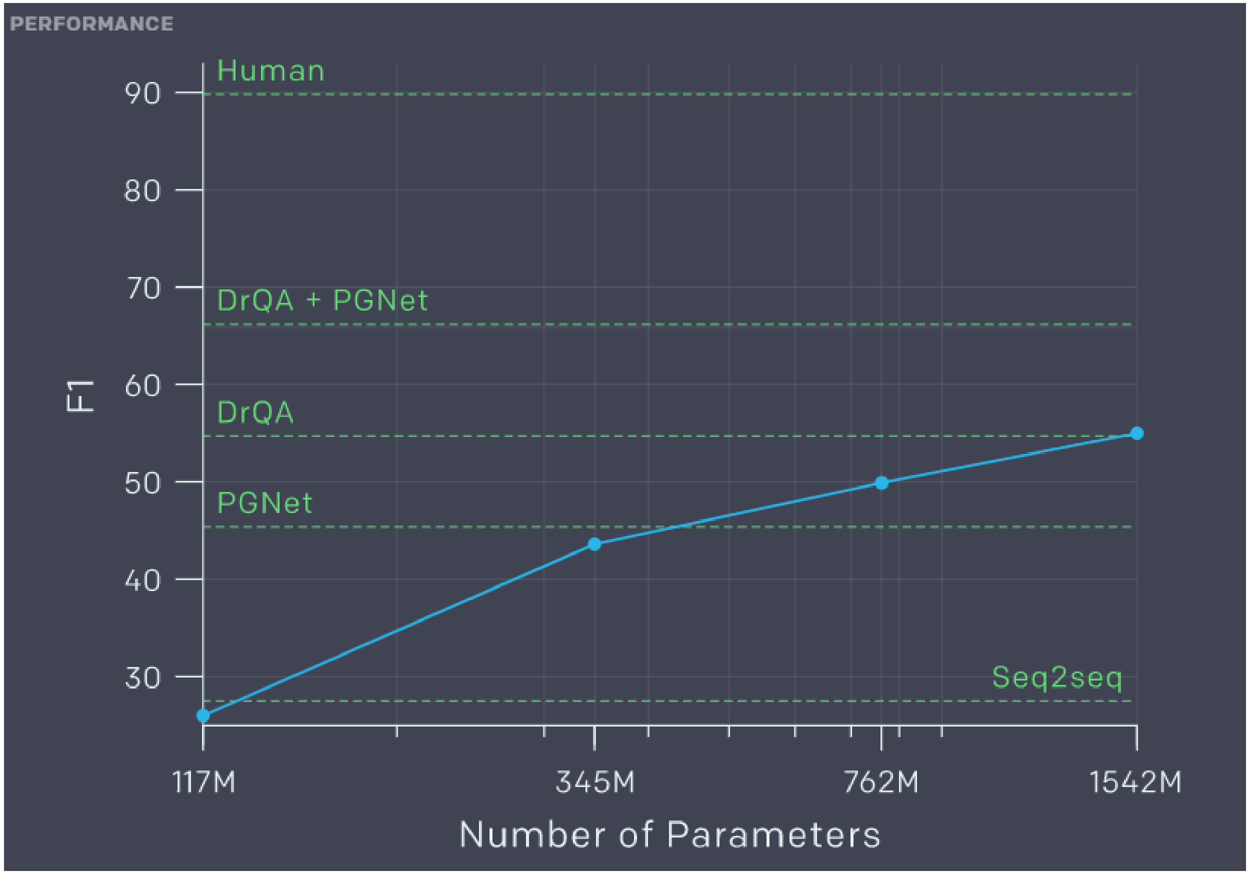


DATASET	METRIC	OUR RESULT	PREVIOUS RECORD	HUMAN
Winograd Schema Challenge	accuracy (+)	**70.70%**	63.7%	92%+
LAMBADA	accuracy (+)	**63.24%**	59.23%	95%+
LAMBADA	perplexity (–)	**8.6**	99	~1-2
Children's Book Test Common Nouns (validation accuracy)	accuracy (+)	**93.30%**	85.7%	96%
Children's Book Test Named Entities (validation accuracy)	accuracy (+)	**89.05%**	82.3%	92%
Penn Tree Bank	perplexity (–)	**35.76**	46.54	unknown
WikiText-2	perplexity (–)	**18.34**	39.14	unknown
enwik8	bits per character (–)	**0.93**	0.99	unknown
text8	bits per character (–)	**0.98**	1.08	unknown
WikiText-103	perplexity (–)	**17.48**	18.3	unknown

GPT-2 achieves state-of-the-art on Winograd Schema, LAMBADA, and other language modeling tasks.

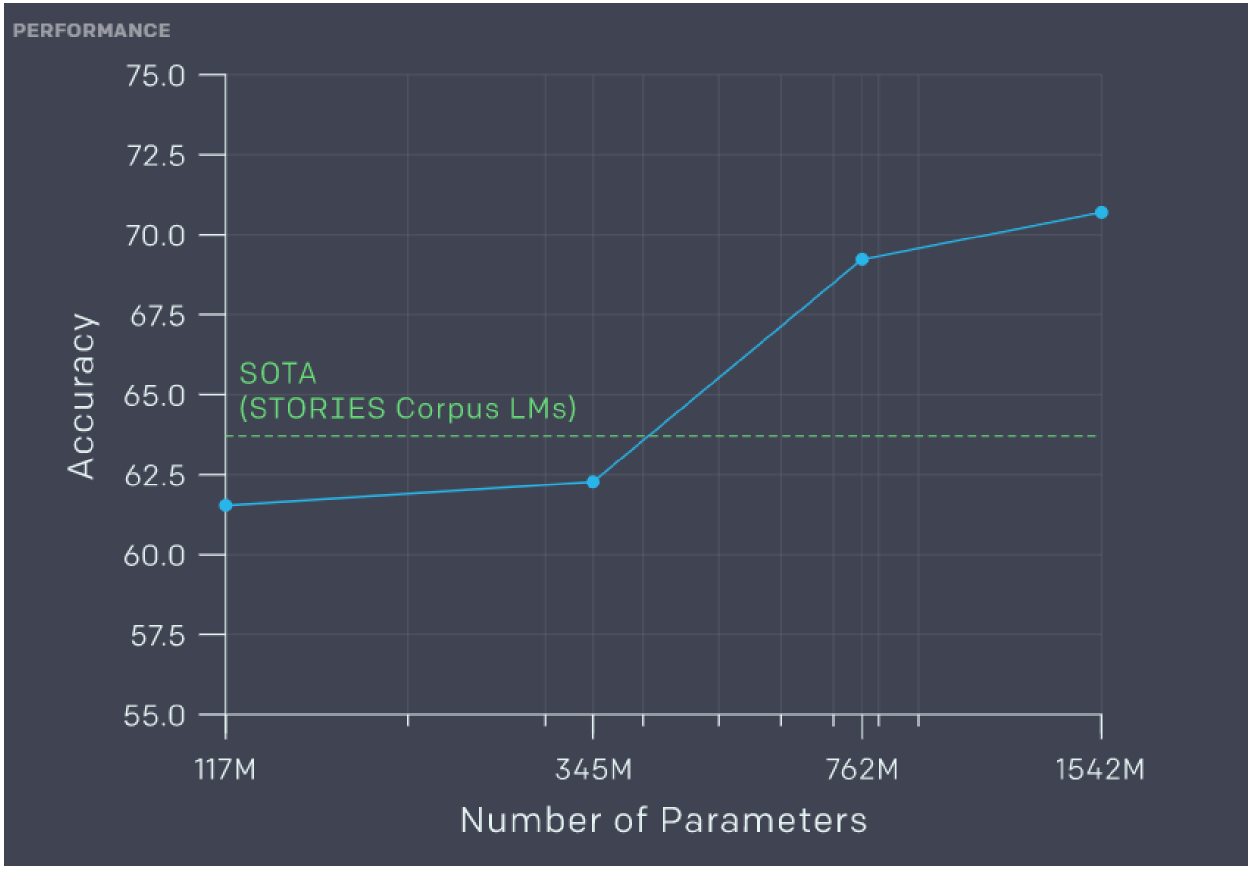
실험 결과

- Result of zero-shot performance on CoQA(Reading Comprehension)
  - Answering questions about given passages



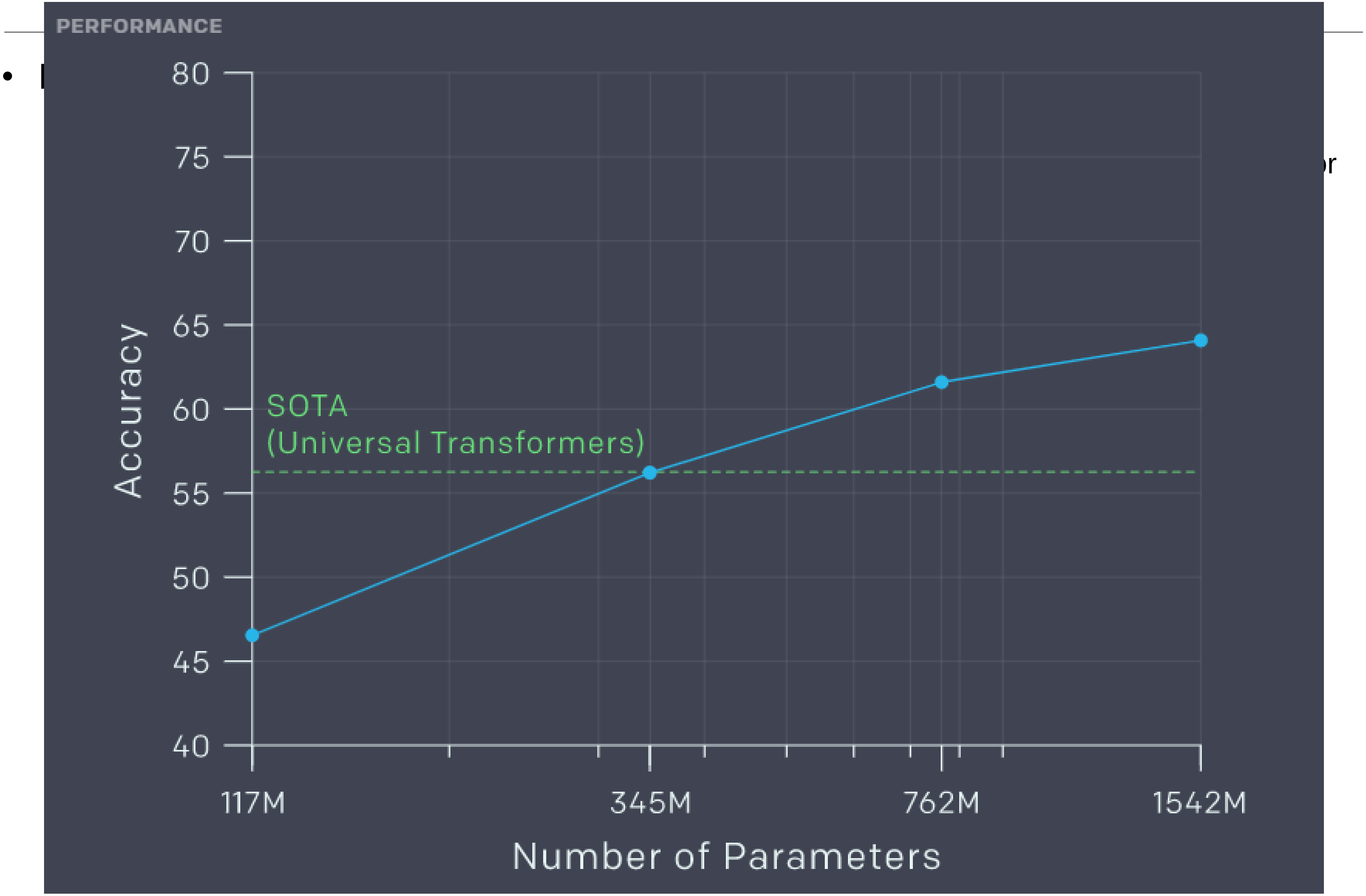
실험 결과

- Result of zero-shot performance on Winograd Schema(Commonsense reasoning)
    - Resolution of an ambiguous pronoun such as ‘it’



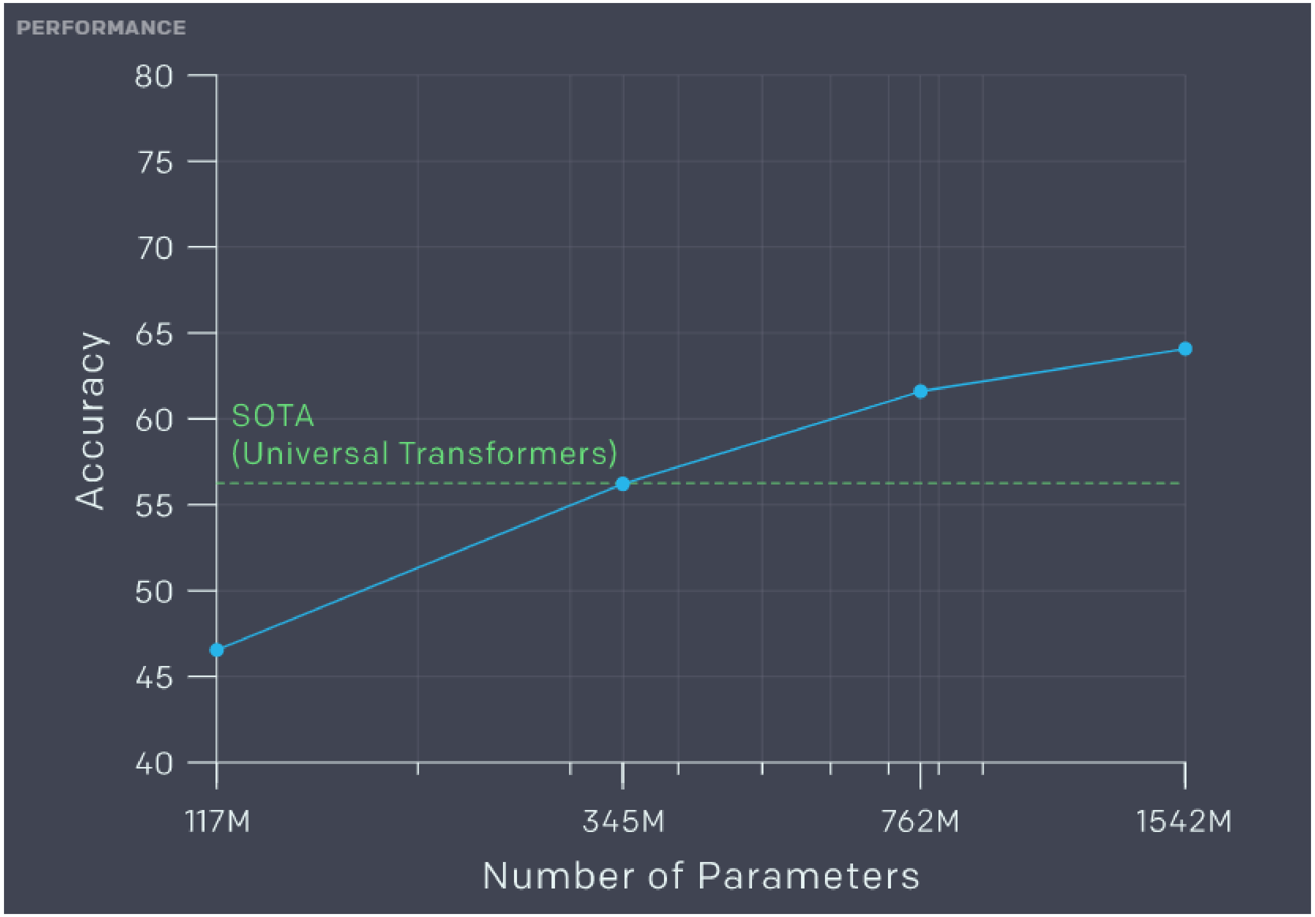
실험 결과

- I


PERFORMANCE
80
75
70
65
60
55
50
45
40
Accuracy
SOTA
(Universal Transformers)
117M
345M
762M
1542M
Number of Parameters

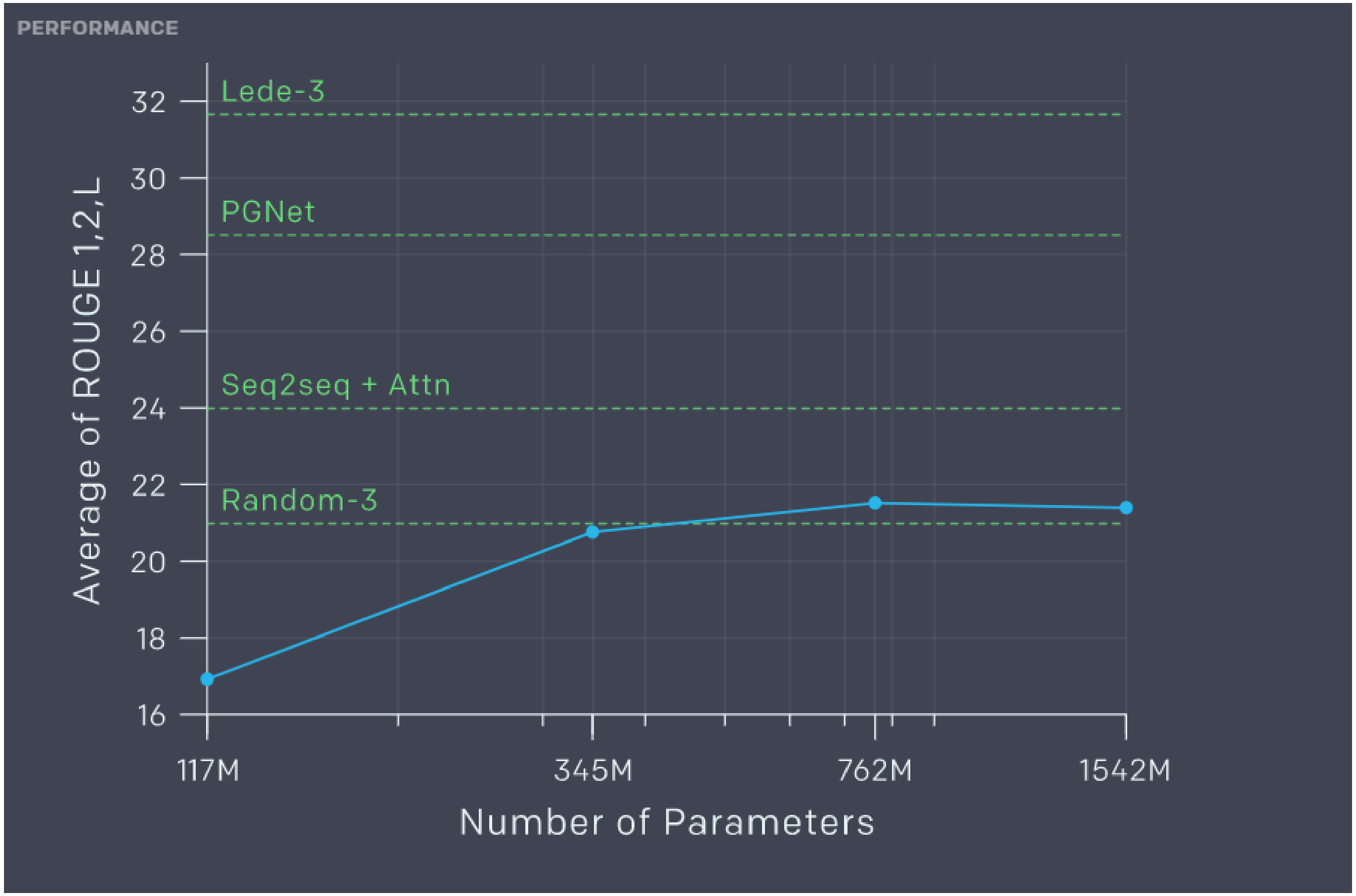
실험 결과

- Result of zero-shot performance on LAMBDA(Language Modeling)
  - Predicting the last word of a passage



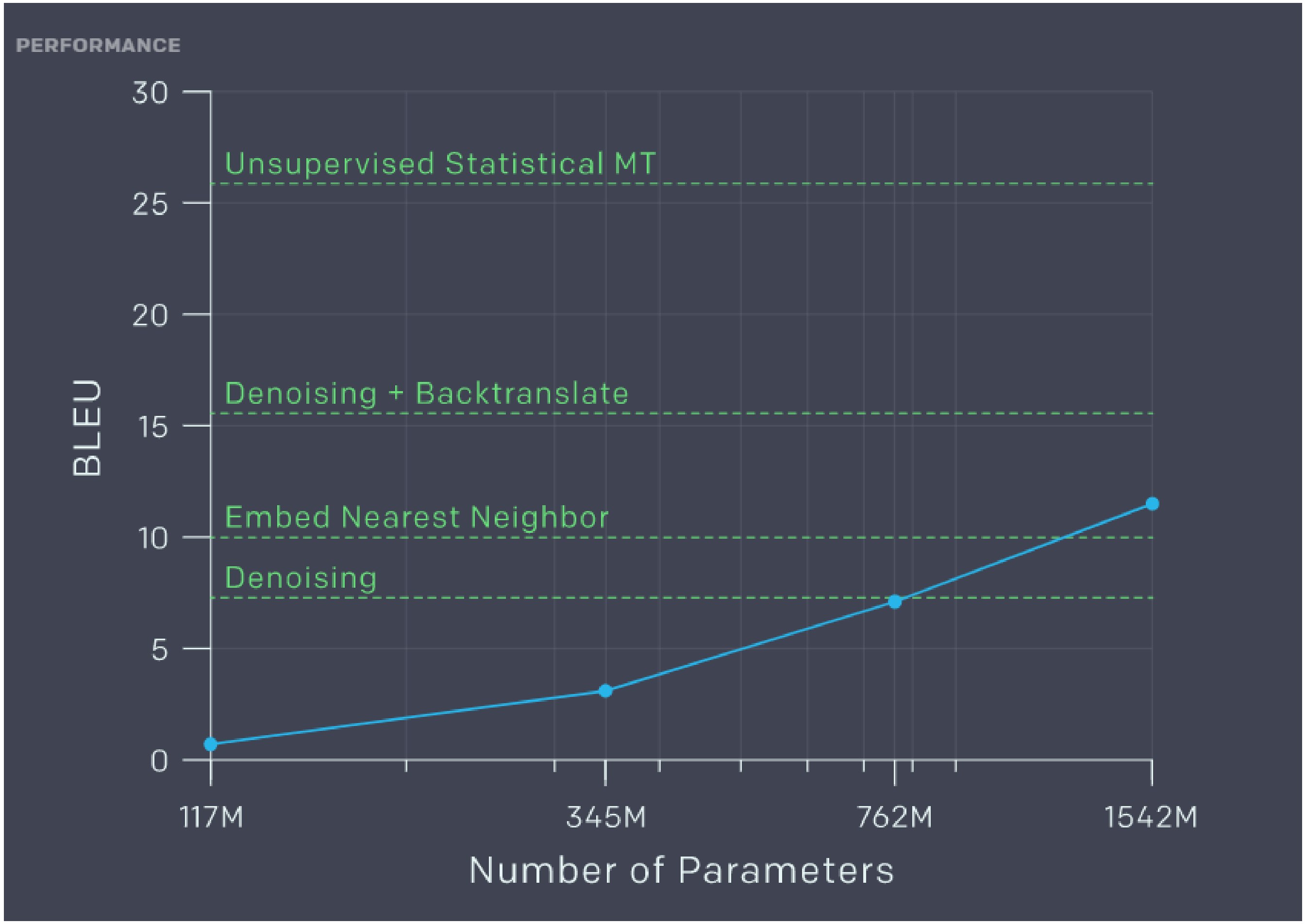
실험 결과

- Result of zero-shot performance on CNN/DM(Summarization)
  - Abstractive summarization given a reference text



실험 결과

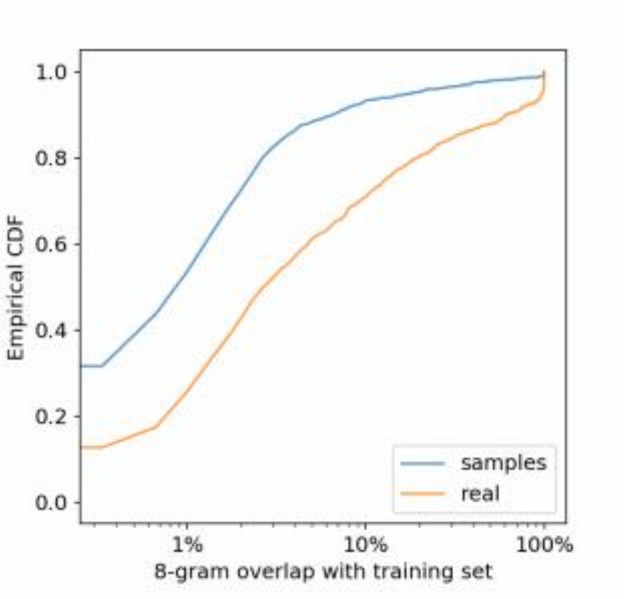
- Result of zero-shot performance on WMT14 Fr-En(Translation)
  - Translate French sentences to English



분석 및 요약

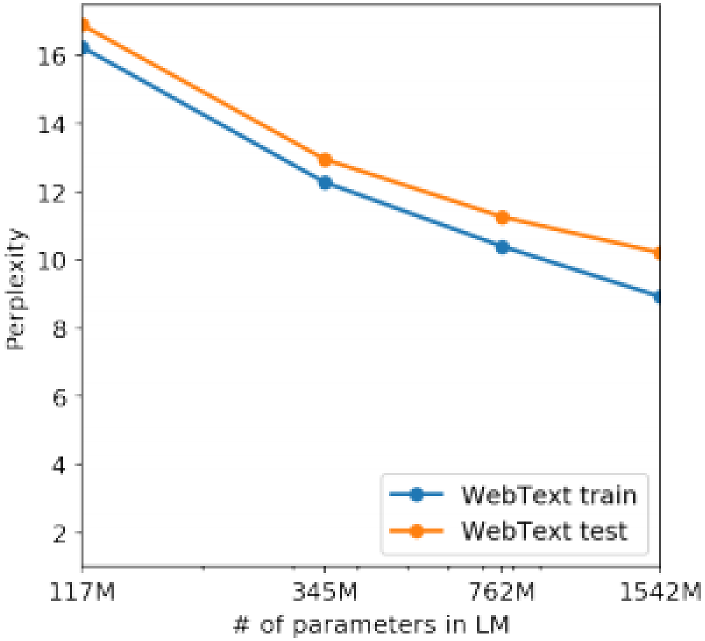
- Generalization vs Memorization
  - It is important to analyze how much test data also shows up in the training data
  - Importance of de-duplication as verification step

- Bloom filters
  - 8-grams of WebText training set token
  - Calculate, given a dataset, the percentage of 8-grams from that dataset also found in WebText
  - Common datasets' test sets have 1~6% overlap with WebText, average of 3.2%
  - Many datasets have larger overlaps with their own training splits, with average of 5.9% overlap



분석 및 요약

- Inspecting performance on their own held-out set
  - Perplexity is a measurement of how well a probability model predicts a sample
  - GPT-2 might still be underfitting on WebText in many ways



$$H(\tilde{p}, q) = -\sum_{x} \tilde{p}(x) \log_2 q(x)$$

결론

- Unsupervised task learning is an additional promising area of research
  - Pre-training techniques begins to learn to perform tasks directly without the need of fine-tuning

- GPT-2 is competitive with supervised baselines in a zero-shot setting
  - However, on tasks like summarization, while qualitatively performing task, performance is rudimentary according to quantitative metrics

- LMs only begin to outperform trivial baselines with sufficient capacity