Building Blocks of Modern LLMs 2: Pretraining Tasks

Chenyan Xiong

11-667

08/31/2023

Pretraining Tasks

Pretraining and Language Modeling

Pre-training: An <u>unsupervised</u> learning phrase before traditional <u>supervised</u> learning

- Original goal: provide better initialization points for supervised training

Language modeling: Predict a part of a given language piece (target) using the rest (context)

- A classic task in NLP et al. to model human usage of natural language

Pretraining and Language Modeling

Why language modeling as pretraining task?

- Infinite data, way more than current computing system can consume
  - Beyond trillions of web pages processed
  - Much more discovered

Pretraining and Language Modeling

Why language modeling as pretraining task?

- Infinite data, way more than current computing system can consume
  - Beyond trillions of web pages processed
  - Much more discovered
- Language, a main carrier of human knowledge
  - We learn, communicate, and invent through language
  - Other modalities often centered around language
  - Not all tasks need language, but one would argue whether that is "human intelligence"

Pretraining and Language Modeling

Why language modeling as pretraining task?

- Infinite data, way more than current computing system can consume
  - Beyond trillions of web pages processed
  - Much more discovered
- Language, a main carrier of human knowledge
  - We learn, communicate, and invent through language
  - Other modalities often centered around language
  - Not all tasks need language, but one would argue whether that is “human intelligence”
- Many real-world applications are centered around language
  - Search, machine translation, question answering, writing assistance, etc.

Autoregressive Language Modeling

Classic language modeling: Given previous words, predict the next word

- Let $X = \{x_1, \dots x_t \dots, x_n\}$ a text sequence of n tokens, the standard language modeling objective is to maximize the likelihood:

$$L_{lm}(X) = \sum_t \log p(x_t | x_{t-k:t-1}; \Theta)$$

- Where:
    - x_t: t-th token, the prediction target
    - $x_{t-k:t-1}$: previous k tokens (context), k=context window size
    - Θ: language model parameters

Autoregressive: predicting the next word given previous words

- Following the nature of language, though can be done reversely too

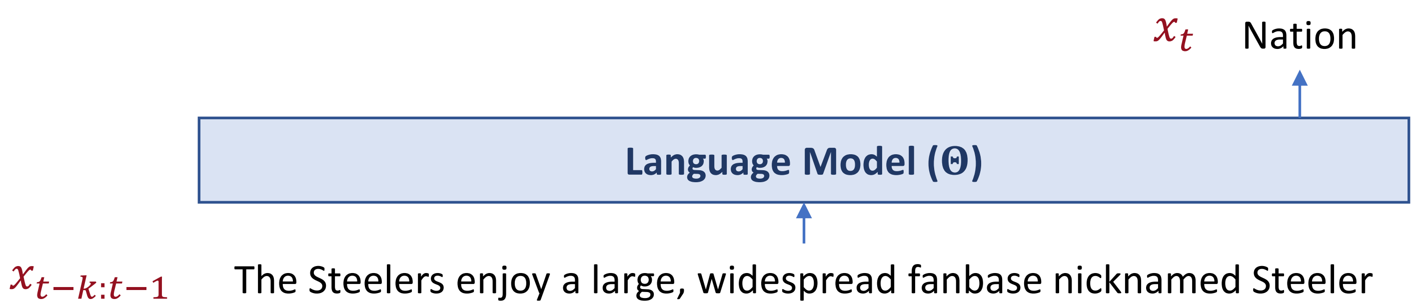
Autoregressive Language Modeling

Classic language modeling: Given previous words, predict the next word

- Let $X = \{x_1, \dots x_t \dots, x_n\}$ a text sequence of n tokens, the standard language modeling objective is to maximize the likelihood:

$$L_{lm}(X) = \sum_t \log p(x_t | x_{t-k:t-1}; \Theta)$$

- Where:
    - x_t: t-th token, the prediction target
    - $x_{t-k:t-1}$: previous k tokens (context), k=context window size
    - Θ: language model parameters



Autoregressive Language Modeling

The language model can be implemented in many ways

- Discrete n-gram frequency based:

$$p(x_t | x_{t-k:t-1}) = \frac{\text{count}(x_{t-k:t-1}, x_t)}{\text{count}(x_{t-k:t-1})}$$

- Continuous neural network models:

$$p(x_t | x_{t-k:t-1}; \Theta) = f(x_t | x_{t-k:t-1}; \Theta)$$

  - $f(;\Theta)$: a neural network, e.g., feedforward network, CNN, RNN, or

Autoregressive Language Modeling

The language model can be implemented in many ways

- Discrete n-gram frequency based:

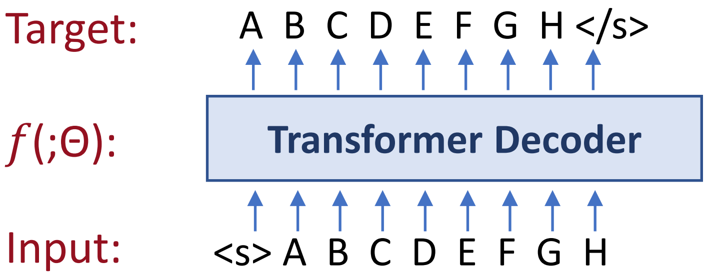
$$p(x_t|x_{t-k:t-1}) = \frac{\text{count}(x_{t-k:t-1}, x_t)}{\text{count}(x_{t-k:t-1})}$$

- Continuous neural network models:

$$p(x_t|x_{t-k:t-1}; \Theta) = f(x_t|x_{t-k:t-1}; \Theta)$$

  - $f(;\Theta)$: a neural network, e.g., feedforward network, CNN, RNN, or

- Transformer Decoder:



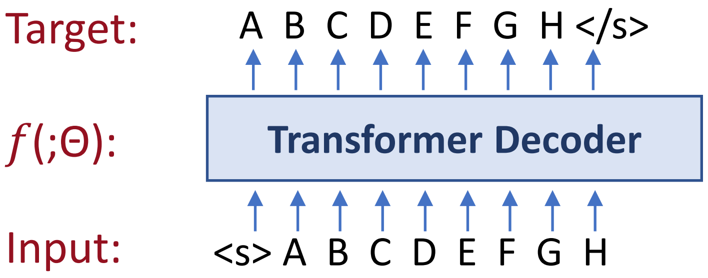
Autoregressive Language Modeling

Advantages of autoregressive language modeling:

- Intuitive, follows the nature flow of human language
- Aligns with many natural language generation style tasks
- Training signals at every token position in the sequence

Constraints:

- More for decoder style models, a.k.a. unidirectional networks→restriction of model flexibility



Auto-Encoder Language Modeling

Learn to reconstruct language from a learned hidden representation

- Given the text sequence $X = \{x_1, \ldots x_t \ldots, x_n\}$, the auto-encoder is to maximize the reconstruction likelihood:

$$L_{\text{AE}}(X) = \sum_t \log p(x_t | x_{t-k:t-1}; \Theta_{\text{dec}}, \boldsymbol{z}) f(\boldsymbol{z} | X, \Theta_{\text{enc}})$$

- Where:
  - Θ_{dec}: language decoder parameters
  - Θ_{enc}: language encoder parameters
  - $\boldsymbol{z}$: the hidden representation. Many viable formulations. In this class it is a neural embedding.

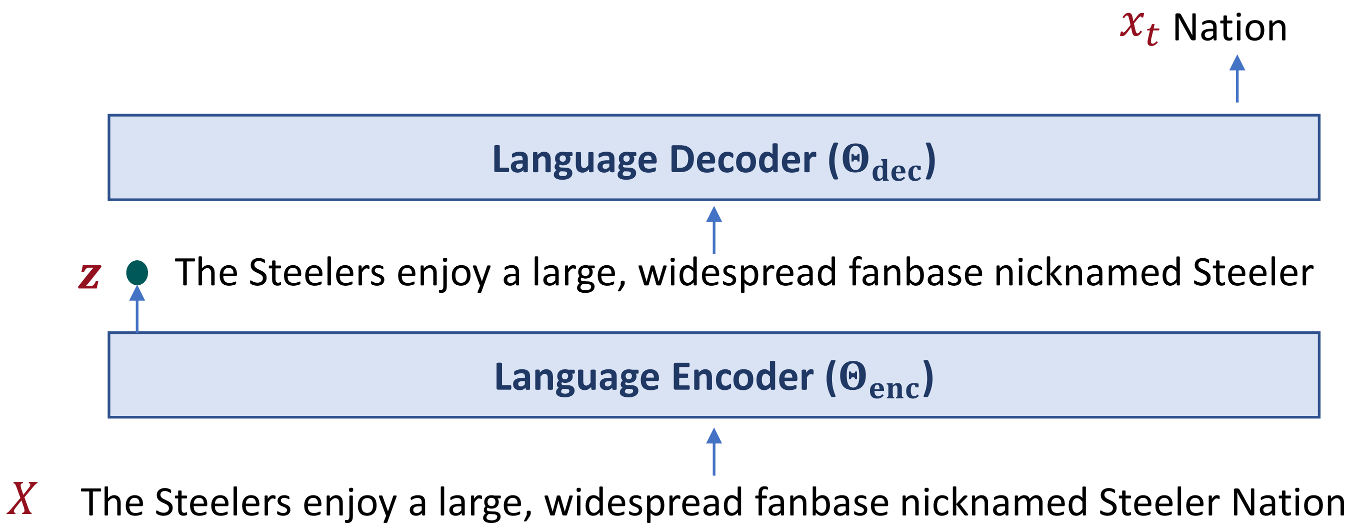
Auto-Encoder Language Modeling

Learn to reconstruct language from a learned hidden representation

- Given the text sequence $X = \{x_1, \ldots x_t \ldots, x_n\}$, the auto-encoder is to maximize the reconstruction likelihood:

$$L_{\mathrm{AE}}(X) = \sum_t \log p(x_t | x_{t-k:t-1}; \Theta_{\mathrm{dec}}, \boldsymbol{z}) f(\boldsymbol{z} | X, \Theta_{\mathrm{enc}})$$

- Where:
  - Θ_{dec}: language decoder parameters
  - Θ_{enc}: language encoder parameters
  - $\boldsymbol{z}$: the hidden representation. Many viable formulations. In this class it is a neural embedding.

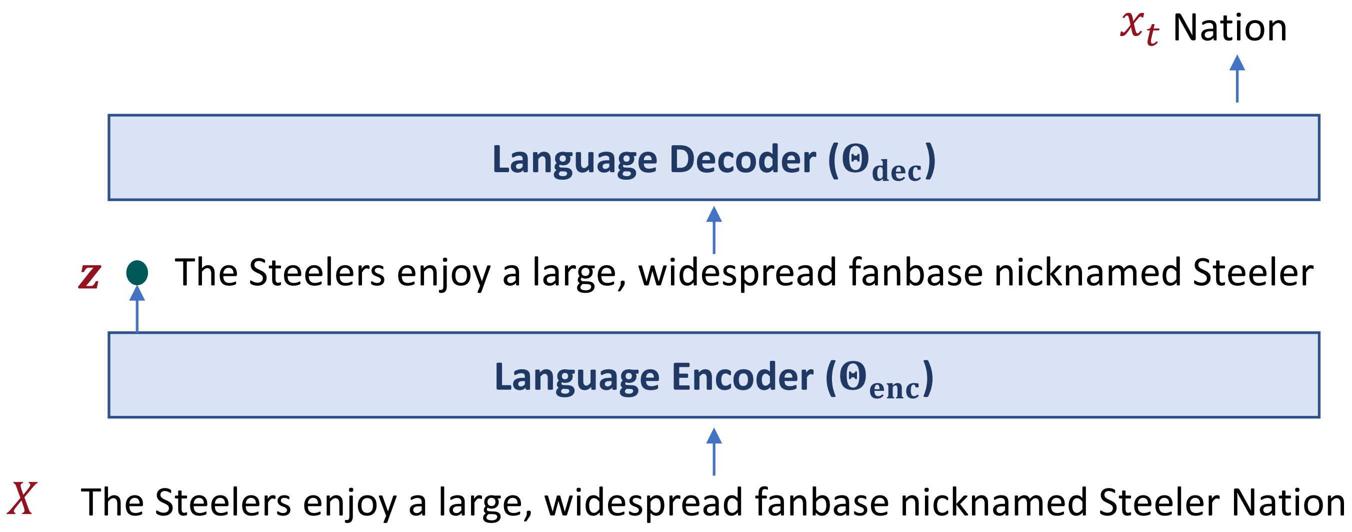


Auto-Encoder Language Modeling

The encoder and decoder can be various types of neural networks

- RNN, CNN, Transformers
- The signature is the information bottleneck $\boldsymbol{z}$ between encoder and decoder

- Advantage of Auto-Encoder language modeling
  - Explicit learning towards the sequence embedding $\boldsymbol{z}$
  - Allows various operations to convey prior knowledge to $\boldsymbol{z}$ for generation, especially for vision-alike modalities
  - Aligns with language representation tasks that need sequence level embeddings



Early experiments with decoder and auto-encoder

Evaluation set up:

- Task: IMDB sentiment classification
  - Given the text of a review from IMDB, classify whether positive or negative

Table 1: Examples of IMDB sentiment classification task [1]

Text	Sentiment
This film is not at all as bad as some people on here are saying. I think it has got a decent horror plot and the acting seem normal to me. People are way over-exagerating what was wrong with this. It is simply classic horror, the type without a plot that we have to think about forever and forever. We can just sit back, relax, and be scared.	Positive
Looking for a REAL super bad movie? If you wanna have great fun, don't hesitate and check this one! Ferrigno is incredibly bad but is also the best of this mediocrity.	Negative

[1] Dai, Andrew M., and Quoc V. Le. "Semi-supervised sequence learning." NeurIPS 2015.

Early experiments with decoder and auto-encoder

Evaluation set up:

- Task: IMDB sentiment classification
- Pretraining: language modeling on 8 million IMDB movie reviews
- Neural network: LSTMs
  - Auto-Encoder: discard decoder, fine-tune encoder
  - Decoder: fine-tune decoder

One of the earliest explorations of language model pretraining, in 2015 [1]

[1] Dai, Andrew M., and Quoc V. Le. "Semi-supervised sequence learning." NeurIPS 2015.

Early experiments with decoder and auto-encoder

Evaluation set up:

- Task: IMDB sentiment classification
- Pretraining: language modeling on 8 million IMDB movie reviews
- Neural network: LSTMs
  - Auto-Encoder: discard decoder, fine-tune encoder
  - Decoder: fine-tune decoder

One of the earliest explorations of language model pretraining, in 2015 [1]

Table 2: Results on IMDB sentiment classification task [1]

Method	**Test Error Rate↓**
LSTM (No Pretraining, Finetune Only)	13.5%
Auto-Regressive LSTM Decoder (Pretrain→Finetune)	7.64%
Auto-Encoder LSTM Encoder (Pretrain→Finetune)	7.24%
Auto-Encoder LSTM Encoder (Pretrain + Finetune, Multi-Task)	14.7%

[1] Dai, Andrew M., and Quoc V. Le. "Semi-supervised sequence learning." NeurIPS 2015.

Early experiments with decoder and auto-encoder

Observations from Dai and Le [1]:

- Pretraining helps significantly, as a better initialization
  - Not only on accuracy but also on stability, and generalization ability
- Decoder LSTM as a representation model is slightly worse than encoder LSTM
- Mixing pretraining and supervised learning hurts.
  - It is pre-training.

Table 2: Results on IMDB sentiment classification task [1]

Method	**Test Error Rate↓**
LSTM (No Pretraining, Finetune Only)	13.5%
Auto-Regressive LSTM Decoder (Pretrain→Finetune)	7.64%
Auto-Encoder LSTM Encoder (Pretrain→Finetune)	7.24%
Auto-Encoder LSTM Encoder (Pretrain + Finetune, Multi-Task)	14.7%

[1] Dai, Andrew M., and Quoc V. Le. "Semi-supervised sequence learning." NeurIPS 2015.

GPT-1: Pretraining + Transformer Decoder

GPT-1 combines unsupervised pretraining and Transformer network

- Auto-regressive language modeling
- Transformer decoder

Another significant difference: Scale

- Much bigger network
  - Transformers are easier to train than LSTM
- More data
  - Books Corpus, ~1 billion words.

[2] Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

GPT-1: Experimental Setup

Evaluation Task: GLUE benchmark

- A set of language classification tasks
- Most informative task is Multi-Genre Natural Language Inference (MNLI)
    - Given a pair of statements, predict whether one entails, contradicts, or is neural to the other

Table 3: Examples of MNLI

Premise	**Hypothesis**	**Label**
Conceptually cream skimming has two basic dimensions - product and geography.	Product and geography are what make cream skimming work.	Neutral
Read for Slate 's take on Jackson's findings.	Slate had an opinion on Jackson's findings.	Entailment
In an increasingly interdependent world, many pressing problems that affect Americans can be addressed only through cooperation with other countries	We should be independent and stay away from talking and working with other nations.	Contradiction

[2] Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

GPT-1: Evaluation Results

Results on MNLI and GLUE Average

Table 4: GPT-1 Results on GLUE [2]

Method	**MNLI (ACC)**	**GLUE AVG**
Pretrained LSTM Decoder	73.7	69.1
Non Pretrained Transformer	75.7	59.9
Pretrained Transformer	81.1	75.0
Pretrained Transformer + LM Multi-Task Finetune	81.8	74.7

Transformer is a much stronger architecture than LSTM

- More power
- Much easier to train

Pretraining brings a huge advantage

- Mixing pretraining with finetuning does not really help

[2] Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

Early Insights on Pretraining and Transformer

Early glimpse of zero-shot task solving

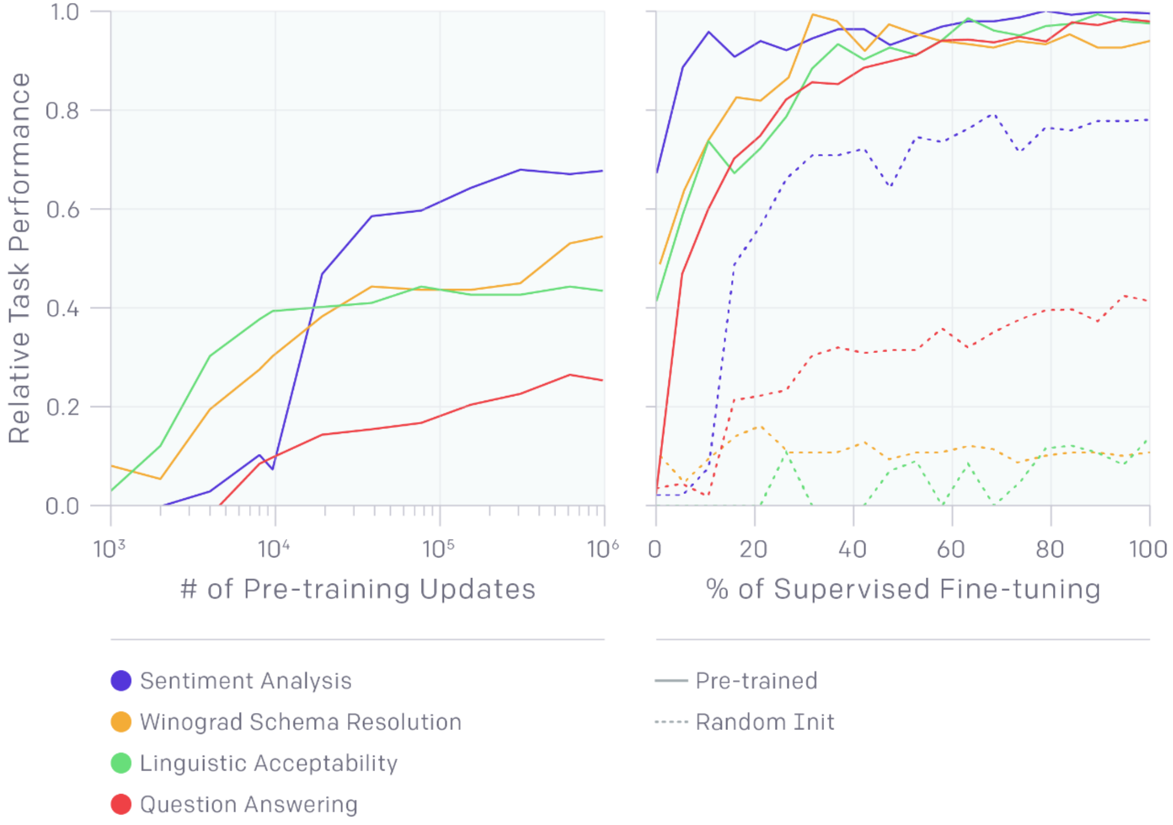


Figure 1: GPT-1 GLUE Performance at Different Stages [2]

[2] Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

Early Insights on Pretraining and Transformer

Early glimpse of zero-shot task solving

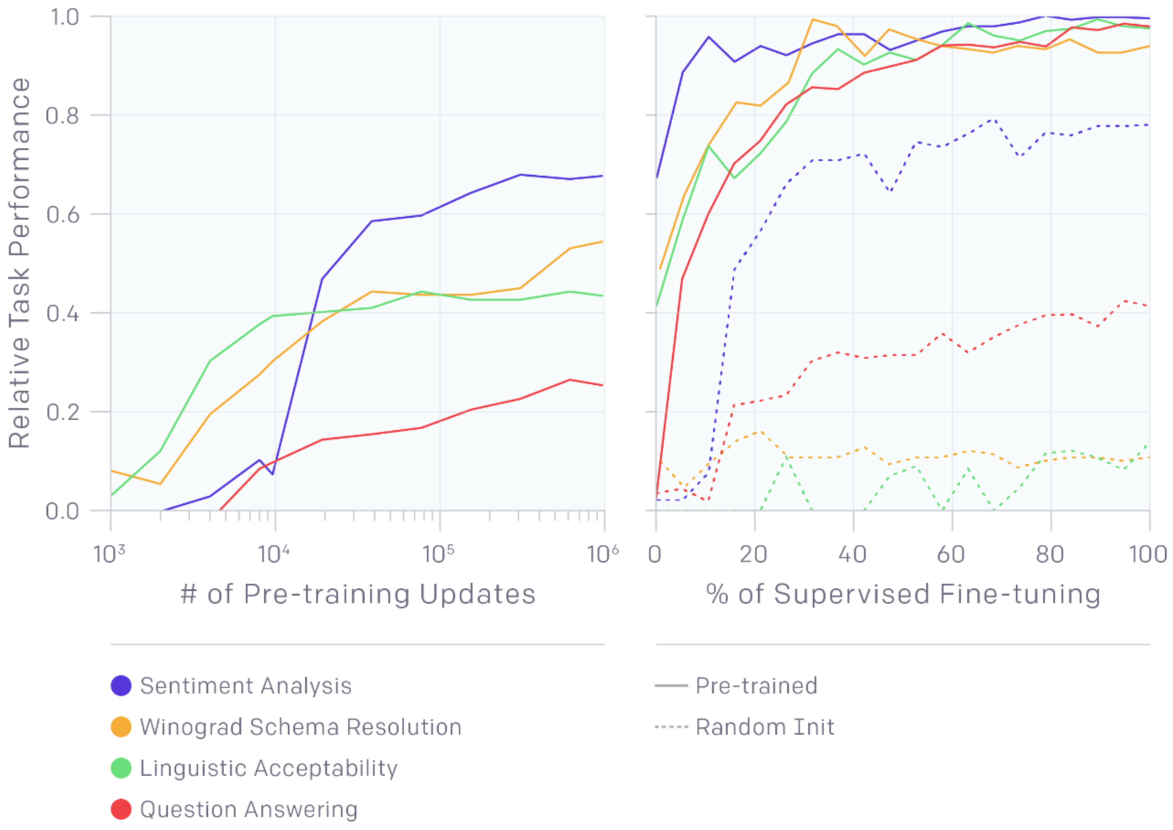


Figure 1: GPT-1 GLUE Performance at Different Stages [2]

Improving zero-shot with more pretraining Steps

- Burst increasements on some tasks
- Different benefits on different tasks

Many benefits as a starting point of finetuning

- Not only a faster initialization but a better one
- Necessary for tasks with limited labels

[2] Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

Pretraining by Denoising Task

Denoising training

- Reconstruct the original input from an input mixed with noises
  - Variety ways to construct the noisy input
- A classic unsupervised learning task used in many modalities
  - Language, vision, molecular, etc.

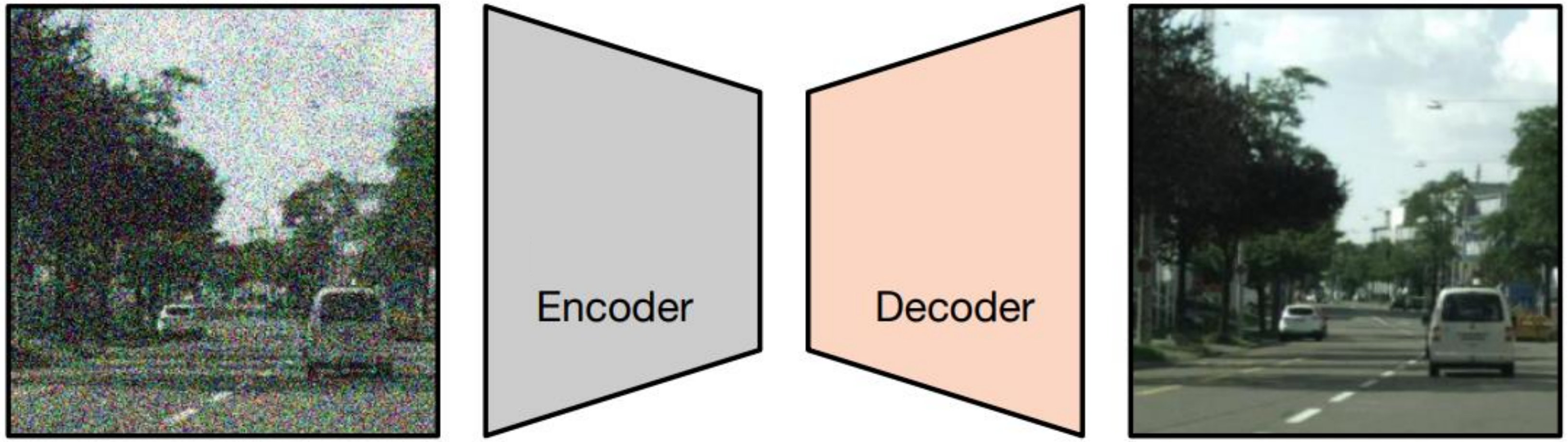


Figure 2: Example of Vision Denoising Training [3]

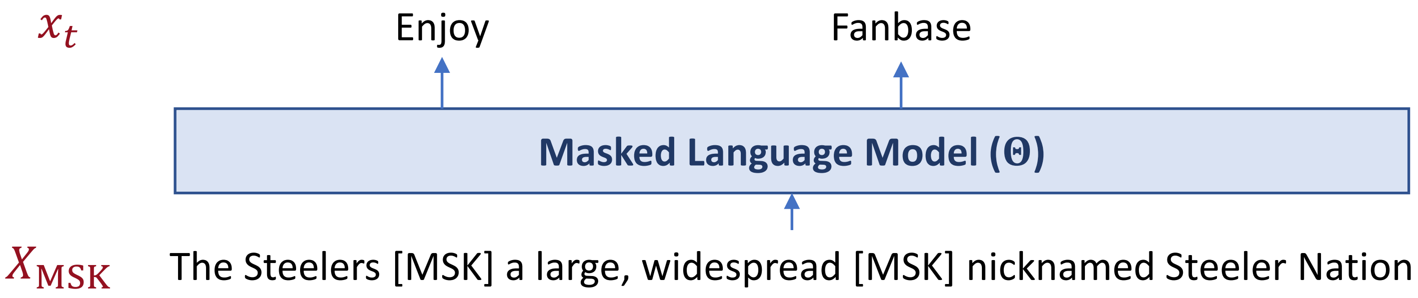
[3] Brempong, Emmanuel Asiedu, et al. "Denoising pretraining for semantic segmentation." CVPR 2022.

Masked Language Modeling

Masked Language Modeling, the denoising pretraining used in BERT

- Noisy Input: Text sequence with masked out token positions
- Reconstruction Target: Original tokens at masked out positions
- Let $X_{\text{MSK}} = \{x_1, \dots [\text{MSK}]_t \dots, x_n\}$ a text sequence of n tokens with positions $t \in M$ replaced with [MSK] tokens,
  - the Masked LM task is to maximize the likelihood of recovering masked out tokens:

$$L_{\text{MLM}}(X) = \sum_{t \in M} \log p(x_t | X_{\text{MSK}}; \Theta)$$



BERT Pretraining with Masked LM

BERT uses a bi-directional Transformer encoder as the language model

- Forward pass: $X_{\text{MSK}} \xrightarrow{\text{Transformer}} \boldsymbol{H} \xrightarrow{\text{MLM Head}} p_{\text{MLM}}(x|\boldsymbol{h}_t)$
- Mask LM Head: $p_{\text{MLM}}(x|\boldsymbol{h}_i) = \dfrac{\exp(\boldsymbol{x}^T \boldsymbol{h}_t)}{\sum_{x_i \in V} \exp \boldsymbol{x}_i^T \boldsymbol{h}_t}$
- Mask LM Loss: $L_{\text{MLM}} = \text{E}(-\sum_{t \in M} \log p_{\text{MLM}}(x_t|\boldsymbol{h}_t))$

Where:

- $\boldsymbol{x}$ the embedding of token x
- $\boldsymbol{H}, \boldsymbol{h}_t$ the last layer's representation of Transformer and the one for the t-th position.

BERT: Experimental Setup

Notable hyper-parameters

Table 5: BERT base and large configurations

	Total Parameters	Transformer Layers	Hidden Dimensions	Sequence Length	Pretraining Corpus	Pretraining Steps
$\text{BERT}_{\text{base}}$	110M	12	768	512	Wikipedia (2.5 billion words)+ BookCorpus (0.8b)	128K tokens/batch * 1M steps
$\text{BERT}_{\text{large}}$	340M	24	1024	512		

- Both became standard experimental settings in the pretraining literature
- Base setting is chosen to be close to GPT-1 for comparison

Other important setups

- Mask fraction: 15%
- Optimizer: Adam with warm up

BERT: Experimental Setup

Evaluation Tasks: GLUE, SQuAD, and many more

SQuAD: Question answering, reading comprehension style

- Given a natural language question and a passage, find the span (n-gram) answer in the passage
- Evaluate by matching the target answer phrase

Table 6: SQuAD Example

Question:	What kind of music does Beyonce do?
Passage:	Beyoncé's music is generally R&B, but she also incorporates pop, soul and funk into her songs. 4 demonstrated Beyoncé's exploration of 90s-style R&B, as well as further use of soul and hip hop than compared to previous releases.…
Target Answer:	R&B

- A good representative of several types of NLP tasks:
  - Knowledge-intensive: Questions require “human knowledge” to answer
  - Token-level tasks: Label prediction at token level
- One of the early QA experiences in commercial search engines (extractive QA)

[4] Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." NAACL-HLT. 2019.

BERT: Evaluation Results

Results on MNLI, GLUE Average, and SQuAD 1.1 Develop set

Table 7: BERT Evaluation Results [4]

	MNLI (ACC)	GLUE AVG	SQuAD (F1)
ELMO	76.3	71.0	85.6
GPT-1	81.8	75.1	n.a.
$\mathrm{BERT}_{\mathrm{base}}$	84.0	79.6	88.5
$\mathrm{BERT}_{\mathrm{large}}$	86.3	82.1	90.9

Much stronger results than GPT-1

- More flexibile architecture (allow bidirectional attention path)
- More data (Wiki + BookCorpus)

Significant gains by scaling from base to large

[4] Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." NAACL-HLT. 2019.

BERT: Analysis

Benefits of Masked LM

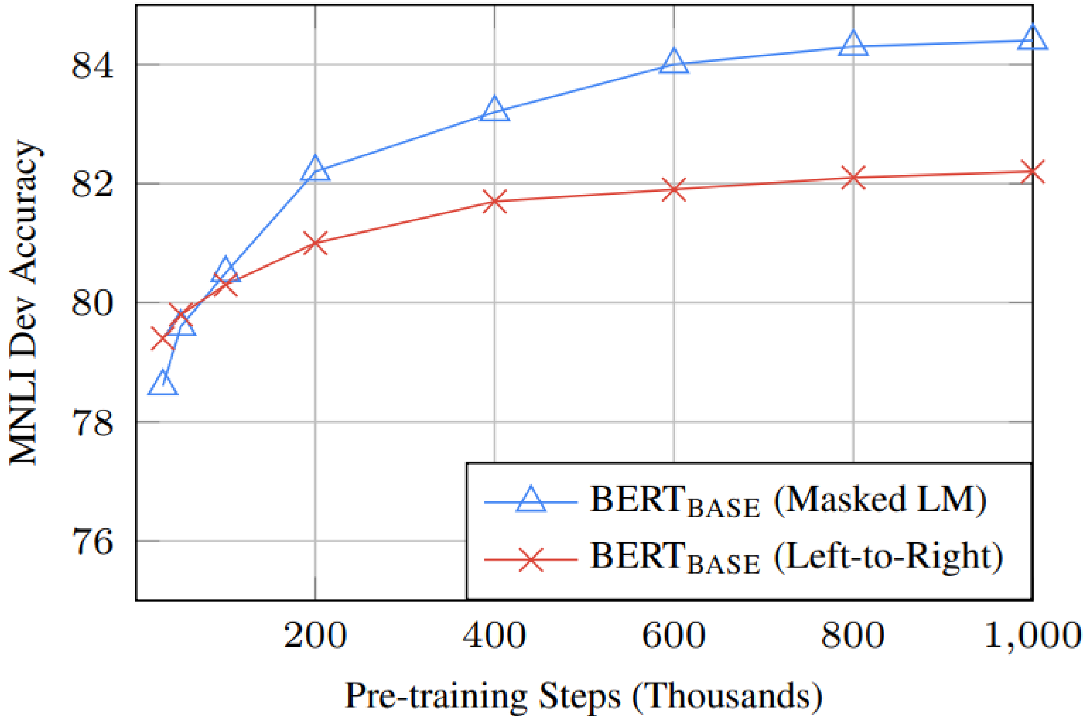


Figure 3: BERT finetuned accuracy after different pretraining steps with Masked LM and Auto-regressive LM [4]

Significant benefits from using Masked LM

- Hard to apply MLM on decoder only models

Auto-regressive LM starts faster

- But quickly by-passed by Masked LM

More Finessed Denoising Task: Span Masking

Span Masking: Instead of randomly sampled token positions, masking out more spans (continuous positions)

$x_{t:t+3}$ fanbase nicknamed steeler

Masked Language Model (Θ)

X_{SpanMSK} The Steelers enjoy a large, widespread [MSK] [MSK] [MSK] nation

[5] Joshi, Mandar, et al. "SpanBERT: Improving pre-training by representing and predicting spans." TACL 2020.

More Finessed Denoising Task: Span Masking

Span Masking: instead of randomly sampled token positions, masking out more spans (continuous positions)

$x_{t:t+3}$ fanbase nicknamed steeler

Masked Language Model (Θ)

X_{SpanMSK} The Steelers enjoy a large, widespread [MSK] [MSK] [MSK] nation

- Span sampling:
  - Sample a span length (# of tokens) from a geometric distribution
  - Randomly sample a starting point of the span to mask
  - Till reached total mask fraction (15%)

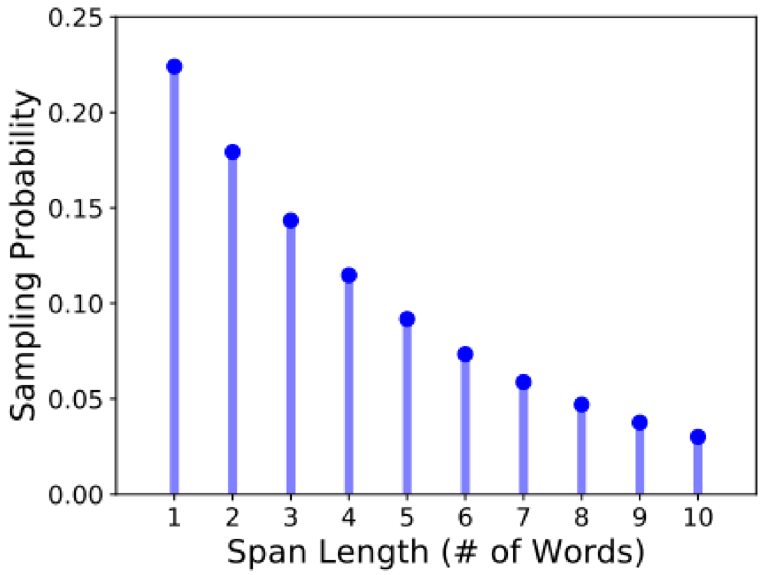


Figure 4: Geometric distribution used to sample span length in SpanBERT [5]

[5] Joshi, Mandar, et al. "SpanBERT: Improving pre-training by representing and predicting spans." TACL 2020.

More Finessed Denoising Task: Span Masking

Span Masking: instead of randomly sampled token positions, masking out more spans (continuous positions)

$x_{t:t+3}$ fanbase nicknamed steeler

Masked Language Model (Θ)

X_{SpanMSK} The Steelers enjoy a large, widespread [MSK] [MSK] [MSK] nation.

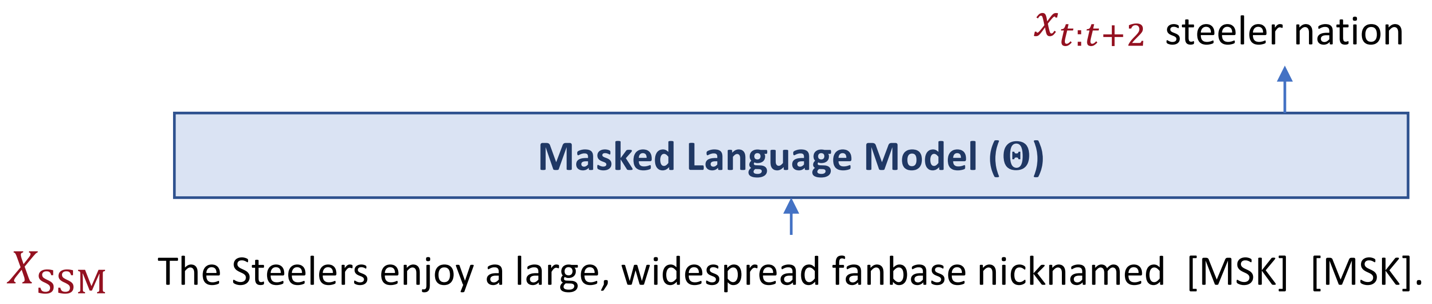
Benefits:

- A little higher granularity (tokens to phrases), thus harder/more semantical?
- Aligns well with some downstream applications, e.g., SQuAD

[5] Joshi, Mandar, et al. "SpanBERT: Improving pre-training by representing and predicting spans." TACL 2020.

More Finessed Denoising Task: Salient Span Masking

Salient Span Mask (SSM): Masking out spans corresponding to entities and attributes (salient)



First use fine-tuned BERT to tag named entities and rules to tag dates (salient spans)

Sample span mask from salient spans

Benefits:

- A lightweight way of introducing knowledge
- Directly targeting knowledge-intensive tasks, e.g., dates

Recap: Autoregressive LM and Masked LM

Table 8: Recap of Autoregressive LM and Masked LM

	Autoregressive LM	**Masked LM**
Neural Architecture	More suited for decoder	Encoder and decoder
Training Density	All Token Positions	15% of Masked Positions
Converging Speed/Stability	Fast and stable	Slower and less stable
Task Fit	Generation	Representation
Notable Models	GPT-*	BERT

Combination of Auto-Regressive and Masked LM

Various efforts to combine the benefits of Auto-Regressive LM and Masked LM

- One model for both generation and representation
- Better training effectiveness from multi-task learning?

Notable examples:

- UniLM: Dong, Li, et al. "Unified language model pre-training for natural language understanding and generation." NeurIPS 2019.
- XL-NET: Yang et al. “XL-NET: Generalized autoregressive pretraining for language understanding." NeurIPS 2019.

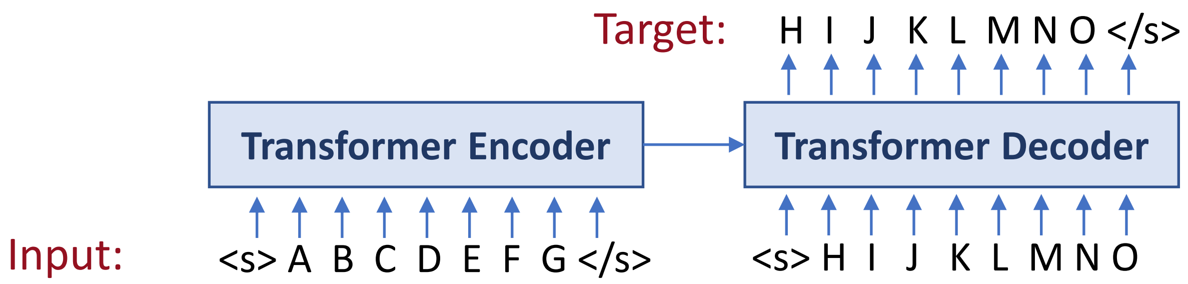
Transformer Encoder-Decoders

Much of the difference of auto-regressive versus Masked LM also resides in the Transformer architecture:

- Encoder: bi-directional representation power
- Decoder: natural generation

Transformer Encoder-Decoder enjoy the benefits of both

- Flexible for various types of denoising tasks
- Support different downstream applications with either side, or both together



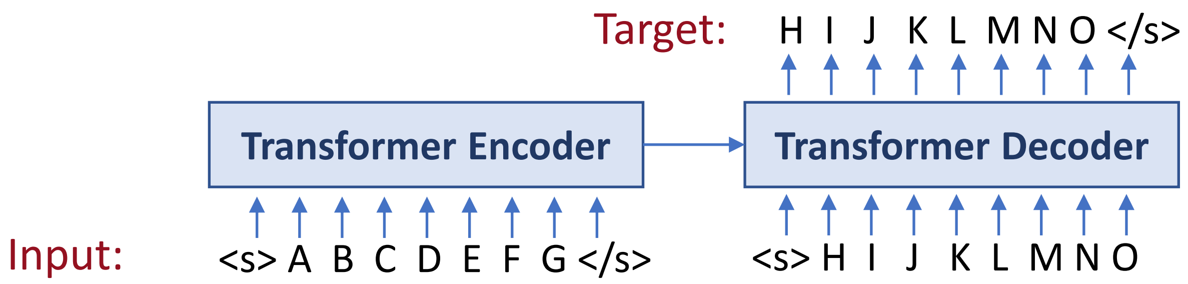
T5: Text-to-Text Transfer Transformers

Encoder-Decoder Transformer pretrained with language modeling tasks

- The flexibility allowed T5 to explore many different denoising tasks

Table 9: Pretraining Tasks Explored in T5 [7].

Objective	Inputs	Targets
Prefix language modeling	Thank you for inviting	me to your party last week .
BERT-style Devlin et al. (2018)	Thank you <M> <M> me to your party apple week .	*(original text)*
Deshuffling	party me for your to . last fun you inviting week Thank	*(original text)*
MASS-style Song et al. (2019)	Thank you <M> <M> me to your party <M> week .	*(original text)*
I.i.d. noise, replace spans	Thank you <X> me to your party <Y> week .	<X> for inviting <Y> last <Z>
I.i.d. noise, drop tokens	Thank you me to your party week .	for inviting last
Random spans	Thank you <X> to <Y> week .	<X> for inviting me <Y> your party last <Z>



[7] Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." JMLR. 2020.

T5 Pretraining Task Studies

Use of T5: fine-tuned with

- Encoder takes task input
- Decoder generating the label word, e.g., “Entailment” for MNLI

Table 9: Pretraining Tasks Results with T5 base [7].

Denoising Task	GLUE AVG	SQuAD
Auto-Regressive LM	80.7	78.0
De-shuffling	73.2	67.6
Masked-LM, Reconstruct All	83.0	80.7
Replace Corrupted Spans	83.3	80.9
Drop Corrupted Tokens	84.4	80.5

[7] Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer.“ JMLR. 2020.

T5 Pretraining Task Studies

Use of T5: fine-tuned with

- Encoder takes task input
- Decoder generating the label word, e.g., “Entailment” for MNLI

Table 9: Pretraining Tasks Results with T5 base [7].

Denoising Task	GLUE AVG	SQuAD
Auto-Regressive LM	80.7	78.0
De-shuffling	73.2	67.6
Masked-LM, Reconstruct All	83.0	80.7
Replace Corrupted Spans	83.3	80.9
Drop Corrupted Tokens	84.4	80.5

- Different variations of Masked-LM style denoising task performed similarly

[7] Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer.“ JMLR. 2020.

BART Pretraining Tasks

Various denoising tasks explored with BART's encoder-decoder

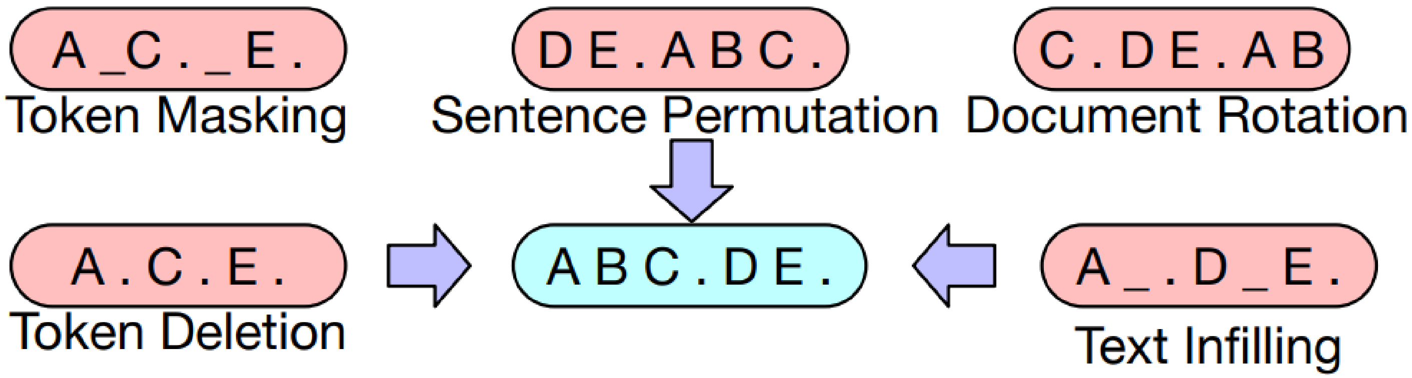


Figure 5: Denoising Tasks Explored in BART [8]

- Both sentence level and token level
- Flexible architecture enabled reconstruction from various types of noises

[8] Lewis, Mike, et al. "BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension." ACL. 2020.

BART Pretraining Task Studies

Use of BART:

- Representation style tasks: feed same inputs to both encoder and decoder, use decoder representations
- Generation: use decoder

Table 10: Pretraining Tasks Results with BART base [7].

	MNLI (Acc)	SQuAD (F1)
Document Rotation	75.3	77.2
Sentence Shuffling	81.5	85.4
Token Masking	84.1	90.4
Token Deletion	84.1	90.4
Text Infilling	84.0	90.8
Text Infilling + Sentence Shuffling	83.8	90.8

BART Pretraining Task Studies

Use of BART:

- Representation style tasks: feed same inputs to both encoder and decoder, use decoder representations
- Generation: use decoder

Table 10: Pretraining Tasks Results with BART base [7].

	MNLI (Acc)	SQuAD (F1)
Document Rotation	75.3	77.2
Sentence Shuffling	81.5	85.4
Token Masking	84.1	90.4
Token Deletion	84.1	90.4
Text Infilling	84.0	90.8
Text Infilling + Sentence Shuffling	83.8	90.8

- Different variations of Masked-LM style denoising task performed similarly

Pretraining Tasks: Summary

Classic Auto-Regressive LM and BERT’s Masked LM are very effective

- A solid foundation to scale up

Early explorations on variant language modeling tasks do not obtain much general improvements

- Application-specific gains are more observed
- All in forms of (rule-based random noise + reconstruction target)

Sequence level tasks not showing much benefits on tasks like GLUE and SQuAD

- Hard to fathom strong “semantic”, “knowledge”, or “intelligence” from some sequence level tasks

TL;DR: for base scale LMs

- Generation→ Auto-Regressive LM
- Representation→ Masked LM

Questions?

References: Pretraining Objectives

- [Pretraining] Dai, Andrew M., and Quoc V. Le. "Semi-supervised sequence learning." Advances in neural information processing systems 28 (2015).
- [ELMO] Sarzynska-Wawer, Justyna, et al. "Detecting formal thought disorder by deep contextualized word representations." Psychiatry Research 304 (2021): 114135.
- [GPT] Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).
- [BERT] Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." In Proceedings of NAACL-HLT, pp. 4171-4186. 2019.
- [XL-NET] Yang, Zhilin, et al. "XLNet: Generalized autoregressive pretraining for language understanding." Advances in neural information processing systems 32 (2019).
- [SpanBERT] Joshi, Mandar, et al. "Spanbert: Improving pre-training by representing and predicting spans." Transactions of the Association for Computational Linguistics 8 (2020): 64-77.
- [REALM] Guu, Kelvin, et al. "Retrieval augmented language model pre-training." International conference on machine learning. PMLR, 2020.
- [BART] Lewis, Mike, et al. “BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension." arXiv preprint arXiv:1910.13461 (2019).
- [T5] Raffel, Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." The Journal of Machine Learning Research 21.1 (2020): 5485-5551.