UL2: Unifying Language Learning Paradigms by Tay et al.

Zhangir Azerbayev, CPSC670

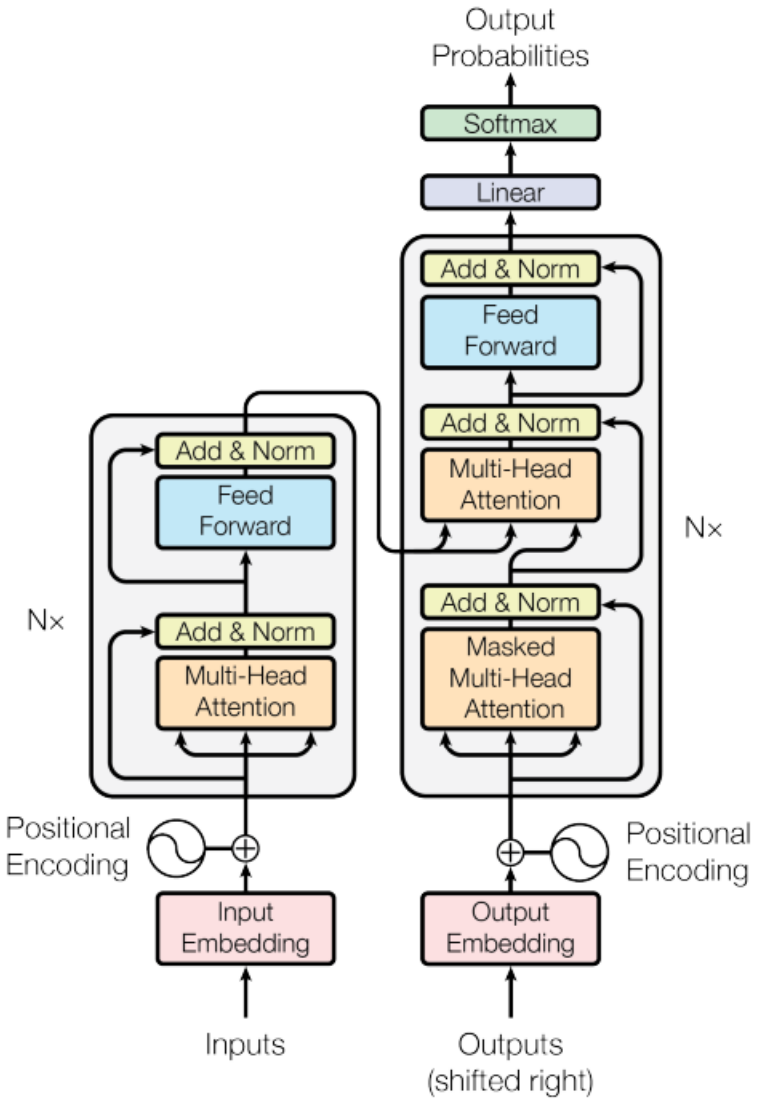
Spring 2023

Introduction

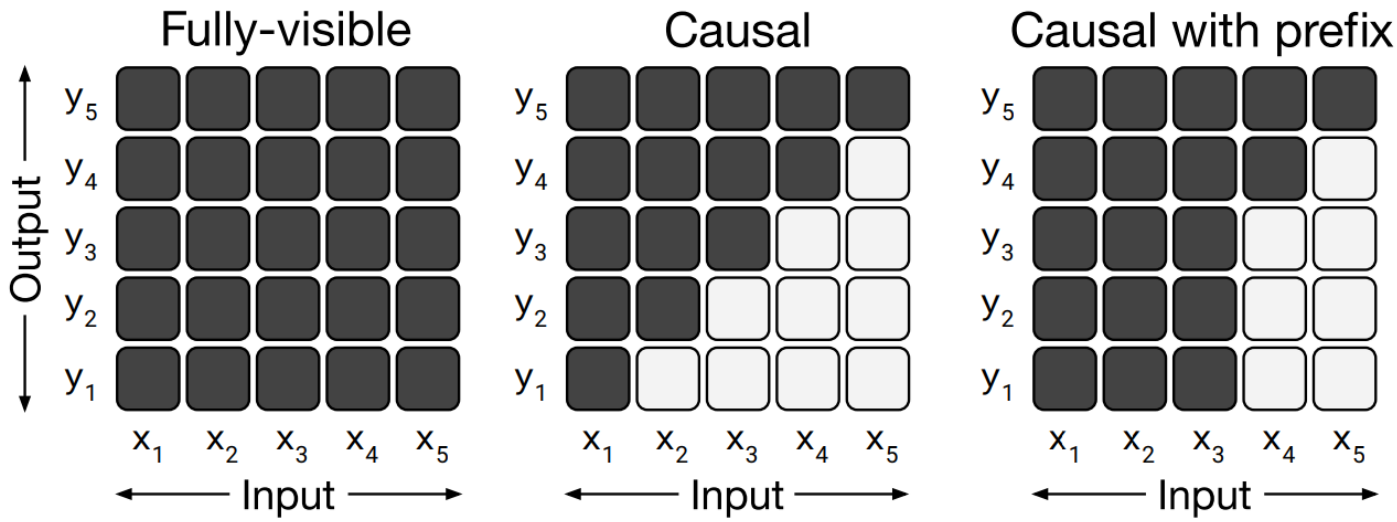
Already in this class, we have seen many competing paradigms for training transformer models on text

- Vaswani et al: encoder-decoder, seq2seq
- BERT: encoder-only, masked language modelling (MLM)
- GPT: decoder-only, causal language modelling (CLM)
- T5: encoder-decoder, prefix-LM and span corruption (SC)

Architectures



Pre-training objectives



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>

Tradeoffs

BERT style best for NLU,

GPT style best for text generation and in-context learning

Vaswani/T5 style best for seq2seq

Are these tradeoffs necessary, or is there a better way that Pareto-dominates all previous approaches?

UL2

Claims of the paper:

1. Architecture and pre-training objective are independent choices.
2. Encoder-decoder is slightly better than decoder-only
3. Training models using the UL2 objective, which is a mixture of different denoising objectives, Pareto-dominates any single pre-training objective.

Unifying Language Learning Paradigms

Single mathematical framework that encompasses self-supervised language learning objectives

Generate denoising tasks using a function $\textsc{SpanCorrupt}(\mu, r, n)$. The parameter μ defines the mean span length, r is the corruption rate, and n is the number of corrupted spans.

- Causal LM: $(\mu = L, r = 1, n = 1)$
- Prefix LM: $(\mu = L - P, r = 1 - P/L, n = 1)$
- BERT: $(\mu = 1, r = 0.15, n)$

UL2 Objective

Propose mixture of three denoising objectives:

Denoiser	Setting
R	$(\mu = 3, r = 0.15, n) \cup (\mu = 8, r = 0.15, n)$
S	$(\mu = {}^{L}/_{4}, r = 0.25, 1)$
X	$(\mu = 3, r = 0.5, n) \cup (\mu = 8, r = 0.5, n) \cup (\mu = 64, r = 0.15, n) \cup (\mu = 64, r = 0.5, n)$

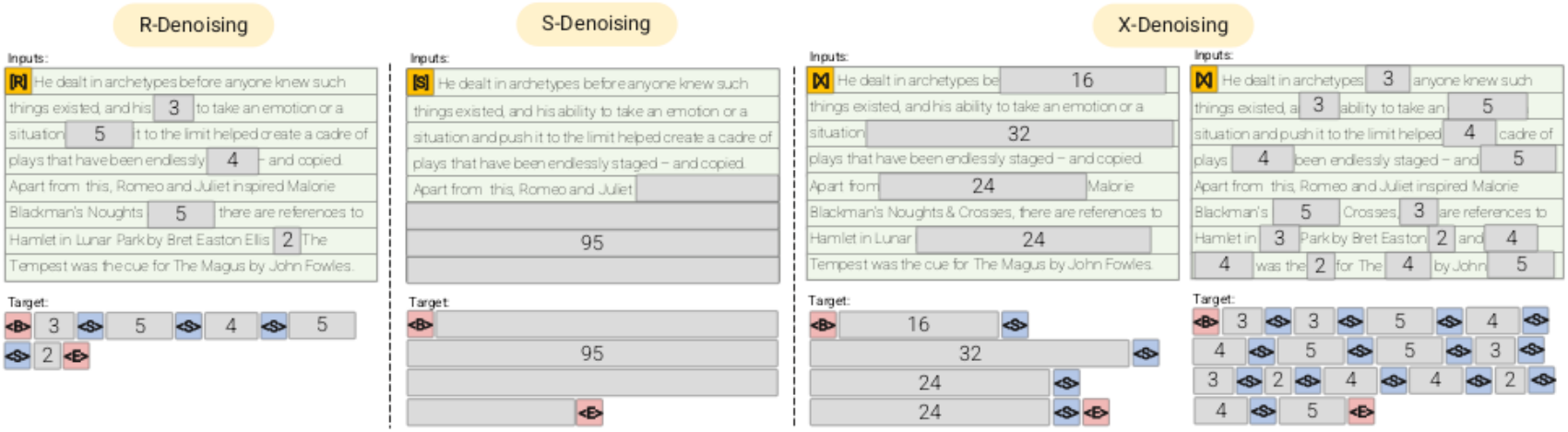


Figure 3: Mixture of denoisers for training UL2. Greyed out rectangles are masked tokens that are shifted to *'targets'* for prediction.

Architectures

Claim: Architecture and Pre-training objective are orthogonal choices

First, the authors argue that the encoder-only architecture is obsolete:

- Very restricted in text generation capabilities (single bit of supervision per sequence)
- Need task-specific classification heads

Architectures

What about encoder-decoder vs decoder-only?

An encoder-decoder model is roughly the same as decoder-only + prefix LM.

Decoder-only is roughly the same as encoder-decoder with an empty input sequence.

So architecture choice and pre-training objective are independent choices

Experimental Results

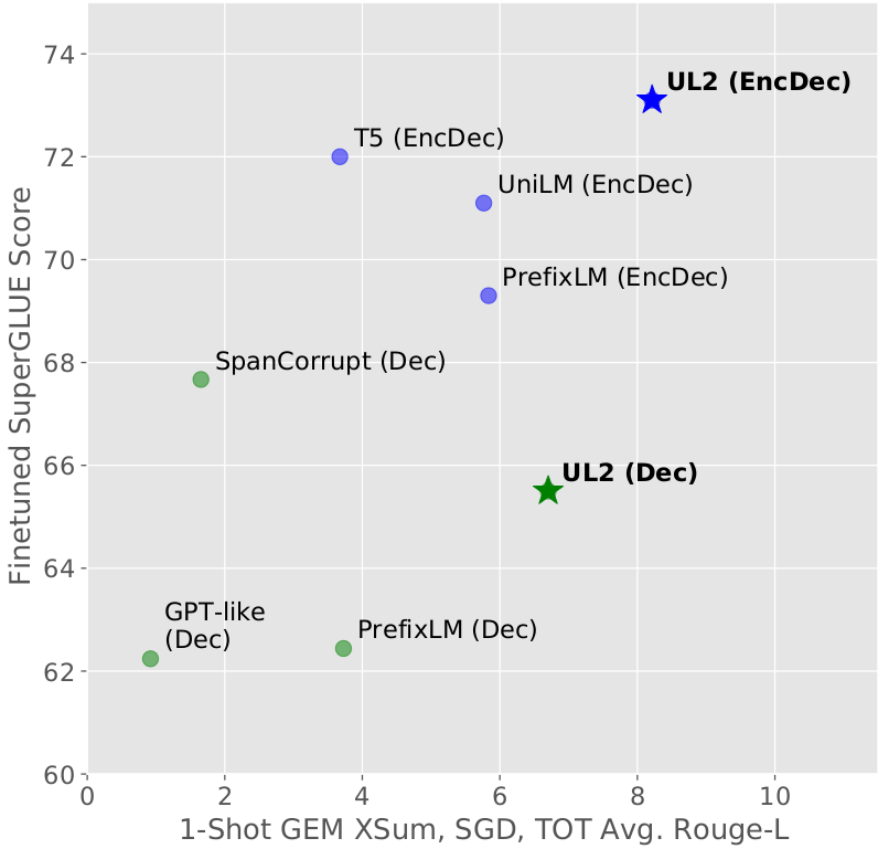
Compare UL2 to suite of baseline pre-training objectives

- Causal LM
- Prefix LM
- Span Corruption (SC)
- Span Corruption + LM (SCLM)
- UniLM (ULM) (hybrid objective similar to UL2 from prior work)

Train 167M for dec, 335M for encdec

Experimental Results

UL2 encdec Pareto-dominates other objectives in NLU and text generation at $\sim$ 100M scale.



Scaling UL2

Authors trained 20B encdec model with UL2.

Table 11: Results on One-Shot Summarization on XSUM.

Model	Rouge-1	Rouge-2	Rouge-L
LaMDA 137B	-	5.4	-
PaLM 62B	-	11.2	-
PaLM 540B	-	**12.2**	-
PaLM 8B	-	4.5	-
T5 XXL 11B	0.6	0.1	0.6
T5 XXL 11B + LM	13.3	2.3	10.7
UL2 20B	**25.5**	**8.6**	**19.8**

Discussion Questions

1. Do you agree with the author's claim that encoder-only architectures are obsolete? When, if ever, might you want to use one?
2. What additional experiments would you want to perform to strengthen/refute the case for UL2?
3. If you were an engineer at OAI/Google/DeepMind, would you use UL2 to train your next LLM? Why or why not?

Why I'm still skeptical

Encdec model trained with UL2 outperforms all baselines at $\sim$100M scale

However,

The most important capabilities of LLMs are in-context learning abilities that are emergent at large scales

We know that GPTs trained with CLM have excellent scaling behavior. It is not demonstrated that UL2 also does.

On other hand, recent work has shown that training for just a little bit with UL2 after pre-training a CLM dec model greatly improves performance (*Transcending Scaling Laws with 0.1% Extra Compute*, Tay et al.)