On the Role of Bidirectionality in Language Model Pre-Training

Mikel Artetxe **Jingfei Du** **Naman Goyal** **Luke Zettlemoyer** **Ves Stoyanov**
Meta AI
{artetxe,jingfeidu,naman,lsz,ves}@meta.com

Abstract

Prior work on language model pre-training has explored different architectures and learning objectives, but differences in data, hyperparameters and evaluation make a principled comparison difficult. In this work, we focus on bidirectionality as a key factor that differentiates existing approaches, and present a comprehensive study of its role in next token prediction, text infilling, zero-shot priming and fine-tuning. We propose a new framework that generalizes prior approaches, including fully unidirectional models like GPT, fully bidirectional models like BERT, and hybrid models like CM3 and prefix LM. Our framework distinguishes between two notions of bidirectionality—bidirectional context and bidirectional attention—and allows us to control each of them separately. We find that the optimal configuration is largely application-dependent (e.g., bidirectional attention is beneficial for fine-tuning and infilling, but harmful for next token prediction and zero-shot priming). We train models with up to 6.7B parameters, and find differences to remain consistent at scale. While prior work on scaling has focused on left-to-right autoregressive models, our results suggest that this approach comes with some trade-offs, and it might be worthwhile to develop very large bidirectional models.

1 Introduction

NLP has undergone a paradigm shift driven by pre-trained models like GPT and BERT (Bommasani et al., 2021). These models are trained on unlabeled corpora in a self-supervised fashion, and can be effectively adapted to downstream tasks either through conventional fine-tuning (Devlin et al., 2019) or few-shot priming (Brown et al., 2020).

Despite their widespread use, there is not a universal formula to pre-train language models: prior work has explored different architectures and learning objectives, often focusing on different applications. For instance, BERT (Devlin et al., 2019) pre-trained masked language models for NLU fine-tuning, BART (Lewis et al., 2020) pre-trained seq2seq models on denoising for both NLU and generation tasks, and GPT-3 (Brown et al., 2020) scaled autoregressive language models focusing on zero- and few-shot priming. However, such models differ on many factors in addition to their architecture and learning objective (e.g., the pre-training data, compute and hyperparameters), making a principled comparison difficult. Motivated by that, Raffel et al. (2020) presented a comprehensive study exploring various pre-training objective and architecture variants in a controlled environment. However, they conducted most of the exploration using small models, while recent work has found that different approaches behave differently at scale (Tay et al., 2022a,b), and their evaluation was limited to fine-tuning.

In this paper, we focus on a key factor that differentiates many pre-training approaches—bidirectionality—and study it in different settings as a function of scale. We propose a new framework that distinguishes between two notions of bidirectionality: **bidirectional context** (whether the prediction of a given token is conditioned on both the right and the left context, or only on either of them), and **bidirectional attention** (whether there are blocks of tokens that can all attend to each other, contrasting with triangular attention masking). Our framework offers knobs to control each of them separately, generalizing several previous approaches (e.g. BERT leverages both types of bidirectionality, GPT does not use any, prefix LMs only leverage bidirectional attention, and CM3 only leverages bidirectional context).

We train a total of 24 models covering 6 variants of our framework and 5 model sizes with up to 6.7B parameters, and evaluate them on 4 settings: language modeling, text infilling, zero-shot priming, and fine-tuning. We find that bidirectional attention and context have a different impact depending on

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 4002 - 4014
December 7-11, 2022 ©2022 Association for Computational Linguistics

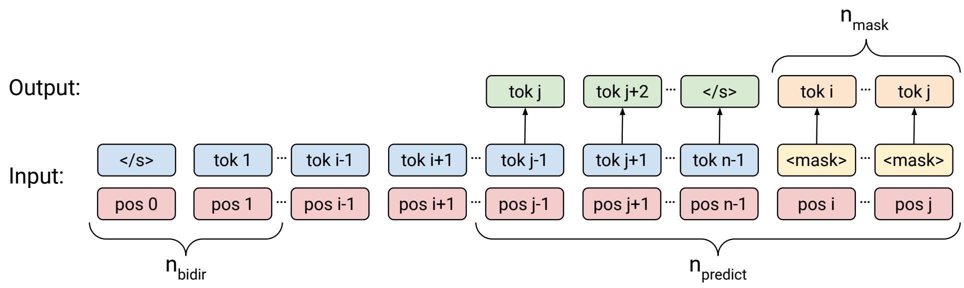

Figure 1: **Proposed framework.** Starting from the original document, we mask n_{mask} tokens at random and move them—along with their positional embeddings—to the end. We define our loss over the last n_{predict} tokens, predicting the masked token for the last n_{mask}, and the next token for the remaining $n_{\text{predict}} - n_{\text{mask}}$. We use bidirectional attention over the first n_{bidir} tokens, and unidirectional attention over the rest. Refer to Appendix A for a more detailed description.

Name	n_{mask}	n_{bidir}	n_{predict}	Related models
NxtUni	0	0	n	GPT (Radford et al., 2018, 2019; Brown et al., 2020)
NxtPre†	0	$\mathrm{U}(1, n)$	$n - n_{\text{bidir}}$	Prefix LM (Raffel et al., 2020; Wu et al., 2021)
MskUni	$\mathrm{B}(n, 0.15)$	0	n_{mask}	–
MskBi	$\mathrm{B}(n, 0.15)$	n	n_{mask}	BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019)
HybUni†	$\mathrm{B}(n, 0.15)$	0	n	CM3 (Aghajanyan et al., 2022)
HybPre†	$\mathrm{B}(n, 0.15)$	$\mathrm{U}(1, n)$	$\max(n - n_{\text{bidir}}, n_{\text{mask}})$	–

Table 1: **Variants of the proposed framework explored in this work.** n denotes the document length; $\mathrm{B}(n, p)$ denotes the binomial distribution; $\mathrm{U}(a, b)$ denotes the discrete uniform distribution. †We set $n_{\text{bidir}} = 0$ and $n_{\text{mask}} = 0$ with probability $p = 0.1$, so that the model gets more exposure to regular language modeling.

the use case, and there is not a single configuration that is optimal for all scenarios. Moreover, we find this behavior to remain consistent at the scale range considered in this study. With recent scaling work focusing on fully unidirectional models, this suggests that there is potential for alternative architectures and learning objectives that might be better suited for other use cases.

2 Proposed framework

As illustrated in Figure 1, we propose a generalized framework to pre-train transformer models on unlabeled corpora. Our framework supports both unidirectional and bidirectional attention, as well as next token prediction and single-token infilling, using the following **parameters** to balance them:

- n_{bidir} controls the length of the prefix using bidirectional attention, whereas the rest of the document uses unidirectional attention. More concretely, we set the attention mask so that the ith token can attend to the jth token if and only if $j \leq \max(i, n_{\text{bidir}})$.
- n_{mask} controls how many tokens are masked. Masked tokens are moved to the end along with their positional embeddings.
- n_{predict} controls the length of the suffix for which we define our supervisory signal. We use the cross-entropy loss to train the model, predicting the masked tokens for the last n_{mask}, and the next token for the remaining $n_{\text{predict}} - n_{\text{mask}}$.[1]

As such, our framework allows us to vary the **two notions of bidirectionality** discussed above: n_{bidir} controls the weight of bidirectional attention, whereas n_{mask} and n_{predict} control the weight of bidirectional context. In addition, larger values of n_{predict} result in more tokens of supervision.

Table 1 summarizes the specific **variants** of this general framework that we explore in our experiments, along with a descriptive name that we will use to refer to each of them. Some variants are equivalent or closely related to existing approaches. In particular, NxtUni is equivalent to conventional autoregressive language models, and NxtPre is equivalent to prefix language models. MskBi is closely related to the RoBERTa objective,[2] except

[1]We set $n_{\text{predict}} \leq n - n_{\text{bidir}} + n_{\text{mask}}$ so we only predict tokens that are either masked or cannot attend to themselves.

[2]Moving masked tokens to the end becomes irrelevant when $n_{\text{bidir}} = n$, as their positional embeddings move with them and transformers operate over sets.

size	*cost*	*l*	*d*	*h*	*bs*	*lr*
125M	0.11	12	768	12	0.5M	6e-4
355M	0.31	24	1024	16	0.5M	3e-4
1.3B	1.11	24	2048	32	1M	2e-4
2.7B	2.23	32	2560	32	1M	1.6e-4
6.7B	5.49	32	4096	32	2M	1.2e-4

Table 2: **Model details**. *size*: number of parameters, *cost*: training ZFLOPs, l: layers, d: hidden dimension, h: attention heads, *bs*: batch size, *lr*: learning rate. All models are trained for 100B tokens with a maximum sequence length of 1024 tokens. We estimate training ZFLOPs analytically following Artetxe et al. (2021).

that we do not replace 10% of the masked tokens with the original or a randomly picked one. HYBUNI is similar to the CM3 objective, except that we mask individual tokens instead of spans and we draw the number of masks from a binomial distribution. Finally, we introduce MSKUNI as a variant of MSKBI using unidirectional attention (or, from another perspective, a variant of HYBUNI predicting masked tokens alone), and HYBPRE as a variant of HYBUNI using a bidirectional attention prefix.

3 Experimental settings

3.1 Models

For each variant in Table 1, we train models at different scales using the same settings as Artetxe et al. (2021), which at the same time roughly follow Brown et al. (2020). So as to reduce the computational cost of our exploration, we differ from Artetxe et al. (2021) in two ways: (i) we use a maximum sequence length of 1024 tokens instead of 2048, and (ii) we train for 100B tokens instead of 300B. At the same time, we only train 125M and 355M models for the NXTPRE and MSKUNI variants. Table 2 summarizes the settings that we use for each model.

We use the same training data as Artetxe et al. (2021), which combines BookCorpus (Zhu et al., 2015), CC-News (Nagel, 2016), OpenWebText (Gokaslan and Cohen, 2019), CC-Stories (Trinh and Le, 2018), and English CC100 (Wenzek et al., 2020), totalling 112B tokens. Following them, we also use the same BPE encoding as GPT-2 (Radford et al., 2019) with a vocabulary of 50k.

Our implementation is based in fairseq (Ott et al., 2019). We apply the procedure described in §2 to each document separately, and combine multiple documents into a single sequence to speed up training.[3] As such, we move the masked tokens to the end of each document (as opposed to the end of the whole sequence), and apply a bidirectional attention prefix to each document rather than the sequence as a whole.[4]

3.2 Evaluation

We evaluate our models in the following settings:

Language modeling. We evaluate the ability of our models to predict the next token in a sequence as measured by perplexity.[5] Different from training, we do not concatenate different documents into the same sequence, and instead score each document as a separate sequence.[6] Given that NXTPRE and HYBPRE are primarily trained to predict the last part of a document conditioned on the first part, we also measure the perplexity at predicting the last 20% tokens in each document conditioned on the first 80%. So as to understand whether using bidirectional attention in the prefix is useful to that end, we try different values of n_{bidir} according to a ratio r_{bidir}, so that $n_{\text{bidir}} = r_{\text{bidir}} \times n_{\text{prefix}}$ and $n_{\text{prefix}} = 0.8n$ is the length of the prefix we are conditioning on.

Single token infilling. We mask a single word in each document at random, and measure the accuracy at predicting it.[7] To that end, we use the same procedure used for training (illustrated in Figure 1), which moves the mask token to the end of the sequence.[8] This approach is not suitable for models trained exclusively on next token prediction like NXTUNI and NXTPRE, as they can only be conditioned on the right context. However, one can still use such models for infilling in a generative fashion, replacing the masked token with each element in the vocabulary, scoring the resulting sequences autoregressively, and predicting the token yield-

[3] We achieve this using `–sample-break-mode complete` in fairseq. This is different from Artetxe et al. (2021), who concatenated all documents and split the resulting sequence into non-overlapping blocks without respecting document boundaries (`–sample-break-mode none`).

[4] As a consequence, a given token cannot attend to tokens in future documents even when $n_{\text{bidir}} = n$, but all tokens can attend to tokens in previous documents.

[5] We exclude MSKBI and MSKUNI as they are not trained on next token prediction.

[6] This corresponds to the `–sample-break-mode complete_doc` option in fairseq.

[7] Similar to language modeling evaluation, we feed each document as a separate sequence.

[8] For models trained with a bidirectional attention prefix, we try different values of r_{bidir} at inference time, so that $n_{\text{bidir}} = r_{\text{bidir}} \times n$.

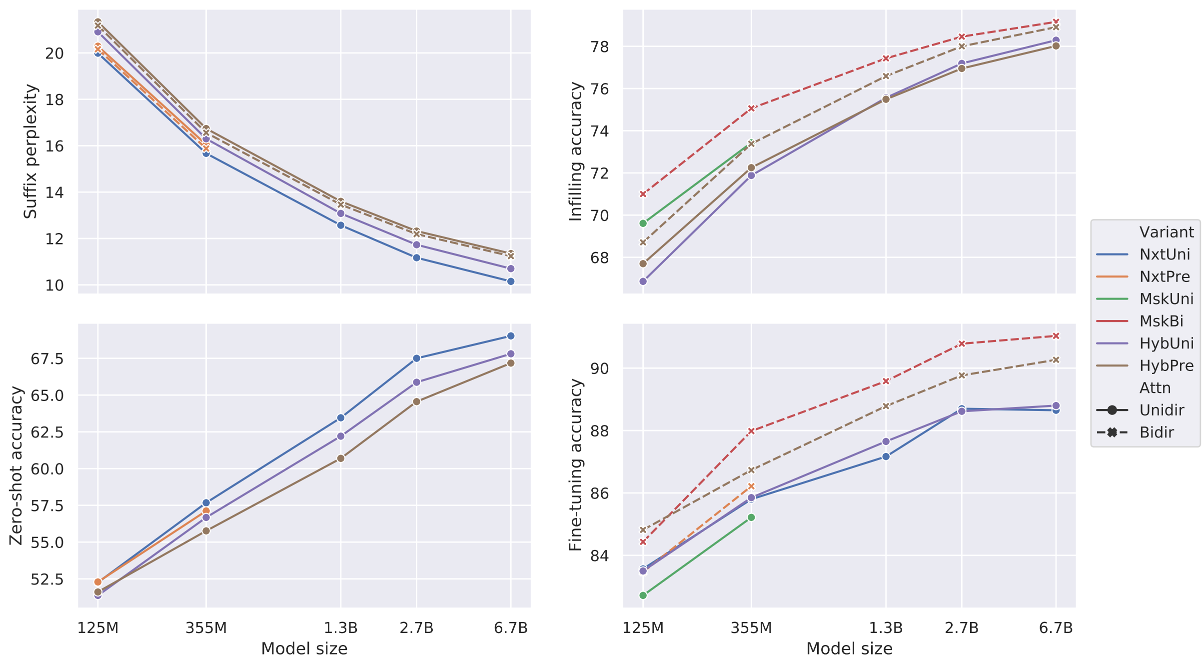

Figure 2: **Main results.** *Unidir* and *Bidir* denote using $n_{\text{bidir}} = 0$ and $n_{\text{bidir}} = n$ after pre-training, respectively (or $n_{\text{bidir}} = n_{\text{prefix}}$ for suffix perplexity).

ing the highest scoring sequence. In addition to our primary evaluation, we compare both of these approaches, which we refer to as *infill* (direct infilling) and *full* (full sequence scoring). Given that *full* can be prohibitively expensive when considering the full vocabulary, we constrain the set of options to the top 32 candidates generated by the 125M MskBi model.[9]

Zero-shot priming. We evaluate our models on zero-shot priming using the exact same settings and tasks as Artetxe et al. (2021), which comprises ReCoRD (Zhang et al., 2018), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), WinoGrande (Sakaguchi et al., 2020), StoryCloze (Mostafazadeh et al., 2016) and OpenBookQA (Mihaylov et al., 2018). These are all multiple choice tasks, so we score the populated prompt corresponding to each option in an autoregressive fashion and predict the highest scoring one.[10] However, when the options differ in a single token—as it is common for classification tasks with single-token verbalizers—one can also score such token directly in an infilling fashion. So as to understand how both approaches compare, we further evaluate our models on MNLI (Williams et al., 2018), using a single-token verbalizer placed in the middle of the prompt.[11]

Fine-tuning. We experiment with the following tasks from GLUE (Wang et al., 2019): COLA (Warstadt et al., 2019), MNLI-m (Williams et al., 2018), MRPC (Dolan and Brockett, 2005), QNLI (Rajpurkar et al., 2016), RTE (Dagan et al., 2006; Haim et al., 2006; Giampiccolo et al., 2007; Bentivogli et al., 2009) and SST-2 (Socher et al., 2013). Our fine-tuning approach closely follows BERT and similar models: we place a special `</s>` token at the end of the sequence (analogous to the special `<CLS>` token used by BERT) and learn a new classification head on top. We ran a grid search with the learning rate in {1e-0.5, 2e-05, 5e-05, 5e-06} and batch size in {16, 32, 64}, and report the best development accuracy for each model. The rest of hyperparameters follow RoBERTa. For all variants, we tried fine-tuning both with fully unidirectional attention ($r_{\text{bidir}} = 0$) and fully bidirectional attention ($r_{\text{bidir}} = 1$). Refer to Appendix B for more details.

[9]The top 32 candidates contain the correct one in 95.19% of the cases, which is the upper bound accuracy in this setting.

[10]Refer to Artetxe et al. (2021) for a description of the scoring function used for each task and the evaluatio protocol.

[11]We use `<premise>, right? {Yes|No|Also}, <hypothesis>` as our template and report results on the matched development set.

	125M	355M	1.3B	2.7B	6.7B
NxtUni	**22.23**	**17.49**	**14.07**	**12.55**	**11.44**
NxtPre	22.75	18.06	–	–	–
HybUni	23.26	18.19	14.65	13.16	12.03
HybPre	23.91	18.81	15.33	13.92	12.86

Table 3: **Full document perplexity**.

	r_{bidir}	125M	355M	1.3B	2.7B	6.7B
NxtUni	0.00	**19.99**	**15.67**	**12.57**	**11.17**	**10.15**
NxtPre	0.00	20.29	16.05	–	–	–
	0.25	20.25	16.00	–	–	–
	0.50	20.21	15.96	–	–	–
	0.75	20.17	15.92	–	–	–
	1.00	20.16	15.88	–	–	–
HybUni	0.00	20.91	16.30	13.08	11.73	10.70
HybPre	0.00	21.34	16.74	13.60	12.32	11.35
	0.25	21.30	16.69	13.56	12.29	11.33
	0.50	21.26	16.66	13.54	12.26	11.30
	0.75	21.23	16.62	13.51	12.23	11.28
	1.00	21.18	16.56	13.46	12.19	11.24

Table 4: **Suffix perplexity**. We measure perplexity at predicting the last 20% of the tokens in each document conditioned on the first 80%, using $n_{\text{bidir}} = r_{\text{bidir}} \times n_{\text{prefix}}$ for inference, where $n_{\text{prefix}} = 0.8n$ is the length of the prefix we are conditioning on.

4 Results

We visualize our main results in Figure 2, and discuss each setting in more detail next.

4.1 Language modeling

We report full document perplexities in Table 3. NxtUni obtains the best results, followed by HybUni and HybPre, and NxtPre doing slightly better than HybUni at small scale. This is consistent with how close the pre-training objective is to the end task: NxtUni is exclusively trained on next token prediction, HybUni combines it with masking (which is not used here), and HybPre further combines it with a bidirectional attention prefix (which is not used here either). However, it is interesting that scaling up does not reduce the gap between them. This suggests that there is some fundamental interference between these different capabilities,[12] and increasing capacity does not mitigate it.

	r_{bidir}	125M	355M	1.3B	2.7B	6.7B
MskUni	0.00	69.61	73.43	–	–	–
MskBi	1.00	**71.00**	**75.06**	**77.43**	**78.46**	**79.16**
HybUni	0.00	66.86	71.88	75.56	77.19	78.29
HybPre	0.00	67.70	72.25	75.49	76.95	78.02
	0.25	68.02	72.57	75.77	77.25	78.22
	0.50	68.23	72.85	76.05	77.48	78.52
	0.75	68.47	73.13	76.32	77.74	78.70
	1.00	68.71	73.38	76.59	78.00	78.91

Table 5: **Single token infilling accuracy**. We mask a random token in each validation document and measure the accuracy at predicting it, using $n_{\text{bidir}} = r_{\text{bidir}} \times n$ for inference.

Table 4 reports suffix perplexity results, where we predict the last 20% of the tokens in each document conditioned on the rest. Compared to the previous results, NxtPre and HybPre reduce the gap with NxtUni and HybUni, but they still lag behind them. In both cases, we find that the models benefit from using bidirectional attention in the prefix at inference time (i.e., higher values of r_{bidir} yield lower perplexity), but the improvement is relatively small. It is intriguing that NxtUni outperforms NxtPre, when the latter was trained on suffix prediction and can leverage bidirectional attention. We attribute this to the bidirectional prefix reducing the number of tokens of supervision during training.

4.2 Single token infilling

We report infilling results in Table 5. MskBi obtains the best results, which can be explained by its use of bidirectional attention and the fact that it is exclusively trained on masking. Our results suggest that both of these factors play a role, but their impact varies at scale. As for the first factor, we find that bidirectional attention has a larger impact on infilling compared to next token prediction (§4.1), as reflected by MskBi doing substantially better than MskUni. Moreover, we find that this also holds at scale, as reflected by HybPre doing better with larger values of r_{bidir}, while outperforming HybUni. Regarding the second factor, we find that combining masking with next token prediction significantly hurts infilling performance for small models, as reflected by the large gap between MskUni and HybUni. However, we also find

[12] There are various factors that could explain this. Both masking and the bidirectional attention prefix reduce the supervision on next token prediction, and masking further introduces some noise in the original sequence. Moreover, training to use both unidirectional and bidirectional attention and/or context might provide a conflicting signal, although our results later in §4.2 suggest that this does not have a major impact at scale.

		125M	355M	1.3B	2.7B	6.7B
NxtUni	full	69.83	73.13	75.90	77.26	77.98
NxtPre	full	69.40	72.75	–	–	–
MskUni	infill	69.65	73.39	–	–	–
MskBi	infill†	**71.00**	**74.98**	**77.17**	**78.07**	**78.70**
HybUni	full	68.94	72.77	75.43	76.61	77.76
	infill	67.02	71.90	75.38	76.90	77.88
HybPre	full	68.53	72.05	74.75	76.03	76.87
	infill	67.82	72.24	75.35	76.66	77.63
	infill†	68.78	73.35	76.36	77.63	78.47

Table 6: **Single token infilling accuracy, re-ranking the top 32 candidates from 125M MskBi**. † denotes $n_{\text{bidir}} = n$, the rest use $n_{\text{bidir}} = 0$. Refer to §3.2 for more details.

		RE	HS	PI	WG	SC	OB	avg
125M	NxtUni	66.7	32.2	65.3	51.9	64.3	33.0	52.3
	NxtPre	65.8	31.2	64.1	54.1	63.5	35.0	52.3
	HybUni	65.4	30.8	63.1	50.9	63.6	34.4	51.4
	HybPre	64.9	30.5	64.2	51.9	63.0	35.2	51.6
355M	NxtUni	74.8	41.0	69.5	52.2	70.0	38.6	57.7
	NxtPre	74.3	40.0	68.9	52.6	69.2	37.8	57.1
	HybUni	73.9	39.3	68.1	52.3	69.3	37.2	56.7
	HybPre	72.9	37.8	67.6	50.4	68.4	37.4	55.8
1.3B	NxtUni	81.0	52.6	73.8	55.6	74.1	43.6	63.5
	HybUni	80.0	50.3	72.1	53.7	74.1	43.0	62.2
	HybPre	79.4	48.5	71.4	52.9	73.9	38.2	60.7
2.7B	NxtUni	83.8	58.8	75.0	**60.1**	76.6	50.8	67.5
	HybUni	83.1	57.5	73.9	58.0	76.9	45.8	65.9
	HybPre	81.7	54.7	72.4	56.7	75.3	46.6	64.6
6.7B	NxtUni	**85.2**	**63.6**	**76.2**	60.0	**77.6**	**51.6**	**69.0**
	HybUni	84.2	61.7	75.5	59.7	76.8	49.0	67.8
	HybPre	83.9	58.9	73.9	58.7	76.9	50.8	67.2

Table 7: **Zero-shot priming accuracy**. We use $n_{\text{bidir}} = 0$ for inference. `RE`: ReCoRD, `HS`: HellaSwag, `PI`: PIQA, `WG`: WinoGrande, `SC`: StoryCloze, `OB`: OpenBookQA.

		125M	355M	1.3B	2.7B	6.7B
NxtUni	full	44.79	**50.12**	**53.63**	55.09	**55.27**
NxtPre	full	**45.41**	49.15	–	–	–
MskUni	infill	41.69	44.15	–	–	–
MskBi	infill†	41.56	48.34	52.24	**55.59**	53.97
HybUni	full	45.12	47.92	52.59	53.40	54.47
	infill	43.03	44.54	48.13	49.94	51.26
HybPre	full	43.37	47.54	51.53	52.36	54.01
	infill	42.16	44.47	47.36	49.98	50.24
	infill†	42.95	46.57	49.13	51.85	52.41

Table 8: **Zero-shot MNLI accuracy**. † denotes $n_{\text{bidir}} = n$, the rest use $n_{\text{bidir}} = 0$.

the impact of this to vanish at scale, as reflected by the gap between MskBi and HybPre with $r_{\text{bidir}} = 1.0$ becoming smaller for larger models. This also explains why HybPre with $r_{\text{bidir}} = 0.0$ outperforms HybUni for small models, but the trend is reversed as we scale up: the bidirectional prefix in HybPre reduces the relative weight of next token prediction during training, which outweighs the discrepancy with not using bidirectional attention at inference time for small models, but not for larger ones. Interestingly, this is different from the behavior observed for language modeling in §4.1, where scale did not significantly mitigate the negative impact of combining masking and next token prediction during training. We attribute this to masking introducing noise in the original document, as well as reducing the amount of tokens that we train on next token prediction.[13]

Table 6 reports infilling results re-ranking the top 32 candidates from the 125M MskBi model. The best results are still obtained by MskBi, but we find the generative approach described in §3.2 to be competitive, with NxtUni obtaining the second best results at 125M and the third best results for larger models. This suggests that models trained exclusively on next token prediction can also be used for infilling as long as the set of candidates is small, even outperforming hybrid models like HybUni that are trained both on next token prediction and infilling itself. In fact, it is remarkable that NxtUni is only outperformed by models using bidirectional attention which, consistent with our previous results, seems strongly beneficial for infilling. Nevertheless, we also find direct infilling (*infill*) to scale better than generative full sequence scoring (*full*) for both HybUni and HybPre, although this could (partly) be explained by the interference between next token prediction and masking diminishing at scale as discussed previously.

4.3 Zero-shot priming

We report zero-shot priming results in Table 7. We observe the same general trends as in language modeling (§4.1), with NxtUni performing best, followed by HybUni and HybPre. The results are generally consistent across tasks.

Table 8 reports MNLI results, comparing full sequence scoring and direct infilling. Consistent

[13]Note that the reverse is not true: the addition of next token prediction in HybUni does not reduce the amount of supervision on infilling with respect to MskUni, as we use the same value of n_{mask} in both cases.

	r_{bidir}	125M	355M	1.3B	2.7B	6.7B
NxtUni	0.0	83.6	85.8	87.2	88.7	88.6
	1.0	75.9	77.1	79.0	79.2	80.3
NxtPre	0.0	84.2	85.8	–	–	–
	1.0	83.5	86.2	–	–	–
MskUni	0.0	82.7	85.2	–	–	–
	1.0	83.2	85.1	–	–	–
MskBi	0.0	79.6	81.0	81.9	81.6	82.6
	1.0	84.4	**88.0**	**89.6**	**90.8**	**91.0**
HybUni	0.0	83.5	85.9	87.6	88.6	88.8
	1.0	80.8	82.5	84.0	85.0	84.7
HybPre	0.0	83.6	86.1	87.1	88.2	88.2
	1.0	**84.8**	86.7	88.8	89.8	90.3

Table 9: **Average fine-tuning accuracy**.

with the intrinsic evaluation in §4.2, we find full sequence scoring with NxtUni to be competitive with direct infilling with MskBi. In fact, full sequence scoring does even better comparatively, obtaining the best results in all but one of the model sizes. Moreover, it is remarkable that both HybUni and HybPre obtain better results with full sequence scoring compared to direct infilling in all cases. Consistent with our previous results, this suggests that left-to-right language models can be a valid or even superior alternative to masked language models for single-token infilling tasks, as long as one can afford scoring each candidate separately.

4.4 Fine-tuning

We report average fine-tuning results comparing unidirectional and bidirectional attention in Table 9, and full results for the optimal setting for each variant in Table 10.

Our results show that bidirectional attention is helpful for fine-tuning regardless of scale, with fully bidirectional models (MskBi) performing the best, followed by models pre-trained with a bidirectional attention prefix (HybPre, NxtPre), and fully unidirectional models performing the worst (HybUni, NxtUni, MskUni). Interestingly, changing the attention type at fine-tuning time (using unidirectional attention for pre-training and bidirectional attention for fine-tuning, or the other way around) works poorly.

At the same time, we find that the role of bidirectional context is dependant on the type of attention used. When using fully unidirectional attention, bidirectional context has no clear impact, with NxtUni and HybUni performing similarly. In contrast, when using bidirectional attention, bidirectional context seems beneficial, with HybPre performing better than NxtPre at small scale. This suggests that pre-training with bidirectional context is important for the model to learn to make effective use of bidirectional attention.

5 Related work

While it was once common to use random initialization for supervised learning, a series of works showed substantial improvements from pre-training autoregressive models on next token prediction (Dai and Le, 2015; Peters et al., 2018; Howard and Ruder, 2018; Radford et al., 2018). The *pre-train/fine-tune* paradigm was further popularized by BERT (Devlin et al., 2019) and its derivatives like RoBERTa (Liu et al., 2019), which obtained further gains from pre-training bidirectional encoders on masked language modeling. Subsequent work explored masking spans instead of individual tokens, using either bidirectional encoder-only models (Joshi et al., 2020) or encoder-decoder models (Lewis et al., 2020; Raffel et al., 2020). More recently, there has been a reborn interest on scaling left-to-right autoregressive language models with a focus on few-shot priming (Radford et al., 2019; Brown et al., 2020; Rae et al., 2021; Hoffmann et al., 2022; Smith et al., 2022; Chowdhery et al., 2022; Zhang et al., 2022).

While unidirectional and bidirectional models have largely been developed as separate strains of work serving a different purpose, there have also been some attempts to combine the best of both worlds. XLNet (Yang et al., 2019) pre-trained autoregressive models over all permutations of the factorization order, enabling the model to use bidirectional context with strong results on fine-tuning. Similarly, CM3 (Aghajanyan et al., 2022) trained left-to-right autoregressive models, masking some spans that are predicted at the end of the sequence. ERNIE 3.0 (Sun et al., 2021) proposed a modular architecture, combining a shared unidirectional module with either another unidirectional module for NLG or a bidirectional module for NLU. Finally, Raffel et al. (2020) and Wu et al. (2021) explored splitting documents in two halves and predicting the second one conditioned on the first one, using unidirectional attention for the former and bidirectional attention for the latter.

Despite the large body of work on language

		COLA	MNLI	MRPC	QNLI	RTE	SST2	avg
125M	NxtUni	82.4	83.1	82.8	88.8	70.4	93.9	83.6
	NxtPre	81.3	83.3	83.1	90.1	69.3	93.7	83.5
	MskUni	82.6	82.2	81.4	88.4	68.6	93.1	82.7
	MskBi	83.2	84.8	85.5	91.0	68.6	93.5	84.4
	HybUni	82.7	83.1	83.6	89.3	69.3	93.0	83.5
	HybPre	82.5	84.2	85.5	90.9	72.6	93.2	84.8
355M	NxtUni	84.2	85.8	84.1	91.2	74.7	94.8	85.8
	NxtPre	83.8	86.3	86.5	92.0	73.3	95.4	86.2
	MskUni	84.0	84.4	84.6	90.5	73.6	94.2	85.2
	MskBi	85.2	87.7	89.7	92.9	76.2	96.2	88.0
	HybUni	85.4	85.3	85.3	91.0	73.3	94.8	85.9
	HybPre	84.5	86.5	87.3	92.5	74.4	95.2	86.7
1.3B	NxtUni	87.0	87.3	85.3	92.4	75.1	95.9	87.2
	MskBi	85.7	89.1	89.7	93.9	82.3	**96.8**	89.6
	HybUni	86.3	87.0	86.0	92.3	78.0	96.3	87.6
	HybPre	85.1	88.4	90.0	93.6	79.4	96.2	88.8
2.7B	NxtUni	86.0	88.5	85.5	93.0	83.0	96.2	88.7
	MskBi	**87.2**	**89.8**	**91.7**	94.0	85.2	**96.8**	90.8
	HybUni	86.2	88.1	86.8	93.0	80.9	96.7	88.6
	HybPre	86.2	89.4	89.5	94.1	82.7	96.7	89.8
6.7B	NxtUni	86.3	88.5	85.8	93.4	81.2	96.7	88.6
	MskBi	86.7	89.6	90.9	**94.5**	**87.7**	**96.8**	**91.0**
	HybUni	86.7	88.4	87.7	93.4	80.5	96.1	88.8
	HybPre	86.0	89.5	89.5	94.3	85.6	96.7	90.3

Table 10: **Fine-tuning accuracy**. We use $n_{\text{bidir}} = 0$ for NxtUni, MskUni and HybUni, and $n_{\text{bidir}} = n$ for the rest.

model pre-training, there is little work comparing different approaches in a systematic manner. As a notable exception, Raffel et al. (2020) compared various architectures and learning objectives with a focus on fine-tuning. Concurrent to our work, Wang et al. (2022) conduct a comprehensive study with a focus on zero-shot learning and multi-task fine-tuning. In contrast, we focus on the specific role of bidirectionality, and compare models of different sizes.

6 Conclusions

In this work, we study the role of bidirectionality in language model pre-training through a new framework that generalizes previous approaches. Our main findings are as follows:

- **Bidirectional attention** is strongly beneficial for infilling and fine-tuning. In contrast, prefix language models lag behind regular language models on next token prediction, even if they get a small benefit from leveraging bidirectional attention in the prefix. This behavior is consistent at scale.

- Models trained jointly to use unidirectional and **bidirectional context**, like HybUni, lag behind regular language models on next token prediction, and scale does not mitigate this. Such models also lag behind pure masked language models on infilling, but scale does help close this gap as long as they are trained with a bidirectional attention prefix. For fine-tuning, bidirectional context is beneficial when used in conjunction with bidirectional attention, but not when used with unidirectional attention.

- While direct **infilling** requires bidirectional context and benefits from bidirectional attention as discussed above, models using unidirectional context and attention are also competitive in infilling when one can separately score each candidate. For settings where the set of candidates is small (e.g., zero-shot priming for classification), regular language models obtain comparable or even superior results to models pre-trained on infilling.

All in all, our results show that there is not a single configuration that is optimal for all use cases, and this remains generally consistent within the scale range explored in this work. While prior work on scaling has focused on left-to-right autoregressive models, this suggests that there might be other objectives and architectures that are better suited for other applications like fine-tuning. Given the cost of pre-training several models, we would like to explore modular (Sun et al., 2021) or adaptation (Wang et al., 2022) approaches in the future, where one would either have a single model with modular components specialized for different use cases, or efficiently adapt an existing model by changing the parameters in our framework instead of training several models from scratch.

Limitations

Our study focuses on the role of bidirectionality on language model pre-training, and does not explore other factors that might affect model performance. In particular, we mask individual tokens without considering longer spans, and do not explore the impact of the masking rate. In addition, we do not consider sequence-to-sequence models in our study, which combine bidirectional attention in the encoder and unidirectional attention in the decoder. Finally, we train all variants for the same number of tokens, making them comparable in terms of training cost, but resulting in models using a bidirectional attention prefix or a masking objective seeing less tokens of supervision.

References

Armen Aghajanyan, Bernie Huang, Candace Ross, Vladimir Karpukhin, Hu Xu, Naman Goyal, Dmytro Okhonko, Mandar Joshi, Gargi Ghosh, Mike Lewis, and Luke Zettlemoyer. 2022. Cm3: A causal masked multimodal model of the internet.

Mikel Artetxe, Shruti Bhosale, Naman Goyal, Todor Mihaylov, Myle Ott, Sam Shleifer, Xi Victoria Lin, Jingfei Du, Srinivasan Iyer, Ramakanth Pasunuru, Giri Anantharaman, Xian Li, Shuohui Chen, Halil Akin, Mandeep Baines, Louis Martin, Xing Zhou, Punit Singh Koura, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Mona Diab, Zornitsa Kozareva, and Ves Stoyanov. 2021. Efficient large scale language modeling with mixtures of experts.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. In *TAC*.

Yonatan Bisk, Rowan Zellers, Ronan Le bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7432–7439.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2021. On the opportunities and risks of foundation models.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2006. The pascal recognising textual entailment challenge. In *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pages 177–190, Berlin, Heidelberg. Springer Berlin Heidelberg.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and Bill Dolan. 2007. The third PASCAL recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, pages 1–9, Prague. Association for Computational Linguistics.

Aaron Gokaslan and Vanya Cohen. 2019. Openwebtext corpus. `http://web.archive.org/save/http://Skylion007.github.io/OpenWebTextCorpus`.

R Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second pascal recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, volume 7.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training compute-optimal large language models.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2020. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Sebastian Nagel. 2016. Cc-news. `http://web.archive.org/save/http://commoncrawl.org/2016/10/news-dataset-available`.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of* *the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Time Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. Technical report, OpenAI.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. Technical report, OpenAI.

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew Johnson, Blake Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Ed Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2021. Scaling language models: Methods, analysis & insights from training gopher.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8732–8740.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, Elton Zhang, Rewon Child, Reza Yazdani Aminabadi, Julie Bernauer, Xia Song, Mohammad Shoeybi, Yuxiong He, Michael Houston, Saurabh Tiwary, and Bryan Catanzaro. 2022. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, Weixin Liu, Zhihua Wu, Weibao Gong, Jianzhong Liang, Zhizhou Shang, Peng Sun, Wei Liu, Xuan Ouyang, Dianhai Yu, Hao Tian, Hua Wu, and Haifeng Wang. 2021. Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation.

Yi Tay, Mostafa Dehghani, Samira Abnar, Hyung Won Chung, William Fedus, Jinfeng Rao, Sharan Narang, Vinh Q. Tran, Dani Yogatama, and Donald Metzler. 2022a. Scaling laws vs model architectures: How does inductive bias influence scaling?

Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, and Donald Metzler. 2022b. Scale efficiently: Insights from pretraining and finetuning transformers. In *International Conference on Learning Representations*.

Trieu H. Trinh and Quoc V. Le. 2018. A simple method for commonsense reasoning.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. 2022. What language model architecture and pretraining objective work best for zero-shot generalization?

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet:

Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Shaohua Wu, Xudong Zhao, Tong Yu, Rongguo Zhang, Chong Shen, Hongli Liu, Feng Li, Hong Zhu, Jiangang Luo, Liang Xu, and Xuanwei Zhang. 2021. Yuan 1.0: Large-scale pre-trained language model in zero-shot and few-shot learning.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open pre-trained transformer language models.

Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books.

A Proposed framework

Figure 3 provides a step-by-step description of how we define our objective starting from the original sequence.

task	*# of updates*
CoLA	5336
SST-2	20935
MNLI	123873
QNLI	33112
MRPC	2296
RTE	2036

Table 11: Number of fine-tuning updates for each task.

B Fine-tuning settings

For fine-tuning, we did grid search on learning rate $\in \{5e-06, 5e-05, 1e-05, 2e-05\}$ and batch size $\in \{16, 32, 64\}$. For each task, we trained the same numbers of updates for different setups and reported the best numbers across the grid. The details of fine-tuning tasks and numbers of updates can be found in Table 11, which were chosen to follow the original settings from RoBERTa. We used Adam and polynomial decay scheduler for optimization.

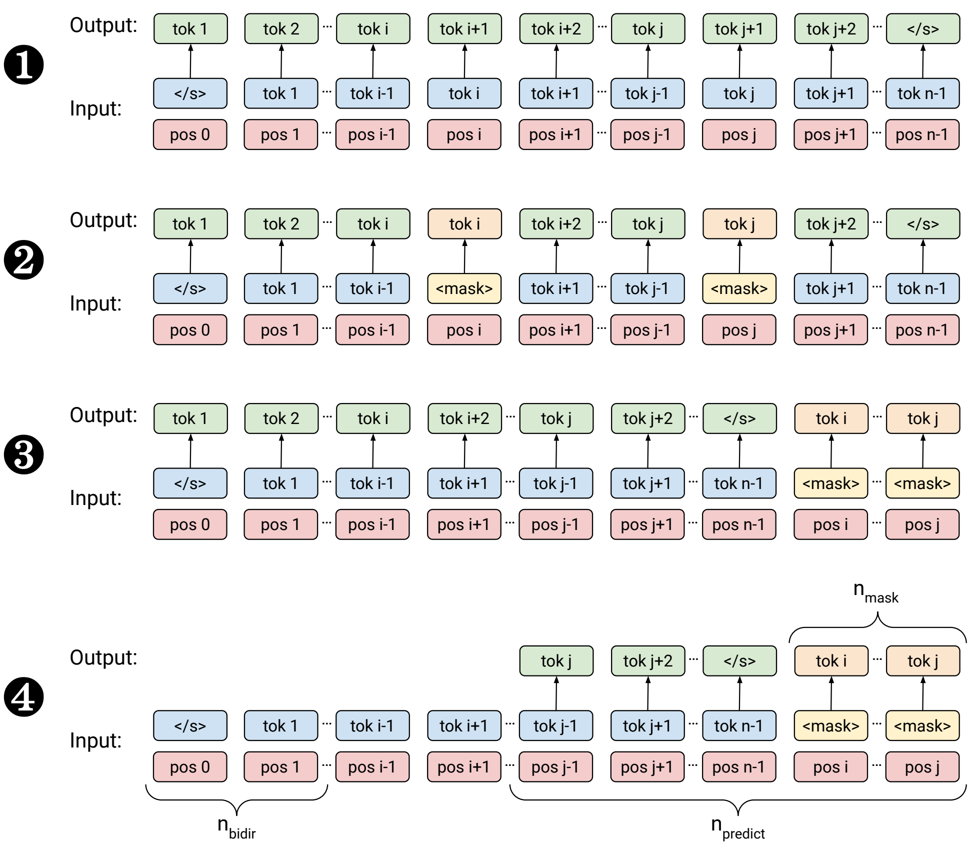

Figure 3: **Proposed framework.** 1) We start with the original sequence in the input, and predict the next token in the output; 2) We choose n_{mask} tokens at random, replace them with the special <mask> token in the input, and predict the masked token (rather than the next token) in the output; 3) We move the masked tokens and their corresponding positional embeddings to the end; 4) We only predict the last n_{predict} tokens, using bidirectional attention for the first n_{bidir} tokens and unidirectional attention for the rest (final objective).