1 Response to Reviewer #1

Interesting to see how well the proposed model would do under such zero-shot setup (i.e. without fine-tuning the model on any particular supervised task). We compared with GPT-2 (345M) on the Winograd Schema Challenge dataset under the zero-shot setup following the GPT-2 paper. Although GPT-2 is trained on much larger corpus, UNILM can achieve slightly better accuracy with comparable number of model parameters.

Model	Number of Parameters	Pretraining Data Size	Accuracy (%)
GPT-2 (345M)	345 million	40GB	62.25
UNILM	340 million	15GB	**64.47**

Table 1: Results on Winograd Schema Challenge under zero-shot setup. GPT-2 accuracy is taken from their paper.

Explain a bit more on what dataset the model was pretrained on, how this dataset was selected, and how the size of the pretraining dataset compares with e.g. ELMo or BERT. We followed the same protocol of pretraining data as BERT. The BERT paper reports that BooksCorpus and Wikipedia contain 0.8B and 2.5B words, respectively. For our processed data, BooksCorpus and Wikipedia contain 0.75B and 2B words, respectively. ELMo was pretrained on Billion Word Benchmark (Chelba et al., 2014), which contains 0.83B tokens.

Explain a bit more on the segment embedding. The implementation is the same as word embedding, i.e., a lookup table is used to store the embeddings of segment indices. We assign a learnable embedding for each segment (such as "Segment 1", and "Segment 2") and feed it to model input, which indicates the segment of input tokens.

Mention pretraining time in L150. Thanks for the suggestion. We will update it in the revised version of the paper.

Interesting to see what happens if beam search decoding is replaced with top-k sampling or nucleus sampling? Top-k sampling and nucleus sampling improve the diversity of unconditioned generation (i.e., sampling text from language models). For conditioned generation (such as summarization, and question generation), beam search achieves better performance in terms of automatic evaluation metrics.

2 Response to Reviewer #2

For the second advantage (L45-48), why single objective LMs will overfit since it is trained on large scale corpus? Using one LM objective makes pre-training biased to a single type of attention pattern. For example, left-to-right LM pretrains how to attend the left context, but the encoders of seq-to-seq downstream tasks need to learn how to utilize both left and right context, which can not be pretrained by only using left-to-right LM objective. We will reword the sentence to avoid confusion.

More experimental results on other generation tasks such as machine translation and response generation. We also evaluate UNILM on a document grounded response generation dataset ("*[ACL-19] Conversing by Reading: Contentful Neural Conversation with On-demand Machine Reading*"). As shown in Table 2, UNILM[1] outperforms the best system (i.e., Team B) in the DSTC7 shared task "*End-to-End Conversation Modeling: Moving beyond Chitchat*".

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

Table 2: Response generation results.

3 Response to Reviewer #3

My only concern point is the initialization of this model from BERT large: what happens if this model is trained from scratch? or what happens if BERT large is continued to be trained on the version of their data? Under the setting of a smaller model size (i.e., BERT-base), we tried both training from scratch and initializing from BERT-base. Both initialization methods on downstream tasks can achieve similar performance, but initializing from BERT-base reduces the number of learning steps. In order to shorten the training time of our large-size model, we initialize it from BERT-large. We will also release a model trained from scratch. We further trained BERT-large using the same hyper-parameters, but the resulted model didn't significantly improve downstream tasks compared to original BERT-large. However, recent work[2] from Facebook (RoBERTa) shows that carefully tuning hyper-parameters, using more training data, and longer training time can improve BERT performance on several language understanding tasks. It is definitely worth further studying how our model will perform with a thorough hyper-parameter tuning and more training data.

[1] Fine-tuning as a sequence-to-sequence model: 20 epochs; batch size=64; masking probability=0.5; maximum length=512. Decoding: beam search with beam size=10; maximum response length=40.

[2] https://arxiv.org/abs/1907.11692