Setting	LAMBADA (acc)	LAMBADA (ppl)	StoryCloze (acc)	HellaSwag (acc)
SOTA	68.0[a]	8.63[b]	**91.8**[c]	**85.6**[d]
GPT-3 Zero-Shot	**76.2**	**3.00**	83.2	78.9
GPT-3 One-Shot	**72.5**	**3.35**	84.7	78.1
GPT-3 Few-Shot	**86.4**	**1.92**	87.7	79.3

Table 3.1: Performance on cloze and completion tasks. GPT-3 significantly improves SOTA on LAMBADA while achieving respectable performance on two difficult completion prediction datasets. [a][Tur20] [b][RWC+19] [c][LDL19] [d][LCH+20]

scraping links over a longer period of time, and first described in [KMH+20], two internet-based books corpora (Books1 and Books2) and English-language Wikipedia (details in the appendix).

2.3 Training Process

As found in [KMH+20, MKAT18], larger models can typically use a larger batch size, but require a smaller learning rate. We measure the gradient noise scale during training and use it to guide our choice of batch size [MKAT18]. Table A.1 shows the parameter settings we used. To train the larger models without running out of memory, we use a mixture of model parallelism within each matrix multiply and model parallelism across the layers of the network. All models were trained on V100 GPU's on part of a high-bandwidth cluster. Details of the training process and hyperparameter settings are described in the appendix.

2.4 Evaluation

For few-shot learning, we evaluate each example in the evaluation set by randomly drawing K examples from that task's training set as conditioning, delimited by 1 or 2 newlines depending on the task. For LAMBADA and Storycloze there is no supervised training set available so we draw conditioning examples from the development set and evaluate on the test set.

For some tasks we use a natural language prompt in addition to (or for $K = 0$, instead of) demonstrations. Similar to [RSR+19] we also sometimes change the formatting of answers. See the appendix for per-task examples.

On tasks with free-form completion, we use beam search with the same parameters as [RSR+19]: a beam width of 4 and a length penalty of $\alpha = 0.6$.

Final results are reported on the test set when publicly available, for each model size and learning setting (zero-, one-, and few-shot). When the test set is private, our model is often too large to fit on the test server, so we report results on the development set.

3 Results

3.1 Language Modeling, Cloze, and Completion Tasks

We test GPT-3's performance on the traditional task of language modeling as well as related tasks. We calculate zero-shot perplexity on the Penn Tree Bank (PTB) [MKM+94] dataset measured in [RWC+19]. We omit the 4 Wikipedia-related tasks and the one-billion word benchmark due to a high fraction of these datasets being contained in our training set. Our largest model sets a new SOTA on PTB by a substantial margin of 15 points.

The LAMBADA dataset [PKL+16] requires the model to predict the last word of a paragraph. Although [BHT+20] suggested scaling language models is yielding diminishing returns on this benchmark, we find that zero-shot GPT-3 achieves a substantive gain of 8% over the previous state-of-the-art. For the few-shot setting, we use a fill-in-the-blank format to encourage the language model to only generate one word (*Alice was friends with Bob. Alice went to visit her friend,* ______. → *Bob*). With this format, GPT-3 achieves an increase of over 18% from the previous state-of-the-art, and