# Pretraining for Conditional Generation with Pseudo Self Attention

Anonymous EMNLP-IJCNLP submission

## Abstract

Large pretrained language representation models have changed the way researchers approach discriminative natural language understanding tasks, leading to the dominance of approaches that finetune a pretrained model. However, such transfer learning approaches have not seen the same success for natural language generation. In this work, we explore transfer learning for conditional generation with large pretrained language models. We propose a simple modification to a pretrained unconditional transformer model to inject arbitrary conditioning into the self attention layer, an approach we call pseudo self attention. Through experiments on four long-form conditional text generation tasks, we show that this technique outperforms strong baselines and other transfer learning approaches, and produces coherent generations.

## 1 Introduction

Large-scale language representation models have recently been shown to dramatically improve the performance of natural language understanding systems on a broad range of tasks (Peters et al., 2018; Devlin et al., 2018; Radford and Salimans, 2018; McCann et al., 2017). These models learn general-purpose contextual representations by pretraining on large corpora of unlabeled text and scaling model size. Optimizing the effectiveness of this approach has been the focus of much study (Houlsby et al., 2019; Wang et al., 2019; Chronopoulou et al., 2019).

Despite this success, the same level of consistent performance improvement has not been demonstrated for conditional language generation tasks. Conditional language generation tasks include both sequence-to-sequence tasks such as translation or summarization, and more general x-to-sequence tasks such as image captioning. Regardless of the modality of the source, however, transfer learning should be able to improve aspects of conditional language generation such as grammaticality and coherence. This is critically important for long-form generation in particular, since for many problems there is adequate supervised data set to learn domain-specific source conditioning, but not enough to learn to produce coherent long-form samples. In this work we therefore study transfer learning applied to the decoder.

Our main observation is that naive approaches to finetune for generation require relearning key parts of the network structure to inject contextual conditioning. In contrast, Radford et al. (2019) observe that simply prepending cleverly chosen phrases can cause language models can give reasonable zero-shot responses. Ideally, through the models self attention, the right phrase could steer it to the right conditional response. Unfortunately, this approach is limited to settings with the same modality and requires using discrete language as the method of control.

We propose to instead learn the correct conditioning to control the models output, in approach we call *pseudo self attention*. The idea is to simply learn an encoder that injects pseudo history into a pretrained self-attention model. Because self attention works with sets of any size, the model can immediately utilize or ignore this history. Finetuning adapts the model to this new input while training an encoder. We compare this approach to two standard variants: a representation learning approach and a contextual attention approach.

Experiments utilize the GPT-2 (Radford et al., 2019) transformer as a pretrained model to compare these approaches. We consider four diverse long-form x-to-sequence generation tasks: class-conditional generation, document summarization, story generation, and image paragraph captioning. Across all tasks, we find that pseudo self attention consistently outperforms other pretraining meth-

ods. We further demonstrate that the approach is data efficient and produces qualitatively more coherent outputs. Code is available at https://removed.for.anonymity.

## 2 Related Work

**Transfer Learning with Language Models** Extending upon the success of pretrained word embeddings (Mikolov et al., 2013), contextual word vectors based on LSTMs first demonstrated strong results across discriminative NLU tasks (McCann et al., 2017; Howard and Ruder, 2018; Peters et al., 2018). Recent work has shown that the transformer (Vaswani et al., 2017) could further improve language representation. BERT (Devlin et al., 2018) trains a transformer via a cloze task and next sentence prediction objectives, leading to state-of-the-art results on many NLU tasks.

GPT and GPT-2 (Radford and Salimans, 2018; Radford et al., 2019) use a similar model in a unidirectional language modeling setting, the latter showing the additional ability to generate impressively coherent unconditional text. As they take the form of standard language models, the GPT models are a natural starting point for pretraining generation models.

**Transfer learning for NLG** NLG tasks have a long history of incorporating unconditional language models with conditional input, especially for machine translation and speech recognition (Bahl et al., 1983; Koehn et al., 2003). These approaches traditionally use the noisy channel model (i.e. Bayes' rule), and $n$-gram models as the language model. Recent adaptations of these ideas include the Neural Noisy Channel (Yu et al., 2017) as well as "fusion" methods (Koehn et al., 2003; Gulcehre et al., 2015; Sriram et al., 2018; Stahlberg et al., 2018) in which the output logits of a language model and a conditional model are combined to calculate the output probabilities. We consider this class of transfer learning as a baseline in a preliminary experiment (see Section 4.1), but focus on alternative "deep" approaches that incorporate the language model weights as an integral part of the model instead of an add-on at the end.

Along these lines, Ramachandran et al. (2017) propose a finetuning-based method for machine translation with LSTMs, in which some of the layers of the LSTM are initialized with pretrained language model weights. As their method is specific to LSTMs, however, it is incompatible with modern transformer architectures. Zhang et al. (2019) use BERT in the encoder and decoder of a summarization model via a unique cloze generative process. They demonstrate strong abstractive summarization performance, but the value of the BERT pretraining relative to other model components is not clear and the cloze process significantly reduces the practicality of the model. Most relevant, Edunov et al. (2019) experiment with a representation-based approach for applying ELMo (Peters et al., 2018) to the source and target sides of a standard seq2seq model separately. Their approach consistently improves performance when applied to the source, but actually *hurts* performance when applied to the decoder. We consider such a representation approach as a baseline in this work.

## 3 Model

We assume that we have a large pretrained language model, $p(\boldsymbol{y}) = p(y_1, \ldots, y_T; \theta)$, that the model is an auto-regressive neural network, and that it is based on self attention to implement conditioning on previous tokens, i.e.

$$\text{SA}(Y) = \text{softmax}((YW_q)(YW_k)^\top)(YW_v)$$

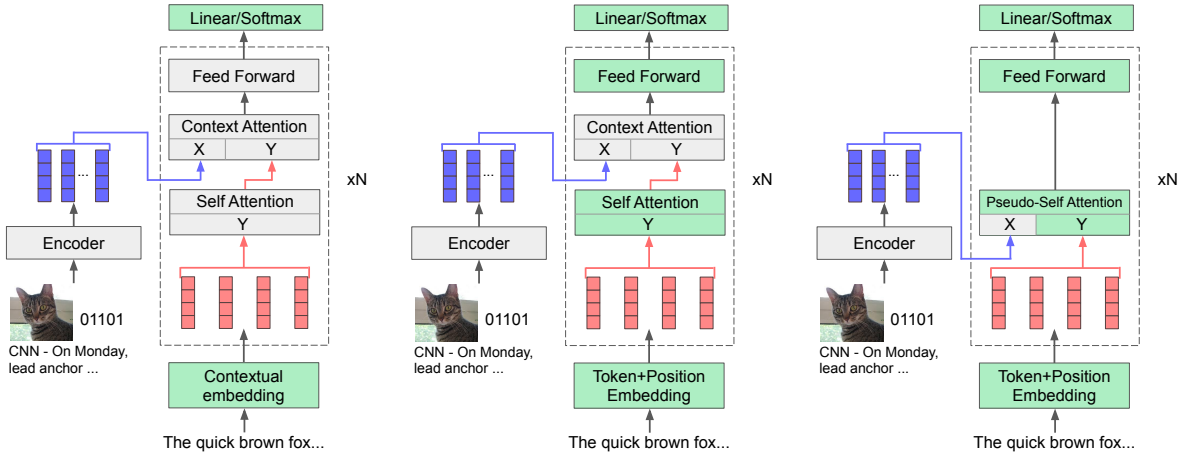
where input $Y \in T \times D$ for hidden dimension $D$, $W_k, W_v, W_q \in D \times D'$ are parameters, representing the key, value, and query projections respectively, and the output is $T \times D'$. [1]

We are interested in using this model to estimate the conditional probability $p(\boldsymbol{y} \mid \boldsymbol{x})$ for an arbitrary input $\boldsymbol{x}$ for which we have a small amount of supervised $(\boldsymbol{x}, \boldsymbol{y})$ pairs. The goal is to learn a model on this new data that best makes use of the pretrained model $p(\boldsymbol{y})$.

One approach to this task is to estimate a randomly initialized $p(\boldsymbol{y} \mid \boldsymbol{x})$ or $p(\boldsymbol{x} \mid \boldsymbol{y})$ model and combine it with the fixed $p(\boldsymbol{y})$. A recent incarnation of this class of model is simple fusion (Stahlberg et al., 2018), in which the output logits of the two models are combined at training and test time. The conditional model's role is to adjust the pretrained LM to fit new data.

A more radical approach is the "zero-shot" model proposed by Radford et al. (2019). Instead of learning a representation for $\boldsymbol{x}$ they note that

[1] In practice many of these units ("heads") are stacked together via concatenation across dimension followed by a final linear projection $W_f \in D \times D$.




(a) Repr-Transformer (b) Context-Attn (c) Pseudo-Self

Figure 1: Pretraining approaches. All methods utilize a problem-specific source encoder, but vary in which parts of the decoder are pretrained and which and randomly initialized. Repr-Transformer trains a new full transformer decoder, Context-Attn trains a new context attention layer, Pseudo-Self attention only modifies part of the self-attention layer. Residual connections and layernorm have been omitted for clarity. Green indicates that parameters are initialized with pretrained weights, gray indicates random initialization. Red vectors indicate the target activations at each layer, Blue vectors indicate the source features at the output of the encoder. xN indicates the section within the dotted lines is stacked N times.

an auto-regressive model, $p(y_t \mid y_{<t})$, is already a conditional model. If $\boldsymbol{x}$ is the same modality as $\boldsymbol{y}$ (e.g. both language), one can condition on $x$ by prepending the source to target: $p(y_t \mid \boldsymbol{x}, y_{<t}) = p(y_t \mid \boldsymbol{x} \odot y_{<t})$.[2] While this does not produce competitive models and is limited in its applicability, it is surprising that it works at all.

Taking inspiration from this approach, we propose learning this contextualization. Our approach, pseudo self attention (**Pseudo-Self**), simply injects learned encoder conditioning directly into the pretrained self-attention of the model. Assume that we have a matrix $X \in S \times D$ representing a size $S$ encoding of $\boldsymbol{x}$, define pseudo self attention as,

$$\text{PSA}(X, Y) = \text{softmax}((YW_q) \begin{bmatrix} XU_k \\ YW_k \end{bmatrix}^\top) \begin{bmatrix} XU_v \\ YW_v \end{bmatrix}$$

where $U_k, U_v \in D \times D'$ are new parameters tasked with projecting encoder outputs into decoder self-attention space. Because attention is inherently variable length, these additional inputs can be injected without changing the module and only act additively on the attention output. The full model is shown in Figure 1c.

[2] This method is most successful when hand-selected task-dependent buffer words are inserted between $\boldsymbol{x}$ and $y_{<t}$ as well such as "tl;dr" for summarization.

**Alternative Approaches** Contextual representation approaches (**Repr-Transformer**, Fig 1a) view the function of the pretrained LM as giving a general-purpose representation of the target text before the source information is introduced. For this method, a standard transformer decoder is used with the target word embeddings replaced by the output representation of the pretrained language model. Preliminary experiments considered both fixing and updating these representations, and found that a fixed weighted-averaging ("ELMo-Style") method performed better, consistent with Edunov et al. (2019).

As an alternative baseline, (**Context-Attn**, fig 1b) considers initializing a standard transformer decoder with the shared weights of a pretrained LM. All possible parameters of the transformer decoder are preinitialized, whereas the newly added context attention weights are randomly initialized. We believe this method places the pretrained parameters closer to the generation signal at finetuning.

**Preliminary Analysis** These alternatives are appealing as they more closely mimic the conditional transformer (Vaswani et al., 2017). However, these new modules may interfere with the pretrained model, whereas the proposed approach only introduces new parameters in the self-attention block.To explore this question, we plot

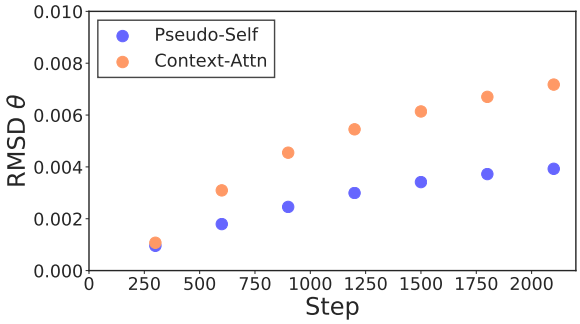



Figure 2: Root median squared deviation between feed forward block parameters and GPT-2 initialization over the course of training, for the Pseudo-Self and Context-Attn models. Squared deviations are taken from all parameters and all layers. The Context-Attn approach requires a larger deviation from the initialization to fit the data. 95% confidence intervals are too small to be seen.

the root median squared deviation of parameters from their original values in the feed-forward layer of our first task (Figure 2). While both start with the same parameters, the Context-Attn parameters change significantly more than Pseudo-Self over training. As the pretrained LM weights encode for generation capability, deviating further from this initialization may lead to worse generation performance.

## 4 Experiments and Results

Experiments consider four generation tasks spanning input modalities, training dataset sizes, and information about the target contained in the source. Tasks are chosen to emphasize long-form targets to probe the natural language generation capabilities of the different models in a conditional setting. Long-form outputs require models to capture general textual phenomena that may be difficult to learn from task-specific training.

For all tasks, GPT-2 is used as the pretrained language model. GPT-2 is a large autoregressive transformer LM trained on 40 GB of non-Wikipedia text (Radford et al., 2019). We use the "small" publicly available version of the model (117M parameters); it has 12 layers, 12 heads per layer, and a model dimension of 768 units.[3] To ensure fair comparisons, all experiments use the same 50k type BPE vocabulary that was used to train GPT-2.

[3]This was the largest publicly available language model at the time of writing, but the method is applicable to larger models as they are released.

| Model | PPL ↓ | Cls Acc ↑ |
|---|---|---|
| Test set | - | 90.1 |
| GPT-2 | 41.21 | - |
| Simple Fusion | 38.31 | 65.1 |
| Transformer | 105.43 | **92.7** |
| Repr-Trans | 39.69 | 72.7 |
| Context-Attn | 40.74 | 88.8 |
| Pseudo-Self | **34.80** | 92.3 |

Table 1: Class-Conditional Generation on IMDb movie reviews. Classification accuracy is measured by a sentiment classifier trained on the IMDb training set.

### 4.1 Class-Conditional Generation

We first consider a baseline task of producing class-conditional samples, e.g. $p(\boldsymbol{y} \mid x = 0)$ and $p(\boldsymbol{y} \mid x = 1)$, from the IMDb sentiment classification dataset (Maas et al., 2011). We set $x$ to be the sentiment bit (positive/negative), and the movie review as the target $\boldsymbol{y}$. We maintain the original IMDb 25k/25k train/test split, with 2.5k reviews of the original train split held out for validation, and truncate reviews to 400 BPE tokens during training. Model quality is evaluated by perplexity, and adherence to the source bit $x$ is evaluated by the sentiment classification accuracy of an external classifier on generated reviews. Reviews are generated via random sampling with a temperature of 0.7. To detect sentiment, we use the fastText external classifier from Joulin et al. (2016) which has an accuracy of 90.1% on the IMDb test set.

Table 1 shows results for all model, as well as unconditional GPT-2 and the results using simple fusion (Stahlberg et al., 2018). GPT-2 model itself already shows a greatly reduced PPL compared to a problem-specific transformer. All pretraining methods further improve perplexity. The pseudo self attention approach significantly outperforms the approaches in terms of class adherence. Despite being initialized as a language model, the approach only sees a decrease of 0.4% classification accuracy compared to the randomly initialized model. In contrast, the Repr-Transformer model sees a decrease in accuracy of 20.0% and the Context-Attn model sees a decrease in accuracy of 3.9%. As a point of comparison, we additionally report the results of simple fusion in Table 1. Compared to Pseudo-Self it gives a worse PPL and extremely poor classification accuracy. Given the weak results, we focus on comparisons

| Model | R1 ↑ / R2 ↑ / RL ↑ | PPL ↓ |
|---|---|---|
| PGenerator+BU | 41.22 / 18.68 / 38.34 | - |
| ELMo+SHDEMB† | 41.56 / 18.94 / **38.47** | - |
| BERT+Two-Stage† | 41.38 / **19.34** / 38.37 | - |
| CopyTransformer | 39.94 / 17.73 / 37.09 | 8.21 |
| Repr-Trans | 37.09 / 13.77 / 33.99 | 13.58 |
| Context-Attn | 40.59 / 18.17 / 37.24 | 6.68 |
| Pseudo-Self | 40.72 / 18.38 / 37.46 | **6.43** |
| Pseudo-Self+BU | **41.62** / 18.66 / 38.46 | **6.43** |

Table 2: CNN/DM summarization results. Literature results above, our models below. † indicates pretraining on the source side. PGenerator+BU from (Gehrmann et al., 2018), ELMo+SHDEMB from (Edunov et al., 2019), BERT+Two-State from (Zhang et al., 2019)

between the deep models for the rest of the paper.

### 4.2 Document Summarization

Next we experiment on a large competitive benchmark for text generation, abstractive document summarization. For these experiments we use the non-anonymized CNN-Daily Mail dataset (Hermann et al., 2015). The dataset is comprised of 280k training examples of document-scale source news articles and corresponding 2-4 sentence target summaries. Summarization is a mature testbed with state-of-the-art models that use task-specific architecture modifications, so transfer learning methods need be able to mesh well with these changes. We use the transformer version of the copy mechanism from (Gehrmann et al., 2018) and employ bottom-up (BU) summarization attention pruning (Gehrmann et al., 2018). For evaluation we report the standard ROUGE-1, ROUGE-2, and ROUGE-L F1 scores. As we are interested in pretraining on the decoder side, in all experiments we start with a randomly initialized encoder (current state-of-the-art models use pretraining for the encoder). Generation is conducted via beam-search with a beam size of 5 with tri-gram blocking, consistent with the literature models (Edunov et al., 2019).

Table 2 shows the performance of the models tested with recent state-of-the-art models for comparison. Compared to the baseline model without pretraining, our approach improves ROUGE-1 by 0.78, ROUGE-2 by 0.65, ROUGE-L by 0.37, and reduced PPL by 20%. The Context-Attn approach nearly matches these results for this task, but the Repr-Transformer approach performs more poorly.

| Model | PPL ↓ | Rank Acc. ↑ |
|---|---|---|
| Transformer | 29.80 | **80.6** |
| Repr-Trans | **21.16** | 77.8 |
| Context-Attn | N/A* | 9.3 |
| Pseudo-Self | 21.21 | 80.3 |

Table 3: WritingPrompt results. "Rank acc." refers to the top-1 prompt ranking accuracy metric described in Section 4.3. Since our experiments use the GPT2 BPE scheme, our PPL numbers are not directly comparable to those reported in (Fan et al., 2018). * The Context-Attn method fails to learn for this task.

Given the strong results of the model, we additionally experiment with simple bottom-up summarization attention pruning approach without pretraining applied at inference time as in (Gehrmann et al., 2018). We achieve a state-of-the-art value for ROUGE-1, demonstrating the ability of the method to be combined with task-specific architecture modifications. Furthermore, because these results only involve pretraining the decoder, the performance can potentially be improved with the addition of ELMo/BERT on the encoder side.

### 4.3 Conditional Story Generation

Conditional story generation with the WritingPrompts dataset (Fan et al., 2018) requires the model to produce an on-topic story given a textual prompt. The dataset is well supervised, containing 300k single sentence writing prompts (the source) and stories (the target). Following the preprocessing of Fan et al. (2018), we truncate the stories to 1000 tokens. Note that due to the length of the stories, the total number of training tokens is on the order of 100 million, resulting in a relatively large in-domain data setting.

To compare models we compute two metrics: perplexity (PPL) and prompt ranking. Perplexity is used as a proxy for generation quality, whereas prompt ranking is used to measure the relevance of the story to the prompt. To calculate prompt ranking, we use the procedure from Fan et al. (2018): For each story in the test set, the likelihood is evaluated under the model for the "true" corresponding prompt and 9 other randomly selected "fake" prompts from the test set. Then, the rank accuracy is the percentage of stories for which the model gave the highest likelihood to the true prompt.

| Model | CIDEr ↑ | B4 ↑ |
|---|---|---|
| LSTM Baseline | 11.1 | 7.3 |
| Krause et al. (2017) | 13.5 | 8.7 |
| Chatterjee et al. (2018) | 20.9 | **9.4** |
| Melas-Kyriazi et al. (2018) | 22.7 | 8.7 |
| Transformer, Repr-Trans | 19.3 | 7.2 |
| Transformer, Context-Attn | 22.6 | 7.6 |
| Transformer, Pseudo-Self | **24.0** | 8.3 |

Table 4: Image paragraph captioning on *Visual Genome*, as measured by CIDEr and BLEU-4 (B4) scores.

Table 3 shows the results. Despite the large dataset size, the Repr-Transfomer and Pseudo-Self approaches still substantially reduce the PPL. That the models are able to improve PPL, despite the 100 million+ target tokens, suggests these models are able effectively make use of the GPT-2 LM. The main approach sees only a 0.3% decrease in prompt ranking accuracy, while the Repr-Transformer approach sees a larger decrease. The Context-Attn model fails to learn in this setting.[4]

### 4.4 Image Paragraph Captioning

Finally, we apply our model to the task of image paragraph captioning on the *Visual Genome* dataset from Krause et al. (2017). As opposed to the standard image captioning task, where captions are single sentences or sentence fragments, the task of image paragraph captioning involves generating an entire paragraph (usually 5-8 sentences) describing a given image.

Recent work in the image captioning literature has argued for a greater focus on paragraph captioning because the descriptive capacity of single-sentence image captions is inherently limited. However, due to the difficulty of producing labeled paragraph captions, existing paragraph captioning datasets are quite small; whereas the MSCOCO (single-sentence captioning) dataset contains around 600,000 image-caption pairs, Visual Genome contains fewer than 20,000 image-paragraph pairs. As a result, models trained from scratch on Visual Genome have been observed to have difficulty learning the structure of language, necessitating the use of heuristics.

To apply pretraining models to this dataset, we extract pre-processed image features from a convolutional neural network encoder. We use the same convolutional encoder as Krause et al. (2017), without the final pooling layer; that is, for each image, the output of the encoder is a tensor of size $(36, 2048)$ extracted from a ResNet. Note that in this experiment, unlike those above, the encoder (CNN) and decoder (finetuned LM) are trained separately rather than end-to-end. Since we are interested in analyzing how to most effectively utilize pretraining for generation, we only compare with approaches using the same loss function (cross-entropy). Recent work shows it is possible to improve paragraph captioning models by incorporating sequence-level (Melas-Kyriazi et al., 2018) and adversarial (Chatterjee and Schwing, 2018) losses, but these loss function improvements are orthogonal to improvements in the underlying model architecture.

Table 4 shows the results on the captioning task, as measured by the widely-used CIDEr and BLEU-4 metrics. We compare our transfer learning method with an LSTM baseline, the Hierarchical-LSTM from Krause et al. (2017), and the current state-of-the-art model. Within the class of pretraining methods, we see similar results as previous experiments. Compared to other cross-entropy based approaches our transfer learning-based model performs well on CIDEr, but slightly worse on BLEU-4.

| Model | PPL ↓ | Cls Acc ↑ |
|---|---|---|
| Pseudo-Self 117M | 34.80 | 92.3 |
| Pseudo-Self 345M | 30.26 | 92.4 |

Table 5: IMDb conditional movie review generation results, comparing the larger 345M parameter GPT2 model to the 117M parameter GPT model.

## 5 Analysis and Discussion

Across all tasks studied, pseudo self attention yields strong results, both in terms of quality and adherence to the source. Other approaches also significantly improve over the non-pretrained baseline, but others are less stable. In this section we discuss other aspects of the system and present qualitative analysis.

### 5.1 Effect of pretrained LM size

There is a continuing trend to larger pretrained LMs. During the preparation of this manuscript,

[4] We have investigated this failure extensively to confirm it is not the result of an error.

| Model | Grammaticality | Non-redundancy | Consistency | Typicality | Combined |
|---|---|---|---|---|---|
| Test set | 71.3 ± 4.3 | 87.2 ± 3.2 | 85.1 ± 3.4 | 74.4 ± 4.1 | 3.18 ± 0.10 |
| Transformer | 55.4 ± 4.7 | 60.5 ± 4.6 | 53.7 ± 4.7 | 39.7 ± 4.6 | 2.09 ± 0.13 |
| Repr-Trans | **62.1** ± 4.4 | **71.0** ± 4.1 | **57.1** ± 4.5 | **43.7** ± 4.5 | **2.34** ± 0.12 |
| Pseudo-Self | **65.2** ± 4.6 | **69.3** ± 4.5 | **61.3** ± 4.7 | **48.4** ± 4.8 | **2.44** ± 0.13 |

Table 6: Human evaluation of story generation quality. Participants were asked specific binary questions concerning the four criteria, the numbers for the four left categories represent percentages of approval. On the right, the methods are rated on a 4-point scale based on the combination of the four criteria. Uncertainties represent a 95% confidence interval, bold indicates statistically significant maxima for each category of the models under consideration.

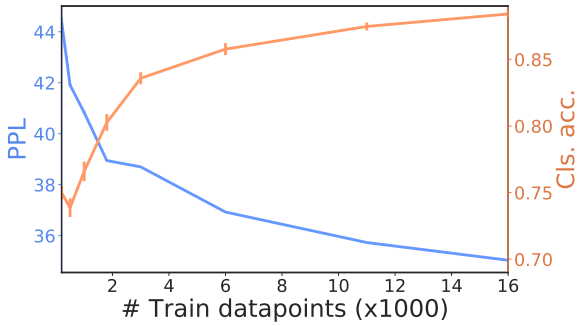



Figure 3: Data efficiency analysis with IMDb. PPL shown in blue (left), classification accuracy shown in orange (right). Error bars show an approximate 95% confidence interval.

a larger version of GPT-2 was made available with 345M parameters, increasing the model dimension to 1028, the number of attention heads to 16, and the number of layers to 24. We retrained our model using this larger LM for class-conditional generation, using the same training hyperparameters and re-tuning the generation temperature (Table 5). The larger model improves PPL by 4.5 points while attaining similarly high classification accuracy. This datapoint suggests that transfer learning effectiveness can continue to improve along with the quality of the pretrained model used.

### 5.2 Low-data supervision

Many of our tasks showed improvements even with medium-to-large training sets. To study the effectiveness of the the approach in low data regimes, we create artificial small datasets by subsampling the IMDb dataset to sizes between 200 and 16k datapoints. We retrain our model using the same hyperparameters and use datasize-dependent early stopping to prevent overfitting. To reduce variance and measure uncertainty we repeat the process 8 times for each dataset size, calculating the PPL and classification accuracy. Results are shown in Figure 3. Note that a non-pretrained model has a PPL of over 1000 when trained on 200 examples. The pretrained model starts with reasonable outputs (44.4 PPL after 200 examples) and increases task accuracy steadily with more data. (See Section 5.4 for representative samples.)

### 5.3 Human evaluation

To assess the quality of generations, we conducted a human evaluation based on the story generation task. Generation uses a temperature of 0.9 and a top-k value of 100. We ask participants on Amazon Mechanical Turk a series of four yes/no questions mapped to desirable linguistic properties outlined in Dang (2006): grammaticality, non-redundancy, consistency, and typicality. 125 stories are evaluated for each model, and each story is evaluated by 5 unique workers. Scores are calculated for each property as the total percent of positive responses. A combined score rates the model overall on a scale from 0-4 based on the equally-weighted combination of the four properties.

The results are shown in Table **??**. In all four categories the Pseudo-Self and Repr-Transformer models show statistically significant performance gains compared to the baseline Transformer model. The Pseudo-Self model achieves a grammaticality score of only 6.1% less than the test set, indicating strong grammaticality, likely a more localized property, is well learned by the pretrained LM and effectively transferred to the conditional models. In contrast, all models score significantly worse than the test data in terms of consistency and typicality. This suggests that these higher level properties, while best transferred in the Pseudo-Self case, still represent a challenge for

| | |
|---|---|
| **Sentiment:** | Positive |
| **No pretraining 1.8k:** | "Set's that I liked this movie. I have seen I remember the original movie is one of the music that it is great movie. I've seen this film and one of the whole movie is like this movie. It is so bad, I watched the top of this movie. i would see the movie was bad, I have seen it. This movie, it's a TV main movie is about the plot, relaxing. I liked this movie takes it is a few times, was awesome. I'm a DVD. The critics in the first time I know it were a lovely plot. You could watch I'm seen the acting, and I don't watch this. I've seen what you really i love the film. The film on the DVD. |
| **No pretraining 22k:** | I was enthralled by the performance of Hartnett at the time of this film. At the time this film was shot, I thought, I was absolutely fantastic in the film, and thought it was a real ride. It is about a family living in a small town San Francisco, where a couple lives in Mexico, where they are working at home with the family. In the end, they are in the family to decide the family will just have their fathers through endless periods, coffee, disillusionment, and finally, can't escape, but there's no story. There are some delightful performances (particularly the two leads) and the actors are well known, I really can't help but feel that the film could have been better as a comedy, with the exception of Ernesto L'Amelie. |
| **Pseudo-Self 1.8k:** | I fell in love with this film in 1985. It's a quintessential short film that explores the everyday lives of the human condition. The main character of the movie is a man named Donald (Husband George). He buys a home and captures a great deal of information about the businessmen who live and work in his neighborhood. This film is a complete departure from the norm or what a typical American would expect from a successful businessman with a short fuse. His stereotypical actors are less than likable, and his best friend is Kelly (Preet Bharara). The film is full of humor, and a great deal of heart and soul. |
| **Pseudo-Self 22k:** | When I saw the preview of this film, I thought it was going to be a horrible movie. I was wrong. The film has some of the funniest and most escapist scenes I've seen in a long time. The acting is superb. The story is decent, but the direction and editing may have been a bit harsh at times. The director and cast achieved a great balance of comedy and drama. I've seen some bad films, but this one is one of the ones I've seen that is really good. I loved the acting and the pace. The two leads were compelling. The only real problem with the film was that I was a bit bored with it. The ending is a bit long, but it's still a funny, good movie. It's efficient. I give it a 7/10. |

Table 7: Example generations from models trained on the movie review generation task. In all cases the indicated sentiment was positive. The number in the left column is the number of training examples (22k is the full dataset).

neural models.

### 5.4 Qualitative examples

Representative samples for the movie review dataset are shown in Table 7. The No-Pretraining model is the transformer from Table 1, and the number in the left column indicates the number of supervised examples in the training dataset. Samples are generated via random sampling with a temperature of 0.75.

Without pretraining, sentences are largely coherent and grammar mistakes are relatively rare. The model makes a number of clear mistakes though such as indicating that the author is in the movie. The Pseudo-Self 22K makes no grammatical mistakes and follows a single train of thought, although it is somewhat more generic.

The distinction between the models is further exaggerated when only 1.8k supervised examples are given. The baseline model trained on only 1.8k datapoints leads to an exceptionally poor generation. In contrast, the Pseudo-Attention model shows significantly improved grammar and sentence structure. Despite a handful of mistakes, the review follows a consistent description of a movie over multiple sentences. Given the poor performance of the baseline model, these properties must have been transferred from the original unconditional LM. These samples were selected to be representative of the broader set for the indicated models.

## 6 Conclusion

In this paper we propose the pseudo self attention approach for improving conditional language generation via transfer learning. Across a set of diverse long-form conditional generation tasks we show that the proposed approach consistently improves performance over strong non-pretraining and pretraining baselines. Furthermore, we demonstrate the data efficiency and qualitative properties of the approach.

This study joins a growing body of work addressing aspects of transfer learning for NLP. Different from previous works based around natural language understanding, however, we find that for generation using pretrained models as contextual features gives less signal than initializing a decoder directly. This is just one example suggesting that the best methods for generation tasks may be different than those for NLU tasks. Future work should investigate these subtle distinctions further.

## References

Lalit R. Bahl, Frederick Jelinek, and Robert L. Mercer. 1983. A Maximum Likelihood Approach to Continuous Speech Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(2):179–190.

Moitreya Chatterjee and Alexander G Schwing. 2018. Diverse and coherent paragraph generation from images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 729–744.

Alexandra Chronopoulou, Christos Baziotis, and Alexandros Potamianos. 2019. An embarrassingly simple approach for transfer learning from pretrained language models. *CoRR*, abs/1902.10547.

Hoa Trang Dang. 2006. Overview of DUC 2006. *Proceedings of HLT-NAACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

Sergey Edunov, Alexei Baevski, and Michael Auli. 2019. Pre-trained Language Model Representations for Language Generation.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical Neural Story Generation. pages 889–898.

Sebastian Gehrmann, Yuntian Deng, and Alexander M. Rush. 2018. Bottom-Up Abstractive Summarization.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On Using Monolingual Corpora in Neural Machine Translation.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching Machines to Read and Comprehend. pages 1–14.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. *CoRR*, abs/1902.00751.

Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of Tricks for Efficient Text Classification. *arXiv preprint arXiv:1607.01759*.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. (June):48–54.

Jonathan Krause, Justin Johnson, Ranjay Krishna, and Li Fei-Fei. 2017. A hierarchical approach for generating descriptive image paragraphs. In *Computer Vision and Patterm Recognition (CVPR)*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 142–150, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in Translation: Contextualized Word Vectors. *31st Conference on Neural Information Processing Systems*.

Luke Melas-Kyriazi, Alexander Rush, and George Han. 2018. Training for diversity in image paragraph captioning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 757–761.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. *Advances in Neural Information Processing Systems*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations.

Alec Radford and Tim Salimans. 2018. GPT: Improving Language Understanding by Generative Pre-Training. *arXiv*, pages 1–12.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners.

Prajit Ramachandran, Peter J. Liu, and Quoc V. Le. 2017. Unsupervised Pretraining for Sequence to Sequence Learning. *Proceedings of EMNLP*.

Anuroop Sriram, Heewoo Jun, Sanjeev Satheesh, and Adam Coates. 2018. Cold fusion: Training Seq2seq models together with language models. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2018-Septe:387–391.

Felix Stahlberg, James Cross, and Veselin Stoyanov. 2018. Simple Fusion: Return of the Language Model. 1:204–211.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *31st Conference on Neural Information Processing Systems*, pages 5998–6008.

Chenguang Wang, Mu Li, and Alexander J. Smola. 2019. Language models with transformers. *CoRR*, abs/1904.09408.

Lei Yu, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Tomas Kocisky. 2017. The Neural Noisy Channel. pages 1–13.

Haoyu Zhang, Yeyun Gong, Yu Yan, Nan Duan, Jianjun Xu, Ji Wang, Ming Gong, and Ming Zhou. 2019. Pretraining-Based Natural Language Generation for Text Summarization.