PALM: Pre-training an Autoencoding&Autoregressive Language Model for Context-conditioned Generation

Bin Bi, Chenliang Li, Chen Wu, Ming Yan,
Wei Wang, Songfang Huang, Fei Huang, Luo Si
Alibaba Group
{b.bi, lcl193798, wuchen.wc, ym119608}@alibaba-inc.com
{hebian.ww, songfang.hsf, f.huang, luo.si}@alibaba-inc.com

Abstract

Self-supervised pre-training, such as BERT (Devlin et al., 2018), MASS (Song et al., 2019) and BART (Lewis et al., 2019), has emerged as a powerful technique for natural language understanding and generation. Existing pre-training techniques employ autoencoding and/or autoregressive objectives to train Transformer-based models by recovering original word tokens from corrupted text with some masked tokens. The training goals of existing techniques are often inconsistent with the goals of many language generation tasks, such as generative question answering and conversational response generation, for producing new text given context.

This work presents PALM with a novel scheme that jointly pre-trains an autoencoding and autoregressive language model on a large unlabeled corpus, specifically designed for generating new text conditioned on context. The new scheme alleviates the mismatch introduced by the existing denoising scheme between pre-training and fine-tuning where generation is more than reconstructing original text. An extensive set of experiments show that PALM achieves new state-of-the-art results on a variety of language generation benchmarks covering generative question answering (Rank 1 on the official MARCO leaderboard), abstractive summarization on CNN/DailyMail as well as Gigaword, question generation on SQuAD, and conversational response generation on Cornell Movie Dialogues.

1 Introduction

Self-supervised pre-training has achieved great success in a wide range of natural language understanding (NLU) tasks (Dai and Le, 2015; Howard and Ruder, 2018; Radford, 2018; Peters et al., 2018; Devlin et al., 2018). Different from language understanding, language generation aims at generating natural language sentences, including tasks like neural machine translation (Bahdanau et al., 2015; Vaswani et al., 2017), abstractive summarization (Rush et al., 2015; See et al., 2017a; Gehrmann et al., 2018), generative question answering (QA) (Tan et al., 2017; Bi et al., 2019), question generation (Zhao et al., 2018) and conversational response generation (Vinyals and Le, 2015). Many of the language generation tasks require the models to read and to comprehend a given document, based on which output text is generated. In this paper, we present PALM, a novel approach to **P**re-training an **A**utoencoding&autoregressive **L**anguage **M**odel for text generation based on reading comprehension of textual context.

Recently, several pre-training methods have been proposed for language generation. GPT (Radford, 2018) and GPT-2 (Radford et al., 2019) use a left-to-right Transformer decoder to generate a text sequence token-by-token, which lacks an encoder to condition generation on context. In contrast, MASS (Song et al., 2019) and BART (Lewis et al., 2019) both employ a Transformer-based encoder-decoder framework, with a bidirectional encoder over corrupted (masked) text and a left-to-right decoder reconstructing the original text. While such denoising pre-training objectives work well for the downstream generation tasks where generated text comes from input but is manipulated, they are less related to the comprehension-based generation tasks asking for instead generating continuations, responses or answers by comprehending input context.

PALM is specifically designed to pre-train a backbone model on a large unlabeled corpus for fine-tuning on the downstream comprehension-based generation tasks, one example of which is generative QA. In generative question answering, QA models are asked to generate an abstractive answer in natural language to a given question by

reading and comprehending a contextual passage. Abstractive answer generation is more than manipulating tokens in the passage. An abstractive answer reflects the understanding of the passage and the question, and can include content out of the passage to be self-contained and well-formed. To address comprehension-based generation like generative QA, PALM uses the pre-training objectives that are closely related to the downstream tasks. Specifically, it differs from existing generative pre-training methods in that PALM goes beyond the solely autoencoding/autoregressive methods and combines the merits of autoencoding and autoregression in a single framework. Moreover, it possesses a mechanism built in pre-training for generating coherent text from given context.

With the new design, PALM surpasses existing language generation methods with or without pre-training – It was trained on 16 NVIDIA V100 GPUs for 3 days in our experiments, and expected to perform even better if trained for longer. PALM gives surprisingly good empirical results on a variety of context-aware generation tasks, including pushing the state-of-the-art Rouge-L on the *MARCO Natural Language Generation* benchmark to 0.498 (Rank 1 on the leaderboard [1]) and on Gigaword summarization to 36.75, as well as establishing the state-of-the-art ROUGE-1 (44.30) and ROUGE-L (41.41) on CNN/Daily Mail.

We make the following major contributions in this paper:

- We propose PALM, a novel approach to pre-training a language model on a large unlabeled text corpus, which is able to comprehend contextual text. The pre-trained model is particularly effective to be fine-tuned for language generation conditioned on context.
- PALM significantly advances the state-of-the-art results on a variety of language generation applications, including generative QA, abstractive summarization, question generation, and conversational response generation. It clearly demonstrates PALM's effectiveness and generalizability in language generation.

2 PALM for Context-conditioned Generation

This section presents the new mechanism and pre-training objectives of PALM for generation conditioned on context. The differences between PALM and prior pre-training approaches are discussed as well.

2.1 Joint Modeling of Autoencoding and Autoregression

We denote $(x, y) \in (\mathcal{X}, \mathcal{Y})$ as a pair of text pieces, where $x = (x_1, x_2, \ldots, x_m)$ is the source text with m tokens, and $y = (y_1, y_2, \ldots, y_n)$ is the target text with n tokens. $\mathcal{X}$ and $\mathcal{Y}$ denote the sets of source text and target text, respectively. PALM uses the standard Transformer encoder-decoder from (Vaswani et al., 2017) as the base architecture, which maximizes the log-likelihood objective: $\mathcal{L}(\theta; (\mathcal{X}, \mathcal{Y})) = \sum_{(x,y)\in(\mathcal{X},\mathcal{Y})} \log P(y|x; \theta)$.

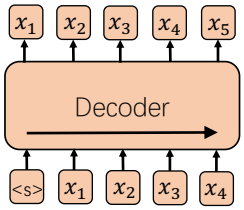
Existing Transformer-based pre-training methods employ either autoencoding or autoregressive objectives for self-supervision. Autoencoding-based pre-training aims to reconstruct the original text from corrupted input. Notable examples are BERT and its variants RoBERTa and ALBERT, where a certain portion of input tokens are replaced by a special symbol `[MASK]`. The models are trained to recover the original tokens from the corrupted version by utilizing bidirectional context. However, these autoencoding methods are not applicable to text generation where bidirectional contexts are not available.

On the other hand, an autoregressive model, such as GPT (Radford, 2018; Radford et al., 2019), is only trained to encode unidirectional context (either forward or backward). Specifically, at each output timestep, a token is sampled from the models predicted distribution and the sample is fed back into the model to produce a prediction for the next output timestep, and so on. While applicable to text generation, the autoregressive methods are not effective at modeling deep bidirectional context. On the contrary, downstream generation tasks often ask a model to condition generation on given textual context. This results in a gap between autoregressive modeling and effective pre-training.

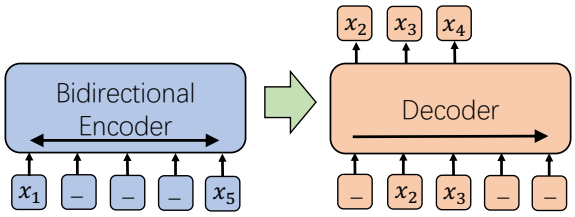
To close the gap, PALM is carefully designed to autoregressively generate a text sequence by comprehending the given context in a bidirectional autoencoding manner. In particular, PALM delegates autoencoding-based comprehension to the encoder in Transformer, and autoregressive generation to the Transformer decoder. The encoder and decoder are jointly pre-trained in two stages:

1. The encoder is first trained as a bidirectional

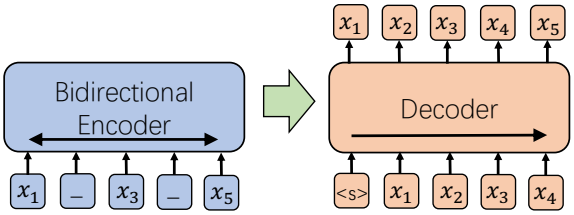
[1] http://www.msmarco.org/leaders.aspx



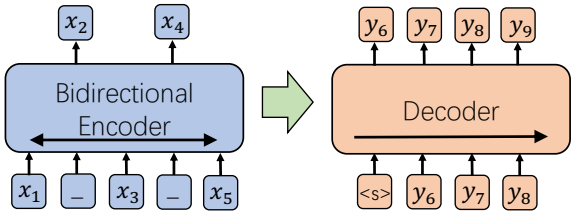
(a) GPT: Tokens are predicted autoregressively, meaning that GPT can be used for generation. However, it lacks an encoder to condition generation on context.



(b) MASS: It is based on the encoder-decoder architecture, but the decoder predicts only the tokens that are masked out in the text input to the encoder.



(c) BART: Rather than masked tokens, the decoder reconstructs the original full sentence from the corrupted input to the encoder. However, it mismatches with most downstream generation which is more than reconstructing original input.



(d) PALM: The encoder predicts masked tokens by encoding context bidirectionally, and the decoder predicts the text segment subsequent to the context. It forces the model to learn to comprehend the context for generating relevant text.

Figure 1: A schematic comparison of PALM with GPT, MASS and BART.

autoencoder to reconstruct the original text from corrupted context in which random tokens are sampled and replaced with `[MASK]` symbols following BERT's practice (Devlin et al., 2018). The training optimizes the cross-entropy reconstruction loss between encoder's output and original context, as Masked Language Modeling (MLM) in BERT. By predicting the actual tokens in context that are masked out, PALM forces the encoder to comprehend the meaning of the unmasked tokens and the full context.

2. The encoder and decoder are then jointly trained to autoregressively generate text output out of the context representations from the encoder. The training maximizes the log-likelihood of the text in ground truth from the decoder's output:

$$\mathcal{L}(\theta) = \sum_{(x,y)\in(\mathcal{X},\mathcal{Y})} \log \prod_{t=1}^{n} P(y_t|y_{<t}, x; \theta), \tag{1}$$

where $\mathcal{X}$ represents the set of context and $\mathcal{Y}$ represents the set of text to be generated. By conditioning the generation on context representations, PALM forces the decoder to rely deeply on the context instead of preceding generated tokens in next token prediction, which facilitates context-sensitive generation.

2.2 Input&Output Representations

In the phase of model pre-training, input and output representations are tailored to minimize the discrepancy between self-supervised pre-training and supervised fine-tuning. In a typical downstream generation task (e.g., abstractive summarization and generative QA), context is given as a rather long passage, and a model is asked to generate a shorter piece of text based on the comprehension of the context.

Given a contiguous text fragment of length L (composed of a few sentences) from an unlabeled corpus, PALM uses the consecutive span of length $80\% \cdot L$ from the beginning of the fragment as context input to the encoder, and uses the remainder of text span of length $20\% \cdot L$ as text output to be generated by the decoder. This representation design mimics the input and output of downstream tasks, with the hypothesis that human-written text is coherent and thus the subsequent text span of length $20\% \cdot L$ captures the comprehension of the preceding context span. In this way, PALM learns to infer the subsequent text content from the preceding content.

The collection of text fragments are constructed from a corpus by following the practice of BERT. In our experiments, we set the maximum length of a fragment to be 500, i.e., $L \leq 500$. Therefore, the context input consists of at most 400 tokens, and the text output consists of at most 100 tokens.

Figure 1 shows a schematic comparison of input&output representations between PALM and the existing pre-training generation methods, GPT, MASS and BART. GPT uses a decoder to predict tokens autoregressively, without an encoder to condition generation on context. MASS and BART are both trained to recover the original tokens that are masked out from corrupted text, where the inputs to the encoder and the decoder come from the same text segment (e.g., the sequence $(x_1, x_2, x_3, x_4, x_5)$ in Figures 1b and 1c). They are also expected to output the tokens from the same text sequence. By contrast, in PALM the encoder and the decoder take two different inputs. The input to the decoder comes from the continuation of the text input to the encoder (e.g., (y_6, y_7, y_8) is subsequent to $(x_1, x_2, x_3, x_4, x_5)$ in the contiguous text segment $(x_1, x_2, x_3, x_4, x_5, y_6, y_7, y_8)$ in Figure 1d). In addition to the continuation predicted by the decoder, PALM produces an extra output from the encoder, which contains the predicted tokens masked in the input (e.g., x_2 and x_4 in Figure 1d). The output predictions from the encoder and the decoder are used for training in the two stages, respectively.

2.3 Copying Tokens from Context

In a human-written document, subsequent text often refers back to entities and tokens present earlier in the preceding text. Therefore, it would increase coherence of text generated in downstream to incorporate the copy mechanism into pre-training on an unlabeled corpus. This allows the model to learn from pre-training when and how to copy tokens in generating text, and the knowledge is transferred to downstream fine-tuning.

PALM incorporates the copy mechanism by plugging in the pointer-generator network (See et al., 2017b; Nishida et al., 2019) on top of the decoder in Transformer. Figure 2 illustrates the pointer-generator network, which allows every token to be either generated from a vocabulary or copied from context in generating text.

Extended vocabulary distribution. Let the extended vocabulary, V, be the union of words in the vocabulary and all tokens present in context. $P^v(y_t)$ then denotes the probability distribution of the t-th word token, y_t, over the extended vocabulary, defined as:

$$P^v(y_t) = \text{softmax}(W^e(W^v s_t + b^v)), \quad (2)$$

where s_t denotes the output representation of t-th token from the decoder. The output embedding

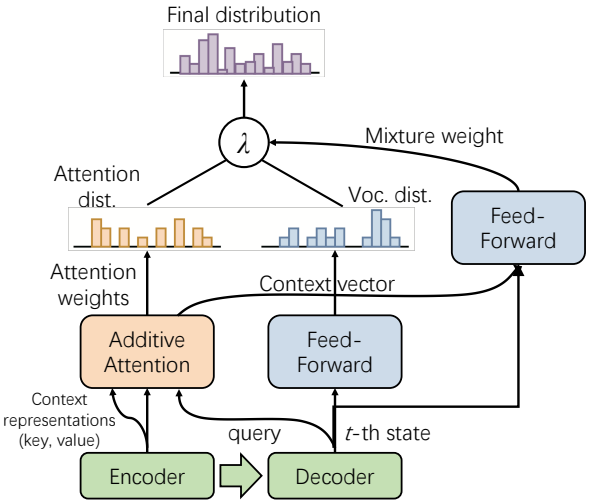


Figure 2: The pointer-generator network on top of the decoder in Transformer. For each decoding step t, mixture weights λ for the probability of generating tokens from the vocabulary and copying tokens from context are calculated. The two distributions are summed in a weighted manner to obtain the final distribution.

W^e is tied with the corresponding part of the input embedding (Inan et al., 2017), and W^v and b^v are learnable parameters.

Copy distribution. PALM uses an additional attention layer for the copy distribution on top of the decoder. In the course of generation, the layer takes s_t as the query, and outputs α_t as the attention weights and z_t^c as the context vector:

$$e_{tl}^c = w^{c\top} \tanh(W^m h_l^c + W^s s_t + b^c), \quad (3)$$

$$\alpha_t^c = \text{softmax}(e_t^c), \quad (4)$$

$$z_t^c = \sum_{l=1}^{m} \alpha_{tl}^c h_l^c, \quad (5)$$

where h_l^c is the representation of l-th token in context from the encoder. w^c, b^c, W^m and W^s are learnable parameters. As a result, $P^c(y_t)$ is the copy distribution over the extended vocabulary, defined as:

$$P^c(y_t) = \sum_{l:x_l=y_t} \alpha_{tl}^c. \quad (6)$$

Final distribution. The final probability of generating y_t is defined as a mixture of the extended vocabulary distribution and the copy distribution:

$$P(y_t) = \lambda P^v(y_t) + (1-\lambda) P^c(y_t), \quad (7)$$

$$\lambda = \text{sigmoid}(w^z z_t^c + w^s s_t + b^m), \quad (8)$$

where w^z, w^s and b^m are learnable parameters.

The parameters in pointer-generator learned in pre-training are all kept and passed downstream for fine-tuning on labeled data.

Example 1	
Input	A classic Aston Martin once owned by Spartacus star Peter Ustinov is set to fetch more than £1 million at auction - twice what it fetched four years ago. The actor bought the Aston Martin DB4 Cabriolet in 1962, shortly after winning a Best Supporting Actor Oscar for his role as Batiatus in Spartacus. It was one of the most luxurious cars of its day, costing £4,000 and was delivered to him at a Swiss hotel at a time when the average house price in Britain was just £2,500.
PALM	The Aston Martin DB4 Cabriolet was bought by Peter Ustinov for £4000 and was expected to fetch for £2.5 million for auction. The car was sold for £1.2 million finally.
MASS	peter ustinov's UNK auctioned for more than $1 million.
Example 2	
Input	Cape Verde's 2-0 win over Portugal was the most eye-catching international result of the week. So, who are Cape Verde and why has this tiny island off the west coast of Africa suddenly become an international football force? Where are the Cape Verde Islands? Cape Verde is a group of islands 400 miles from Senegal off the west coast of Africa. Its population is around 500,000 and boasts both beautiful beaches and striking volcanic landscapes, making it a haven for tourism.
PALM	Cape Verde is a small island off the west coast of Africa with a population of around 500,000 and boasts both beautiful beaches and striking volcanic landscapes, making it a haven for tourism. Cape Verde is home to the Cape Verde Islands National Park with a number of islands.
MASS	tiny african island nation cape verde has beautiful beaches.

Table 1: Example generated continuations of the text input to PALM and MASS.

3 Experiments

In this section, we present the experimental setup and results of PALM pre-training on a large unlabeled corpus and fine-tuning on a variety of language generation tasks, including generative QA, abstractive summarization, question generation, and conversational response generation.

3.1 Pre-training Configuration

Experimental Setup. PALM is based on the Transformer which consists of a 12-layer encoder and a 12-layer decoder with 768 embedding/hidden size, 3072 feed-forward filter size and 12 attention heads. We have also trained a larger model, referred to as PALM $_{\text{LARGE}}$, to compare with the baseline models of the same size. PALM $_{\text{LARGE}}$ has an encoder of 24 layers and a decoder of 6 layers, with 1024 embedding/hidden size and 16 attention heads. The parameters of PALM's encoder are initialized by the pre-trained RoBERTa model[2] which was trained with the Masked LM objective, removing Next Sentence Prediction from BERT.

PALM is trained with a dropout rate of 0.1 on all layers and attention weights, and a GELU activation function (Hendrycks and Gimpel, 2016) used as GPT. The learning rate is set to 1e-5, with linear warmup over the first 10k steps and linear decay. The pre-training procedure runs on 16 NVIDIA V100 GPU cards for 800K steps, with each mini-batch containing 64 sequences of maximum length 500 tokens.

Pre-training Dataset. We use documents of English Wikipedia and BookCorpus (Zhu et al., 2015) as our pre-training corpus, and perform WordPiece tokenization as BERT (Devlin et al., 2018). The documents are split into sentences. Different from BERT, we use multiple consecutive sentences up to 400 tokens as the source text input to the encoder, and use the subsequent consecutive sentences up to 100 tokens as the target text to the decoder. The pre-training dataset $(\mathcal{X}, \mathcal{Y})$ is constructed from the documents by a sliding window with the stride of one sentence, resulting in 50M (x, y) pre-training pairs.

3.2 Unsupervised Pre-training

To understand the performance of PALM pre-training, we compare generation quality of the pre-trained models of PALM and MASS [3]. Specifically, we feed a few sentences from a news article to both pre-trained models, and the models generate a continuation of the input sentences by beam search with a beam of size 5. The news articles from CNN [4] are used as input text to eliminate the possibility of the text present in the models' pre-training corpus, i.e., Wikipedia and BookCorpus.

The overall perplexity of PALM is 17.22, which is much better than MASS's perplexity of 170.32, indicating PALM's better language modeling. Table 1 illustrates a couple of example continuations generated by PALM and MASS. In both examples, PALM generates fluent and grammatical English, while MASS outputs a short sentence that is much

[2] https://github.com/pytorch/fairseq

[3] https://modelrelease.blob.core.windows.net/mass/mass_summarization_1024.pth

[4] https://drive.google.com/uc?export=download&id=0BwmD_VLjROrfTHk4NFg2SndKcjQ

less relevant to input text, since the MASS model was trained on individual sentences. In the first example, it is interesting to observe that in addition to summarizing the input content, PALM is able to make a non-trivial inference of the expected auction price and the final selling price of the car (might not be factually accurate though). An inference is also made by PALM in the second example in addition to summarization, although the Cape Verde Islands National Park does not really exist.

These examples demonstrate that PALM pre-training has learned to infer and to reason from the input text. Although in the pre-training phase the generated content may not be factually accurate in the absence of rich context, the capability of inference can be transferred downstream by fine-tuning on specific generation tasks.

3.3 Fine-tuning on Generative QA

We also experiment with fine-tuning PALM on several downstream generation tasks. The MARCO benchmark (Nguyen et al., 2016) released by Microsoft is a good fit for evaluating generative QA models. In the MARCO dataset, the questions are user queries issued to the Bing search engine and the contextual passages are from real web documents. The data has been split into a training set (153,725 QA pairs), a dev set (12,467 QA pairs) and a test set (101,092 questions with unpublished answers). To evaluate the generative capability, we focus on the *Q&A + Natural Language Generation* task, the goal of which is to provide the best answer available in natural language that could be used by a smart device / digital assistant.

The answers are human-generated and not necessarily sub-spans of the contextual passages, so we use the ROUGE-L (Lin, 2004) metric for our evaluation to measure the quality of generated answers against the ground truth.

We fine-tune the pre-trained PALM on the MARCO training set for 10 epochs. We set the batch size to 64, the learning rate to 1e-5, and the maximum input length to 512. The other hyper-parameters are kept the same as pre-training. In fine-tuning PALM, the encoder takes as input x a contextual passage concatenated with a question at the end, and the decoder takes an answer as input y. During decoding, we use beam search with a beam of size 5.

Table 2 presents the answer generation results on the test set obtained from the official MARCO leaderboard. PALM achieves the 1st place on the leaderboard, outperforming all competing methods in generation quality. Note that PALM pre-trains a single model, while some of the top-performing methods are ensemble models, such as Masque, on the leaderboard. Crucially, the superiority of PALM-single over Masque-ensemble with pre-trained ELMo (Peters et al., 2018) and BERT-based methods clearly demonstrates the effectiveness and generalizability of PALM over the other pre-training approaches in language modeling.

Method	**Rouge-L**
ConZNet (Indurthi et al., 2018)	0.421
Reader-Writer	0.439
KIGN-QA	0.441
SNET+CES2S	0.450
Communicating BERT	0.483
VNET (Wang et al., 2018)	0.484
Selector NLGEN	0.487
BERT+Multi-Pointer	0.495
Masque (Nishida et al., 2019)	0.496
PALM	**0.498**

Table 2: Test results of answer generation on the official MARCO leaderboard as of December 9, 2019.

3.4 Fine-tuning on Summarization

Text summarization produces a concise and fluent summary conveying the key information in the input (e.g., a news article). We focus on abstractive summarization, a generation task where the summary is not constrained to reusing the phrases or sentences in the input text. We conduct experiments on both the CNN/DailyMail dataset (Hermann et al., 2015) and the Gigaword dataset (Graff and Cieri, 2003). The CNN/DailyMail dataset contains 93K news articles from CNN and 220K articles from Daily Mail, while the Gigaword dataset consists of a total of 3.8M article-title pairs. We take the articles as the input to the encoder and the summary for the decoder. We adopt the same optimization hyperparameters from generative QA fine-tuning for the summarization task. The F1 scores of Rouge-1, Rouge-2 and Rouge-L are reported on the test set of both datasets for evaluation.

Table 3 shows the results of abstractive summarization on the CNN/DailyMail test set and the Gigaword test set. PALM achieves better performance than all strong summarization mod-

	CNN/DailyMail			Gigaword		
	RG-1	**RG-2**	**RG-L**	**RG-1**	**RG-2**	**RG-L**
BERTSUMABS (Liu and Lapata, 2019)	41.72	19.39	38.76	-	-	-
MASS (Song et al., 2019)	42.12	19.50	39.01	38.13	19.81	35.62
$\text{UniLM}_{\text{LARGE}}$ (Dong et al., 2019)	43.33	20.21	40.51	38.45	19.45	35.75
T5_{LARGE} (Raffel et al., 2019)	42.50	20.68	39.75	-	-	-
$\text{BART}_{\text{LARGE}}$ (Lewis et al., 2019)	44.16	21.28	40.90	-	-	-
PEGASUS (Zhang et al., 2019)	44.17	**21.47**	41.11	39.12	19.86	36.24
$\text{ERNIE-GEN}_{\text{LARGE}}$ (Xiao et al., 2020)	44.02	21.17	41.26	39.25	20.25	36.53
PALM	42.71	19.97	39.71	38.75	19.79	35.98
$\textbf{PALM}_{\text{LARGE}}$	**44.30**	21.12	**41.41**	**39.45**	**20.37**	**36.75**

Table 3: Results of abstractive summarization on the CNN/DailyMail test set and the Gigaword test set. RG is short for ROUGE

els with pre-training recently proposed, including UniLM (Dong et al., 2019), T5 (Raffel et al., 2019), BART (Lewis et al., 2019), PEGASUS (Zhang et al., 2019) and ERNIE-GEN (Xiao et al., 2020). By consistently outperforming the pre-training methods, PALM confirms its effectiveness in leveraging unsupervision signals for language generation.

3.5 Fine-tuning on Question Generation

We conduct experiments for the answer-aware question generation task. Given an input passage and an answer span, question generation aims to generate a question that leads to the answer. Following the practice in (Zhao et al., 2018; Dong et al., 2019), we use the SQuAD 1.1 (Rajpurkar et al., 2016) dataset, and the BLEU-4, METEOR and ROUGE-L metrics for evaluation.

As shown in Table 4, PALM outperforms all previous question generation systems and achieves a new state-of-the-art result on BLEU-4 and ROUGE-L for question generation on the SQuAD 1.1 dataset.

Method	**BLEU-4**	**MTR**	**RG-L**
CorefNQG[a]	15.16	19.12	-
MP-GSN[b]	16.38	20.25	44.48
UNILM[c]	22.88	24.94	51.80
ERNIE [d]	22.28	25.13	50.58
$\text{ERNIE-GEN}_{\text{LARGE}}$ [d]	24.03	**26.31**	52.36
PALM	22.78	25.02	50.96
$\textbf{PALM}_{\text{LARGE}}$	**24.11**	25.85	**52.38**

Table 4: Question generation results on the SQuAD dataset. MTR is short for METEOR and RG is short for ROUGE. [a] (Du and Cardie, 2018); [b] (Zhao et al., 2018); [c] (Dong et al., 2019); [d] (Xiao et al., 2020).

3.6 Fine-tuning on Response Generation

Conversational response generation aims to produce a flexible response to a conversation (Vinyals and Le, 2015). Following MASS, we conduct experiments on the Cornell Movie Dialog corpus[5] (Danescu-Niculescu-Mizil and Lee, 2011) that contains 140K conversation pairs, and use the training/test splits provided by the dataset. The same training hyperparameters from generative QA fine-tuning are adopted on the response generation task. We report the results in perplexity following (Vinyals and Le, 2015) (lower is better).

We compare PALM with the competing methods including the baseline trained on the data pairs available and the pre-trained BERT+LM and MASS. Following MASS, we train every model on 10K pairs randomly sampled and all 110K training pairs. As shown in Table 5, PALM significantly performs better than all the competitors by a large margin on both the 10K and 110K data, demonstrating its capability in generating responses to context thanks to its new pre-training objectives.

3.7 Ablation Studies

We conduct ablation studies to assess the individual contribution of every component in PALM. Table 6 reports the results of full PALM and its ablations on the CNN/Daily Mail summarization dataset.

We evaluate how much the pointer-generator network contributes to generation quality by removing it from PALM pre-training. This ablation results in a drop from 39.71 to 39.49 on Rouge-L, demonstrating the role of the pointer-generator in generative modeling. Given the slight drop, one may choose to exclude it from the full model for

[5] https://github.com/suriyadeepan/datasets/tree/master/seq2seq/cornell_movie_corpus

Method	Perplexity (10K Data)	Perplexity (110K Data)
Baseline	82.39	26.38
BERT+LM	80.11	24.84
MASS	74.32	23.52
PALM	**45.43**	**21.98**

Table 5: Results of conversational response generation in terms of perplexity on Cornell Movie Dialog corpus (lower is better).

Ablation	RG-1	RG-2	RG-L
PALM	**42.71**	**19.97**	**39.71**
✗ pointer-generator	42.54	19.86	39.49
✗ autoencoding	41.78	19.32	38.81
✗ autoregression	41.89	19.48	38.92
✗ pre-training	40.32	17.78	37.12

Table 6: Ablation tests of PALM on the CNN/Daily Mail summarization dataset.

training efficiency. In our experiments, the pointer-generator is used in every generation task for optimal generation performance.

To study the effect of the pre-trained encoder and decoder in PALM, we ablate autoencoding and autoregression by randomly initializing the weights of the encoder and the decoder, respectively. The autoencoding and autoregression components both prove to be critical with significant drops on the three Rouge metrics after the ablation. Finally, we study the significance of full PALM pre-training. Over 6.5% of performance degradation resulted from ablating pre-training clearly demonstrates the power of PALM in leveraging an unlabeled corpus for downstream generation.

4 Related Work

ELMo (Peters et al., 2018) is an early prominent pre-training method based on bidirectional LSTMs. It concatenates left-only and right-only representations, but does not pre-train interactions between these features. GPT (Radford, 2018), GPT-2 (Radford et al., 2019) and GPT-3 (Brown et al., 2020) are proposed to base language modeling on the Transformer architecture, and use only the Transformer decoder for pre-training. Edunov *et al.* (Edunov et al., 2019) examine different strategies (e.g., ELMo) to add contextualized embeddings to sequence-to-sequence models, and observe the most improvement by adding the learned embeddings to the encoder.

BERT (Devlin et al., 2018) introduces Masked Language Modelling, which allows pre-training to learn interactions between left and right context words. Recent work has shown that very strong performance can be achieved by training for longer (Liu et al., 2019), by tying parameters across layers (Lan et al., 2019), and by masking spans instead of words (Joshi et al., 2019). However, BERT does not make predictions autoregressively, so it is not effective for generation tasks.

UniLMs (Dong et al., 2019; Hangbo et al., 2020) fine-tune BERT with an ensemble of masks, some of which use only leftward context, allowing UniLMs to be used for generation tasks. A difference between UniLMs and PALM is that UniLMs are not fully autoregressive in the pre-training process. In contrast, PALM reduces the mismatch between pre-training and context-conditioned generation tasks by forcing the decoder to predict the continuation of text input on an unlabeled corpus.

MASS (Song et al., 2019) and BART (Lewis et al., 2019) are the two pre-training methods most similar to PALM. In MASS, an input sequence with a masked span of tokens is mapped to a sequence consisting of the missing tokens, whereas BART is trained to reconstruct the original text from corrupted input with some masked tokens. The difference in input & output representations between PALM and MASS & BART is detailed in Section 2.2.

5 Conclusions

In this work, we propose PALM, a novel approach to pre-training an autoencoding and autoregressive language model on a large unlabeled corpus, designed to be fine-tuned on downstream generation conditioned on context. It is built upon an extension of the Transformer encoder-decoder, and jointly pre-trains the encoder and the decoder in an autoencoding denoising stage followed by an autoregressive generation stage.

PALM significantly advances the state-of-the-art results on a variety of context-conditioned generation applications, including generative QA (Rank 1 on the MARCO leaderboard), abstractive summarization, question generation, and conversational response generation. It has been shown in prior work (Liu et al., 2019) that training for more steps over a larger corpus can potentially improve the performance of pre-training. Our future work will explore the potential of training PALM for longer on much more unlabeled text data.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Bin Bi, Chen Wu, Ming Yan, Wei Wang, Jiangnan Xia, and Chenliang Li. 2019. Incorporating external knowledge into machine reading for generative question answering.

T. Brown, B. Mann, Nick Ryder, Melanie Subbiah, J. Kaplan, P. Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, G. Krüger, Tom Henighan, R. Child, Aditya Ramesh, D. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, E. Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, J. Clark, Christopher Berner, Sam McCandlish, A. Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3079–3087. Curran Associates, Inc.

Cristian Danescu-Niculescu-Mizil and Lillian Lee. 2011. Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs. In *Proceedings of the 2nd Workshop on Cognitive Modeling and Computational Linguistics*, pages 76–87, Portland, Oregon, USA. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems 32*, pages 13042–13054. Curran Associates, Inc.

Xinya Du and Claire Cardie. 2018. Harvesting Paragraph-Level Question-Answer Pairs from Wikipedia. *arXiv e-prints*, page arXiv:1805.05942.

Sergey Edunov, Alexei Baevski, and Michael Auli. 2019. Pre-trained language model representations for language generation. 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 4052–4059, Minneapolis, Minnesota. Association for Computational Linguistics.

Sebastian Gehrmann, Yuntian Deng, and Alexander M Rush. 2018. Bottom-up abstractive summarization. *arXiv preprint arXiv:1808.10792*.

David Graff and Christopher Cieri. 2003. English gigaword. In *Linguistic Data Consortium*, Philadelphia.

Bao Hangbo, Dong Li, Wei Furu, Wang Wenhui, Yang Nan, Liu Xiaodong, Wang Yu, Piao Songhao, Gao Jianfeng, Zhou Ming, and Hon Hsiao-Wuen. 2020. Unilmv2: Pseudo-masked language models for unified language model pre-training. In *Proceedings of International Conference on Machine Learning (ICML)*.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *CoRR*, abs/1506.03340.

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.

Hakan Inan, Khashayar Khosravi, and Richard Socher. 2017. Tying word vectors and word classifiers: A loss framework for language modeling. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Sathish Reddy Indurthi, Seunghak Yu, Seohyun Back, and Heriberto Cuayáhuitl. 2018. Cut to the chase: A context zoom-in network for reading comprehension. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 570–575, Brussels, Belgium. Association for Computational Linguistics.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. 2019. SpanBERT: Improving pre-training by representing and predicting spans. *arXiv preprint arXiv:1907.10529*.

Zhen-Zhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *ArXiv*, abs/1909.11942.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. *arXiv e-prints*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Proceedings of the ACL Workshop: Text Summarization Braches Out 2004*, page 10.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3730–3740, Hong Kong, China. 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. *CoRR*, abs/1907.11692.

Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. 2016. MS MARCO: A human generated machine reading comprehension dataset. In *Proceedings of the Workshop on Cognitive Computation: Integrating neural and symbolic approaches 2016 colocated with the 30th Annual Conference on Neural Information Processing Systems (NIPS)*.

Kyosuke Nishida, Itsumi Saito, Kosuke Nishida, Kazutoshi Shinoda, Atsushi Otsuka, Hisako Asano, and Junji Tomita. 2019. Multi-style generative reading comprehension. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2273–2284, Florence, Italy. Association for Computational Linguistics.

Matthew 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. 2018. Improving language understanding by generative pre-training.

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

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv e-prints*, page arXiv:1606.05250.

Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, Lisbon, Portugal. Association for Computational Linguistics.

Abigail See, Peter J Liu, and Christopher D Manning. 2017a. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017b. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. MASS: masked sequence to sequence pre-training for language generation. *CoRR*, abs/1905.02450.

Chuanqi Tan, Furu Wei, Nan Yang, Weifeng Lv, and Ming Zhou. 2017. S-net: From answer extraction to answer generation for machine reading comprehension. *CoRR*, abs/1706.04815.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Oriol Vinyals and Quoc Le. 2015. A neural conversational model. *ICML Deep Learning Workshop, 2015*.

Yizhong Wang, Kai Liu, Jing Liu, Wei He, Yajuan Lyu, Hua Wu, Sujian Li, and Haifeng Wang. 2018. Multi-passage machine reading comprehension with cross-passage answer verification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 1918–1927.

Dongling Xiao, Han Zhang, Yukun Li, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE-GEN: An Enhanced Multi-Flow Pre-training and Fine-tuning Framework for Natural Language Generation. *arXiv e-prints*, page arXiv:2001.11314.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2019. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. *arXiv e-prints*, page arXiv:1912.08777.

Yao Zhao, Xiaochuan Ni, Yuanyuan Ding, and Qifa Ke. 2018. Paragraph-level neural question generation with maxout pointer and gated self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3901–3910, Brussels, Belgium. Association for Computational Linguistics.

Yukun Zhu, Ryan Kiros, Richard S. 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. *CoRR*, abs/1506.06724.