Bidirectional Language Models Are Also Few-shot Learners

Anonymous ACL submission

Abstract

Large language models such as GPT-3 (Brown et al., 2020) can perform certain tasks without undergoing fine-tuning after seeing only a few labeled examples. An arbitrary task can be reformulated as a natural language prompt, and a language model can be asked to generate the completion, indirectly performing the task in a paradigm known as prompt-based learning. To date, emergent prompt-based learning capabilities have mainly been demonstrated for unidirectional language models. Bidirectional language models pre-trained on denoising objectives such as masked language modeling produce stronger learned representations. Prompting bidirectional models has long been desired, but their pre-training objectives have made them incompatible with the prompting paradigm. We present SAP (Sequential Autoregressive Prompting), a technique that enables the prompting of bidirectional models. Utilizing the machine translation task as a case study, we prompt the bidirectional mT5 (Xue et al., 2021) model with SAP and demonstrate its few-shot and zero-shot translations outperform the few-shot translations of unidirectional models like GPT-3 and XGLM (Lin et al., 2021) with approximately 50% fewer parameters. We further show SAP extends its effectiveness to the tasks of question answering and summarization. For the first time, our results demonstrate prompt-based learning is an emergent property of a broader class of language models, rather than a property of only unidirectional models.

1 Introduction

Recent work on GPT-2 (Radford et al., 2019) and GPT-3 (Brown et al., 2020) have shown that large language models possess few-shot learning capabilities and zero-shot performance, despite only being pre-trained with a self-supervised causal language modeling objective (which is to predict the next token).

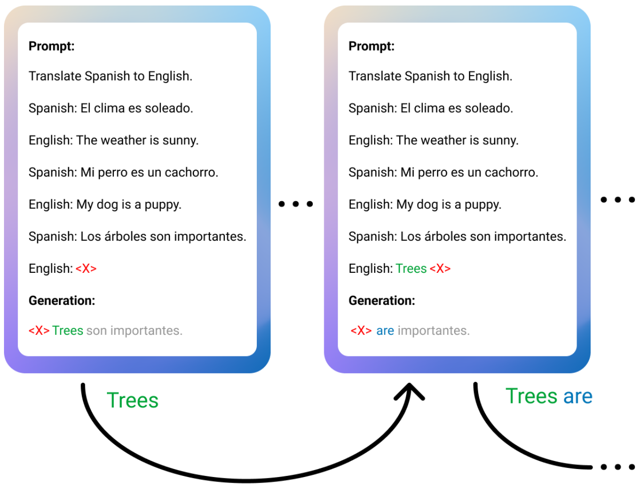

Figure 1: A visualization of our SAP technique extracting high-quality translations from mT5. In the zero-shot setting, the examples used in the prompt are synthetic examples retrieved in a fully unsupervised manner.

An arbitrary task can be converted into a natural language task specification, often called a *prompt*. Prompting a task in this way makes its format similar to the language modeling objective used to pre-train large language models. In the zero-shot setting, this prompt contains just the task, whereas in the few-shot setting, the prompt contains both the task and several example demonstrations. When a language model is tasked to generate text to complete this prompt, it can perform the task in the process. The paradigm of reframing all tasks as text generation is known as *prompt-based learning*. In the few-shot setting, the learning that occurs from examples provided in a given prompt (the context) is known as *in-context learning* (Liu et al., 2021).

Emergent prompt-based learning capabilities have mainly been demonstrated for unidirectional language models. Bidirectional language models have stronger learned representations (Devlin et al., 2019; Conneau et al., 2020; Raffel et al., 2020);

however, they have not been able to broadly demonstrate the same few-shot learning capabilities or zero-shot performance due to the incompatibility bidirectional denoising pre-training objectives have with the prompting paradigm and instead typically require fine-tuning or prompt-tuning (Lester et al., 2021). Bidirectional models are not able to generate long, fluent completions to prompts since they are usually only trained to output short spans of text, mask in-fills, during pre-training. We discuss this more in-depth in Section 2.1.

Today, language model architects are faced with a difficult choice between unidirectional or bidirectional models. The authors of GPT-3 lay out this design dilemma in Brown et al. (2020):

> "GPT-3 has several structural and algorithmic limitations ... as a result our experiments do not include any bidirectional architectures or other training objectives such as denoising ... our design decision comes at the cost of potentially worse performance on tasks which empirically benefit from bidirectionality ... making a bidirectional model at the scale of GPT-3, and/or trying to make bidirectional models work with few- or zero-shot learning, is a promising direction for future research, and could help achieve the 'best of both worlds'."

In this paper, we directly address this dilemma. We contribute a new technique, SAP (**S**equential **A**utoregressive **P**rompting), that enables bidirectional language models to take advantage of prompting and allows them to perform at the level of unidirectional models in few- or zero-shot learning without fine-tuning. SAP iteratively prompts bidirectional models, concatenating previous generations back into the prompt, to produce longer generations from models that were only pre-trained to output short, mask-infill spans.

Using the machine translation task as an in-depth case study, we empirically demonstrate mT5 (Xue et al., 2021), a bidirectional language model, used with SAP outperforms its unidirectional counterparts, GPT-3 and XGLM (Brown et al., 2020; Lin et al., 2021), while utilizing approximately 50% fewer parameters. We find both the few-shot and zero-shot translations produced by SAP with mT5 can outperform the few-shot translations produced by GPT-3 and XGLM. We then examine SAP's effectiveness on other tasks such as question answering and summarization, demonstrating that bidirectional models can be prompted for tasks beyond machine translation.

Our work hints at the possibility of more efficient and performant few-shot learners through pre-trained language models that incorporate bidirectionality. We discuss this impact and outline future research directions to this end in Section 6. In summary, our key contributions are:

1. We introduce SAP, a technique that enables bidirectional language models to work with few-shot and zero-shot in-context learning at a level that exceeds unidirectional models, addressing a long-standing challenge in language model design. Our results demonstrate prompt-based learning is an emergent property of a broader class of language models, rather than only unidirectional models.

2. We perform an in-depth study of the effectiveness of a bidirectional language model, mT5, with SAP on the machine translation task. We find, despite using approximately 50% fewer parameters than GPT-3 and XGLM, SAP with mT5 exceeds in average performance over 14 language pairs and achieves significant improved zero-shot translation performance on many low-resource language pairs.

3. We propose a range of improvements—filtering, prompt ensembling, and English-centric bootstrapping—to the unsupervised machine translation procedure outlined by Han et al. (2021) to better adapt the bootstrapping process for unsupervised low-resource machine translation.

4. We assess SAP's performance on the tasks of question answering and summarization, and find the technique enables the few-shot learning capabilities of bidirectional models beyond machine translation.

2 Related Work

2.1 Unidirectional and Bidirectional Language Models

Transformer-based language models (Vaswani et al., 2017) can be broadly categorized into bidirectional and unidirectional models. Bidirectional models are models that use a denoising pre-training objective (such as masked language modeling), allowing them to utilize *bidirectional* context when learning language representations. Unidirectional language models are models with a causal—or a left-to-right—language modeling objective (such as next token prediction), restricting them to be *unidirectional* when learning representations (Liu et al., 2021).

The T5 family of models, such as T5 v1.1 and mT5, are bidirectional, while GPT-style models, such as GPT-2, GPT-3, and XGLM are unidirectional. BERT-style models are bidirectional, but they cannot be easily utilized for prompting since they are encoder-only (Wang and Cho, 2019). Usually, but not always, bidirectional models are paired with an encoder-decoder architecture, while unidirectional models are paired with a decoder-only architecture (Devlin et al., 2019; Raffel et al., 2020; Xue et al., 2021; Radford et al., 2019; Brown et al., 2020; Lin et al., 2021; Wang et al., 2022).

Devlin et al. (2019) and Raffel et al. (2020) have both shown that after transfer learning, bidirectional denoising pre-training objectives such as BERT's masked language modeling and T5's random span corruption outperform causal language modeling on downstream tasks. Brown et al. (2020) concedes this to be a potential source of weakness for the GPT-3 model on certain tasks where bidirectionality is important.

Despite the advantages of denoising objectives, prompting ability has been shown to be weaker on bidirectional language models, disqualifying them when few-shot in-context learning and zero-shot prompting is desired. Lester et al. (2021) explains this may be because:

> "...a T5 model pre-trained exclusively on span corruption, such as T5.1.1, has never seen truly natural input text (free of sentinel tokens), nor has it ever been asked to predict truly natural targets"

In other words: when pre-trained on their denoising objectives, language models like T5 that utilize bidirectionality are only conditioned to output a single token or short spans of tokens (the in-fill of the mask) rather than full and complete sentences; this inhibits their ability to generate arbitrary-length natural responses to a variety of prompts.

Despite the stronger learned representations of bidirectional models, their shortcomings in prompt-based learning motivate Brown et al. (2020) and Lin et al. (2021) to explicitly choose unidirectional models over bidirectional models for GPT-3 and XGLM.

2.2 Prompting Bidirectional Language Models

Unlike prior approaches to backfill prompt-based learning capabilities into bidirectional models, our technique, SAP, neither requires fine-tuning, weight updates, nor supervised instruction-tuning datasets. It demonstrates for the first time that bidirectional language models have innate few-shot learning capabilities.

Cloze-style prompts Schick and Schütze (2021a) and Schick and Schütze (2021b) find that bidirectional models such as RoBERTa and ALBERT (Liu et al., 2019; Lan et al., 2019) can be prompted with cloze-style phrases. They propose a few-shot training paradigm called PET where the model's predicted mask in-fill, called a "verbalizer," is used to label fine-tuning examples for the model. These verbalizers are only a single word or a few words, e.g. "yes", "no", "amazing", "worse". These works primarily demonstrate effectiveness on classification tasks such as sentiment classification, rather than more challenging generation tasks such as machine translation or question answering. While their paradigm has success in bringing few-shot learning to bidirectional models, it requires fine-tuning, a major limitation contrasted with the in-context learning ability of undirectional models such as GPT-3.

LM-adaptation Lester et al. (2021) finds some success with prompting the T5 v1.1 models after continued pre-training on the unidirectional prefix-LM objective described in Raffel et al. (2020). The resulting model, T5 v1.1 LM-adapted (T5+LM), is described as a late-stage adaptation to a unidirectional objective. Adaptation requires performing weight updates and given that representations learned by the original denoising objective have been shown to be superior (Raffel et al., 2020), we hypothesize that such an adaptation could degrade the quality of the learned representations.

Prompt-tuning Lester et al. (2021) and Li and Liang (2021) find by fine-tuning only a portion of the parameters in an otherwise frozen pre-trained bidirectional language model, a "soft prompt" can be discovered through backpropagation. Soft prompts are prompts discovered in the embedding space of the model and are not grounded in natural language. The prompt-tuning approach requires training the learned prompt embeddings and benefits from initialization from LM-adaptation. The nature of soft prompts lacking grounding in natural language makes their use and flexibility limited, a stark difference from the prompting capabilities of unidirectional models. (Liu et al., 2021)

Instruction-tuning Language models can be fine-tuned on a supervised dataset consisting of natural language prompts and their respective target completions (Wei et al., 2021; Sanh et al., 2022; Ouyang et al., 2022; Min et al., 2021). This "instruction-tuning" technique allows these models to improve performance on instruction following and therefore exhibit few-shot and zero-shot capabilities through prompting. The T0 model in particular is an instruction-tuned version of the T5+LM model (Lester et al., 2021) and is able to augment the bidirectional T5 v1.1 model with prompting capabilities. While instruction-tuning likely bolsters the instruction following performance of a model, we hypothesize that by instruction-tuning, the T0 model is to some degree surfacing the innate prompting ability that the bidirectional model already has. We provide evidence towards this hypothesis by demonstrating that bidirectional models can be prompted without instruction-tuning.

2.3 Unsupervised Machine Translation through Prompting

GPT-2 (Radford et al., 2019) and GPT-3 (Brown et al., 2020) have shown it is possible to perform few-shot machine translation and unsupervised zero-shot machine translation using large language models, prompting, and in-context learning. The XGLM model (Lin et al., 2021) trains a similar architecture to GPT-3 on a diverse multilingual corpus, resulting in XGLM performing better on few-shot, low-resource machine translation. Han et al. (2021) introduce a bootstrapping technique to further improve the unsupervised zero-shot performance on machine translation.

3 Few-shot Machine Translation

To motivate our method for enabling few-shot in-context learning in bidirectional language models, we first focus on applying $\text{mT5}_{3.7\text{B}}$ (mT5-XL) (Xue et al., 2021) to the machine translation task as an in-depth case study since the task benefits greatly from bidirectionality (Conneau et al., 2020; Lin et al., 2021). mT5 is a bidirectional model trained on random span corruption, a variant of masked language modeling. We demonstrate that with SAP, mT5 can perform few-shot machine translation using prompting and in-context examples with no fine-tuning. We formulate a prompt format that utilizes its random span masking scheme to complete the translation task:

> Translate Spanish to English.
> Spanish: El clima es soleado.</s>
> English: The weather is sunny.</s>
> Spanish: Mi perro es un cachorro.</s>
> English: My dog is a puppy.</s>
> Spanish: Los árboles son importantes.</s>
> English: <X>

3.1 Sequential Autoregressive Prompting (SAP) Technique

By requiring mT5 to in-fill <X>, we are effectively asking it to translate the requested source language sentence. However, due to the limitations of the denoising pre-training objective on prompting (described in Section 2.1), we observe mT5 often outputs a partial translation of the beginning of the source sentence, rather than the full translation. To overcome this, we prompt mT5 T times until the model generates a stop token </s>, resulting in a longer translation. At each time step of iteration, we keep the first word generated (using the space character as delimiter) and concatenate it into the last line of the prompt to use in the next time step. This iterative prompting enables us to extract longer generations. Formally, we denote the generation at each time step t as G_t. We denote the first word generated at each time step as F_t where $F_t = \texttt{SPLIT}(G_t, \texttt{" "})\texttt{[0]}$. We update the prompt at each time step P_t to include the cumulative generation from all previous time steps concatenated in the last line of the prompt. The prompt used at each time step P_t is as follows:

> Translate Spanish to English.
> Spanish: El clima es soleado.</s>
> English: The weather is sunny.</s>
> Spanish: Mi perro es un cachorro.</s>
> English: My dog is a puppy.</s>
> Spanish: Los árboles son importantes.</s>
> English: $\texttt{CONCAT}(F_0, \ldots, F_{t-1})$ <X>

In Table 1, we also consider concatenating the entire generation G_t instead of just the first word of the generation F_t, but find that it produces significantly inferior results as low-quality tokens are generated after the first word. By conditioning the model to generate the next word in the translation based on previous words generated, this technique resembles autoregression. mT5 is already autoregressive, but it is autoregressive only at the decoder level. Adding previously generated words back into the prompt allows them to pass through the encoder layers as well. For this reason, we call this technique SAP (**S**equential **A**utoregressive **P**rompting).

To provide a signal to stop generation, we add a custom stop token at the end of each example

	English-Russian	Russian-English
Prompting (mT5$_{3.7B}$)		
Using the full generation from the first time step only – G_0	1.9	5.6
Sequential **P**rompting (mT5$_{3.7B}$ + SP)		
Concatenating the full generation at each time step – CONCAT($G_0, \ldots, G_t$)	9.3	17.9
Sequential **A**utoregressive **P**rompting (mT5$_{3.7B}$ + SAP)		
Concatenating the first word of the generation at each time step – CONCAT($F_0, \ldots, F_t$)	**20.1**	**26.9**

Table 1: Few-shot (2-shot) machine translation results on FLORES-101 devtest (spBLEU) using mT5$_{3.7B}$ as described in Section 3. In this experiment, we ablate simply prompting the model once and taking the full generation G_0 with concatenating the full generation G_t or just the first word of the generation F_t at each time step to the prompt in the next time step over two language pairs, English-Russian and Russian-English.

in the prompt. We stop prompting after the model generates a stop token[1]. We also implement a basic post-processing step to automatically detect and remove repetitive generations or cycles.

The overall process is graphically depicted, with stop tokens omitted, in Figure 1.

3.2 Results

Following Lin et al. (2021), we evaluate our technique on 14 languages from the FLORES-101 dataset (Goyal et al., 2021) that span high-resource and low-resource languages[2]. We evaluate SentencePiece BLEU (spBLEU) (Goyal et al., 2021) in every direction leading to an evaluation over 182 language pairs in total. Abbreviated results can be found in Table 2, and the matrix of full results can be found in Appendix A. Examples generations can be found in Appendix G.

On an average spBLEU score over all 182 pairs, we find that our model matches the performance of the unidirectional XGLM and GPT-3 models (+0.1 spBLEU)—with approximately 50% fewer parameters and 16x fewer examples. Notably, our technique significantly improves performance on language pairs with at least one low-resource language, but trails slightly on high-resource pairs.

4 Unsupervised Zero-shot Machine Translation

We now perform fully unsupervised zero-shot machine translation with SAP and mT5 to extend our in-depth case study on the machine translation task. We ultimately will replace the examples in the few-shot prompt with synthetic parallel examples. These synthetic parallel examples are bootstrapped in a completely unsupervised fashion using a zero-shot translation prompt with no examples. The zero-shot prompt format looks like:

> Translate Spanish to English.
> Spanish: Los árboles son importantes.</s>
> English: <X>

We adapt the bootstrap process of Han et al. (2021) to retrieve these synthetic parallel examples. The process, as depicted in Figure 2, consists of three steps:

> **Step 1 (sampling)**: Generate synthetic parallel examples using a zero-shot translation prompt (with no examples) to translate sentences from a monolingual source language corpus.

> **Step 2 (filtering)**: Filter out low-quality synthetic examples to keep only high-quality synthetic examples using an unsupervised scoring technique (discussed in Section 4.1).

> **Step 3 (self-amplification)**: Translate any source language sentence desired using these synthetic parallel examples in the few-shot prompt.

We iteratively run multiple rounds of this bootstrap by repeating step 2 and step 3 to form a better few-shot prompt. The few-shot prompt after self-amplification is used to translate more source language sentences. These are then filtered using the scoring technique used in step 2 and so on. We run four bootstrapping rounds in our experiments and sample 100 source language sentences from the training dataset in each round of the bootstrap. Note that the target language parallel sentences

[1] We repurpose the 100th sentinel token from the mT5 vocabulary as our stop token.

[2] High-resource Languages: en, de, fr, ca, fi, ru, bg, zh
Low-resource Languages: ko, ar, sw, hi, my, ta

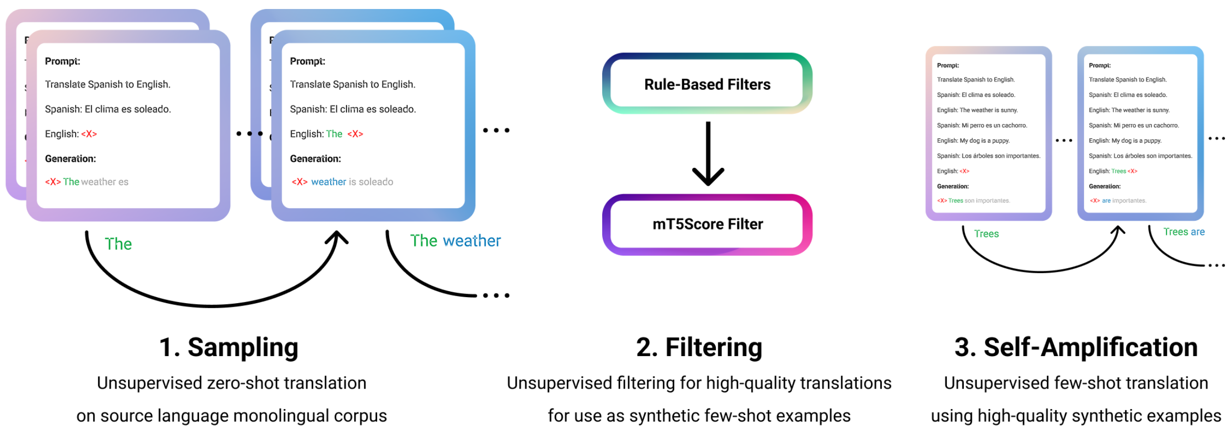

Figure 2: A visualization of the bootstrapping process described in Section 4.

from the training dataset are not used in this zero-shot setting; following Han et al. (2021), only the source language sentences are used.

4.1 Filtering Down to High-quality Translations

The filtering step of the bootstrap requires an unsupervised scoring method for assessing the quality of translations. We first utilize `langdetect`[3], a language identifier we use as a simple rule-based filter, to ensure the generated text is in the desired target language. We then score the remaining generated translations against their corresponding original sentence in the source language. For this unsupervised multilingual similarity metric, we utilize the BERTScore (Zhang et al., 2019) algorithm with mT5_{300M} (mT5-small)[4], dubbing it "mT5Score". We ablate the use of mT5Score as a filter in Appendix C.

We take the top two synthetic parallel examples with the highest mT5Score in the filtering step and use those as synthetic few-shot examples in the prompt in the self-amplification step.

4.2 Translating with an Ensemble of Prompts

Because the two examples used in the prompt can greatly affect the quality of the generated translations, some prompts containing low-quality synthetic examples may cause poor translations for certain sentences. To combat this and reduce variation in performance, we keep the top N synthetic examples instead of two synthetic examples. We use these to form $\frac{N}{2}$ different few-shot prompts with two synthetic parallel examples each. Each sentence in the test set is then translated with these $\frac{N}{2}$ different prompts to produce $\frac{N}{2}$ translations. The best translation of the $\frac{N}{2}$ translations is chosen in a fully unsupervised manner with mT5Score, as done in the filtering step of the bootstrap.

We find this ensembling technique helps make unsupervised zero-shot performance competitive with few-shot performance. Ablation experiments can be found in Appendix D. Unless otherwise stated, we use a 4 prompt ensemble in this paper: $\frac{N}{2} = 4$. In sum, we sample and zero-shot translate 100 sentences from a monolingual corpus, keep the top eight synthetic parallel examples scored by mT5Score, and use them to form four few-shot prompts with two synthetic examples in each prompt.

4.3 English-centric Bootstrapping

While Han et al. (2021) only performed a bootstrap on English-French and French-English pairs, we perform bootstrapping on some language pairs which may contain at least one low-resource language or non-English language.

It has been found that multilingual language models perform best in English due to the imbalance of languages in the pre-training corpus (Lin et al., 2021). Therefore, when running the bootstrap on various language pairs, we modify the bootstrap to favor generating English, or pivot through English when neither the source nor target language is English. Ablation experiments can be found in Appendix E.

We outline examples of our modified English-centric bootstrapping process for various language pairs below:

- **Example 1** (Russian-English): No change.

[3] https://pypi.org/project/langdetect/

[4] The BERTScore Python library provided by Zhang et al. (2019) directly supports using mT5 instead of BERT.

		HR → HR	LR → HR	HR → LR	LR → LR	All
Number of Language Pairs		56	48	48	30	182
Supervised		25.5	15.4	12.6	8.2	16.6
GPT-$3_{6.7B}$	(32-shot)	14.0	2.1	0.4	0.1	5.0
XGLM$_{7.5B}$	(32-shot)	**20.5**	11.6	7.9	4.4	12.2
mT$5_{3.7B}$ + SAP	(2-shot)	18.2	12.2	9.2	6.4	12.3
mT$5_{3.7B}$ + SAP	(zero-shot)	19.3	**13.1**	**10.0**	**7.3**	**13.2**

Table 2: Abbreviated few-shot and unsupervised zero-shot machine translation results on FLORES-101 devtest (spBLEU). The matrix of full results can be found in Appendix A. Results are average spBLEU scores over subsets of the 182 language pairs (`src → tgt`) where "LR" is a low-resource language and "HR" is a high-resource language. "All" represents the average spBLEU score over the full set of 182 language pairs. Bold denotes best of GPT-3, XGLM, and mT5. spBLEU computed using the implementation from Goyal et al. (2021).

- **Example 2** (English-Russian): In step 1, generate Russian-English synthetic examples using a Russian monolingual corpus. Then, reverse the examples to get English-Russian synthetic examples.

- **Example 3** (Russian-Chinese): In step 1, for the first three rounds of the bootstrap, generate Russian-English synthetic examples and Chinese-English synthetic examples using Russian and Chinese monolingual corpora. On the fourth and final round, use an English monolingual corpus along with the reversed previous synthetic examples to produce English-Russian and English-Chinese synthetic examples. Since the same English sentences are used to produce both sets, we can align these to form synthetic Russian-Chinese examples. In step 2, we use the harmonic mean of the two mT5Scores to filter examples.

4.4 Results

We report results using the few-shot evaluation method described in Section 3.2. Abbreviated results can be found in Table 2 and the matrix of full results can be found in Appendix A.

In this unsupervised setting, we find our zero-shot results exceed our 2-shot results; furthermore, they significantly exceed the performance of XGLM and GPT-3 on an average spBLEU score over all 182 pairs (+1.0 spBLEU). Again, we note strong performance on language pairs that contain one or more low-resource languages.

Intuitively, we can explain the zero-shot performance surpassing the few-shot performance through our use of prompt ensembling in the zero-shot setting. As prompt ensembling utilizes four prompts with two synthetic parallel examples each, it essentially uses eight synthetic examples, instead of just two real examples in the few-shot setting. Our synthetic examples are nearly as high-quality as real examples (similar to the findings of Han et al. (2021)) as demonstrated by the ablation in Appendix D. Prompt ensembling not only reduces performance variation if low-quality synthetic examples are selected during the bootstrap, but it also boosts performance beyond the few-shot setting as demonstrated by Table 1 and the Appendix D ablation (Russian-English 26.9 → 27.9 spBLEU).

We also compare our WMT14 (Bojar et al., 2014) results to those of GPT-3_{175B} from Han et al. (2021) in Appendix B. Our performance nearly matches (<0.5 BLEU) the performance of the largest GPT-3 model on high-resource language pairs. This is in spite of our approach using only 2% of the number of the parameters of GPT-3_{175B}.

5 Other Tasks

We next demonstrate that bidirectional models have a generalized ability, beyond machine translation, to be prompted for arbitrary tasks. We evaluate their performance on question answering and summarization tasks. Example generations can be found in Appendix G.

5.1 Question Answering

We compare the zero-shot question answering performance of mT5 against XGLM on the XQuAD dataset (Artetxe et al., 2020), a multilingual question answering dataset, in Table 3. We find mT5 with SAP outperforms XGLM significantly (+1.7/+12.3 EM/F1).

In Table 4, we also compare against T5+LM (Lester et al., 2021) described in Section 2.2. As T5+LM is English-only, we compare using the English-only SQuAD v1.1 dataset (Rajpurkar et al., 2016). We still utilize the multilingual mT5 with SAP due to observations

		en	ar	de	el	es	hi	ru	th	tr	vi	zh	avg
XGLM$_{7.5B}$	(zero-shot)	19.5/31.9	12.9/29.6	12.2/25.3	7.2/28.2	12.5/24.0	**11.0**/14.0	10.9/27.8	**16.8/26.4**	13.6/26.8	12.5/21.2	13.2/20.3	12.9/25.0
mT5$_{3.7B}$ + SAP	(zero-shot)	**25.0/48.8**	**17.4/39.4**	**19.4/43.0**	**9.7/41.0**	**15.0/42.1**	6.6/**32.1**	**16.1/39.0**	2.8/17.4	**15.8/37.0**	**18.2/41.9**	**15.0/29.0**	**14.6/37.3**

Table 3: Zero-shot multilingual question answering results (EM/F1) on the XQuAD test set (Artetxe et al., 2020).

		EM	F1
Zero-shot			
T5+LM$_{3B}$	(zero-shot)	23.5	48.4
mT5$_{3.7B}$ + SAP	(zero-shot)	**30.2**	**54.0**
Few-shot			
mT5$_{3.7B}$	(16-shot)	23.0	54.5
mT5$_{3.7B}$ + SAP	(16-shot)	**35.4**	**60.0**

Table 4: Zero-shot and few-shot question answering results on the SQuAD v1.1 dev set (Rajpurkar et al., 2016).

		ROUGE-1	ROUGE-2	ROUGE-L
T5+LM$_{3B}$	(2-shot)	14.1	4.4	13.2
mT5$_{3.7B}$	(2-shot)	15.9	4.5	15.0
mT5$_{3.7B}$ + SAP	(2-shot)	**22.0**	**6.8**	**20.3**

Table 5: Few-shot summarization results on the CNN / Daily Mail v3.0.0 test set evaluated with ROUGE (Nallapati et al., 2016; See et al., 2017; Hermann et al., 2015; Lin, 2004).

that the English-only T5 v1.1 model does not perform as well as mT5 in prompt-based learning[5]. SAP achieves +6.7/+5.6 EM/F1 over T5+LM.

SAP, as an iterative technique, is useful for producing long generations from a bidirectional model for tasks such as machine translation. We find, however, it still has utility on tasks like question answering where answer generations are shorter spans of text. We ablate utilizing SAP with mT5 against the simple approach of prompting mT5 once and using the mask in-fill generated on SQuAD v1.1. In the few-shot (16-shot) setting, we find that utilizing SAP still markedly improves performance (+12.5/+5.5 EM/F1) even on short-form generation tasks like question answering.

5.2 Summarization

We next perform summarization on the CNN/Daily Mail v3.0.0 dataset (Nallapati et al., 2016; See et al., 2017; Hermann et al., 2015) as another long-form text generation task. In the few-shot setting, we compare mT5 with T5+LM and ablate the usage of SAP once again in Table 5. Again, we find a significant lead against T5+LM with +7.1 ROUGE-L. Of that +7.1 ROUGE-L boost, an ablation of our usage of SAP finds the SAP technique itself is responsible for a large component of the boost, +5.3 ROUGE-L.

6 Conclusion and Future Directions

In this paper, we introduce Sequential Autoregressive Prompting (SAP), a novel technique to prompt bidirectional models without fine-tuning. We demonstrate SAP with the bidirectional mT5 model enables few- and zero-shot machine translation and zero-shot multilingual question answering that outperforms unidirectional models, despite using far fewer parameters and examples.

Our results suggest that the bidirectionality of models such as mT5 contributes to their improved performances in machine translation and multilingual question answering, even with fewer parameters. The representional power of bidirectionality is something both the authors of GPT-3 and XGLM have explicitly stated as desiderata, but did not experiment with, lacking a method to prompt bidirectional models (Brown et al., 2020; Lin et al., 2021). Still, we concede that our results do not conclusively prove bidirectionality explains the difference in performance. Beyond bidirectionality and pre-training objectives, mT5, XGLM, and GPT-3 further differ in architecture, pre-training corpus, and hyperparameters. A complete ablation experiment here would be computationally expensive, and we leave it as future work.

Importantly, these results demonstrate bidirectional models possess few-shot and zero-shot learning capabilities innately, without the previously required post-hoc modifications discussed in Section 2.2. We show that prompt-based learning and few-shot learning is an emergent property of bidirectional models and they can outperform unidirectional models on tasks that benefit from bidirectionality. Our results contribute strong evidence towards the strength and efficiency of bidirectional pre-training objectives and motivate further research into bidirectional architectures, pre-training objectives, and language models designed and optimized for prompting and few-shot learning.

[5] We discuss this observation in more detail in Appendix F.

7 Limitations

The main limitation of this work lies in the efficiency of our technique. SAP requires T total forward passes to produce a generation instead of a single forward pass, where T equals the number of words in the generation before reaching a stop token. For example, to produce a translation that has 14 words, SAP requires 14 inferences of the bidirectional model. For tasks with shorter generations with only a few words, such as multilingual question answering, SAP is more practical, especially since it uses fewer parameters. While these inferences must be performed sequentially due to the autoregressive nature of the technique, utilizing batching over a test set can still ensure maximum GPU utilization, which is how our experiments were performed. Nevertheless, SAP uncovers an important result: prompting is an emergent property of bidirectional models. We hypothesize that further research into pre-training objectives and language model design following Wang et al. (2022) could yield a bidirectional pre-training objective better optimized for few-shot prompting, lifting the requirement to perform multiple forward passes sequentially to generate longer completions.

8 Ethical Considerations and Broader Impacts

Energy and efficiency The technique we describe in this paper does not require fine-tuning in order to perform machine translation which is computationally expensive. By avoiding fine-tuning and utilizing prompting, a single large language model can be used for many downstream tasks, a significantly more efficient approach than using a different model per downstream task.

Diversity and inclusion While our work contributes to the greater body of research enabling machine translation of low-resource languages where machine translation has typically underperformed compared to high-resource languages, our work does rely on English-centric techniques to improve performance on low-resource languages.

References

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4623–4637.

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Ales Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

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

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *ACL*.

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

Naman Goyal, Cynthia Gao, Vishrav Chaudhary, Peng-Jen Chen, Guillaume Wenzek, Da Ju, Sanjana Krishnan, Marc'Aurelio Ranzato, Francisco Guzman, and Angela Fan. 2021. The flores-101 evaluation benchmark for low-resource and multilingual machine translation. *arXiv preprint arXiv:2106.03193*.

Jesse Michael Han, Igor Babuschkin, Harrison Edwards, Arvind Neelakantan, Tao Xu, Stanislas Polu, Alex Ray, Pranav Shyam, Aditya Ramesh, Alec Radford, et al. 2021. Unsupervised neural machine translation with generative language models only. *arXiv preprint arXiv:2110.05448*.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NIPS*, pages 1693–1701.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona T. Diab, Veselin Stoyanov, and Xian Li. 2021. Few-shot learning with multilingual language models. *CoRR*, abs/2112.10668.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pretrain, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

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. *arXiv preprint arXiv:1907.11692*.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2021. Metaicl: Learning to learn in context. *arXiv preprint arXiv:2110.15943*.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. 2016. Abstractive text summarization using sequence-to-sequence RNNs and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, Berlin, Germany. Association for Computational Linguistics.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

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. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21:1–67.

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

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2022. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations*.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *EACL*.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. 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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Alex Wang and Kyunghyun Cho. 2019. Bert has a mouth, and it must speak: Bert as a markov random field language model. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 30–36.

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

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A FLORES-101 Few-shot and Unsupervised Zero-shot Machine Translation Results

			en	de	fr	ca	fi	ru	bg	zh	ko	ar	sw	hi	my	ta	avg
en	Supervised		–	32.6	42.0	31.2	24.2	27.1	37.4	19.3	18.5	17.9	26.9	28.1	3.5	3.4	24.0
	GPT-3$_{6.7B}$	(32-shot)	–	25.9	**36.1**	23.8	10.2	11.2	5.9	12.5	1.2	1.1	0.5	0.3	0.1	0.0	9.9
	XGLM$_{7.5B}$	(32-shot)	–	**27.6**	36.0	**34.0**	**23.3**	**24.2**	**33.1**	**15.6**	**12.0**	11.5	18.0	**19.9**	11.0	8.5	**21.1**
	mT5$_{3.7B}$ + SAP	(2-shot)	–	23.2	34.2	26.2	15.8	20.1	27.9	9.5	10.4	11.4	17.3	14.0	11.0	**11.2**	17.9
	mT5$_{3.7B}$ + SAP	(zero-shot)	–	26.0	33.2	28.4	15.7	21.2	27.1	11.3	10.5	**12.7**	**19.1**	16.1	**13.2**	**13.1**	19.0
de	Supervised		35.8	–	35.5	25.8	22.6	24.6	31.5	17.2	16.6	14.8	21.0	23.4	2.3	2.3	21.0
	GPT-3$_{6.7B}$	(32-shot)	**40.4**	–	26.2	17.2	8.1	9.3	4.8	9.0	1.0	0.9	0.5	0.3	0.1	0.1	9.1
	XGLM$_{7.5B}$	(32-shot)	38.8	–	**27.9**	19.1	**20.5**	**19.7**	**25.8**	**12.3**	3.4	6.6	11.7	**14.3**	**9.9**	4.8	**16.5**
	mT5$_{3.7B}$ + SAP	(2-shot)	33.0	–	24.4	17.8	14.1	15.7	20.2	8.2	**9.1**	7.7	11.0	10.0	9.8	9.6	14.7
	mT5$_{3.7B}$ + SAP	(zero-shot)	35.9	–	25.9	**22.5**	14.3	17.4	21.0	8.2	8.4	**8.7**	**13.4**	10.4	9.0	**10.8**	15.8
fr	Supervised		37.2	28.5	–	28.7	21.9	24.5	32.2	17.6	16.7	15.4	17.2	22.9	2.1	0.8	20.4
	GPT-3$_{6.7B}$	(32-shot)	**42.8**	20.9	–	23.7	8.0	9.7	4.6	9.1	1.0	1.0	0.4	0.3	0.1	0.0	9.4
	XGLM$_{7.5B}$	(32-shot)	40.4	20.4	–	**32.1**	**19.4**	**19.8**	**26.3**	**10.6**	2.4	5.9	14.5	**13.7**	9.7	6.6	**17.1**
	mT5$_{3.7B}$ + SAP	(2-shot)	38.0	19.2	–	26.7	13.7	18.3	23.5	8.6	**9.2**	9.9	15.0	12.1	**10.8**	9.7	16.5
	mT5$_{3.7B}$ + SAP	(zero-shot)	38.1	**21.1**	–	30.1	12.9	18.1	22.3	8.7	**9.2**	**11.1**	**15.7**	11.0	9.6	**11.1**	16.8
ca	Supervised		33.4	24.8	35.1	–	19.0	21.1	28.6	15.1	13.9	13.4	18.7	20.5	2.1	2.6	19.1
	GPT-3$_{6.7B}$	(32-shot)	40.2	18.6	31.4	–	7.0	9.3	4.3	8.0	0.9	0.9	0.3	0.4	0.1	0.1	9.3
	XGLM$_{7.5B}$	(32-shot)	**41.1**	18.9	**33.8**	–	11.3	3.3	**23.9**	**10.8**	1.3	0.8	13.8	6.1	7.9	3.1	13.6
	mT5$_{3.7B}$ + SAP	(2-shot)	33.4	14.9	29.5	–	10.7	14.0	15.6	6.5	7.0	5.6	12.4	7.3	**8.7**	6.7	13.3
	mT5$_{3.7B}$ + SAP	(zero-shot)	37.1	**19.3**	32.4	–	**12.4**	**16.7**	19.1	7.9	**7.4**	**8.5**	**14.5**	**9.4**	8.3	**9.8**	**15.6**
fi	Supervised		27.2	23.0	29.3	21.6	–	20.6	26.4	16.0	14.8	12.4	14.2	19.8	1.7	0.9	17.5
	GPT-3$_{6.7B}$	(32-shot)	25.3	13.5	17.1	10.0	–	6.4	2.8	5.7	0.7	0.7	0.3	0.3	0.1	0.0	6.4
	XGLM$_{7.5B}$	(32-shot)	**29.2**	**17.4**	**22.2**	**17.0**	–	**16.5**	**17.5**	**12.4**	**7.5**	**7.6**	8.0	**10.1**	6.2	2.0	**13.4**
	mT5$_{3.7B}$ + SAP	(2-shot)	24.1	16.1	19.8	14.9	–	14.2	17.0	7.0	5.8	7.1	8.3	5.6	**8.5**	3.9	11.7
	mT5$_{3.7B}$ + SAP	(zero-shot)	23.2	16.1	20.5	16.3	–	14.5	16.3	8.0	5.9	6.3	**10.0**	7.5	5.9	**8.2**	12.2
ru	Supervised		27.5	23.5	30.1	22.0	19.4	–	31.0	16.5	15.3	13.5	18.1	20.9	2.2	2.3	18.6
	GPT-3$_{6.7B}$	(32-shot)	28.1	14.8	20.4	13.1	5.4	–	7.4	1.2	0.2	0.2	0.1	0.2	0.1	0.1	7.0
	XGLM$_{7.5B}$	(32-shot)	**30.4**	**17.9**	**24.0**	14.6	8.0	–	**26.3**	**11.6**	5.5	7.4	7.1	**9.1**	7.3	3.1	13.2
	mT5$_{3.7B}$ + SAP	(2-shot)	26.9	16.6	22.4	14.5	11.2	–	25.2	6.1	8.0	6.4	11.3	**9.1**	**9.8**	8.4	13.5
	mT5$_{3.7B}$ + SAP	(zero-shot)	27.9	17.1	22.5	**19.4**	**13.1**	–	25.4	8.3	**8.7**	**9.1**	**12.0**	9.0	9.0	**10.3**	**14.8**
bg	Supervised		33.0	26.1	33.7	24.9	20.8	26.5	–	17.5	16.4	14.5	20.9	23.1	2.3	2.4	20.2
	GPT-3$_{6.7B}$	(32-shot)	21.6	11.4	16.0	9.7	4.3	6.5	–	1.2	0.2	0.2	0.1	0.2	0.1	0.1	5.5
	XGLM$_{7.5B}$	(32-shot)	**35.5**	**19.2**	**26.3**	12.9	**14.2**	22.9	–	**11.9**	6.8	**9.2**	9.4	7.5	3.2	1.0	13.9
	mT5$_{3.7B}$ + SAP	(2-shot)	31.0	17.0	23.8	18.3	10.9	22.9	–	7.2	**8.3**	8.1	11.7	7.4	**9.5**	6.6	14.1
	mT5$_{3.7B}$ + SAP	(zero-shot)	32.5	17.3	24.5	**21.7**	10.6	**23.2**	–	8.7	7.5	9.0	**13.0**	**8.6**	7.9	**10.1**	**15.0**
zh	Supervised		20.9	17.6	24.3	17.4	16.0	17.2	22.1	–	15.9	11.6	15.5	18.5	1.9	2.5	15.5
	GPT-3$_{6.7B}$	(32-shot)	**21.1**	9.5	14.3	8.2	4.3	3.6	1.3	–	1.1	0.4	0.2	0.2	0.1	0.0	4.9
	XGLM$_{7.5B}$	(32-shot)	20.7	8.3	8.5	10.5	4.4	4.8	**14.8**	–	**9.3**	4.2	5.6	**12.0**	8.6	6.2	9.1
	mT5$_{3.7B}$ + SAP	(2-shot)	19.0	**10.9**	**14.9**	11.9	8.0	10.6	11.9	–	8.9	6.0	**9.1**	8.0	**10.0**	7.6	10.5
	mT5$_{3.7B}$ + SAP	(zero-shot)	18.5	**10.9**	14.8	**12.8**	**8.8**	**10.7**	11.8	–	9.2	**6.5**	9.0	8.9	8.2	**8.9**	**10.7**
ko	Supervised		20.9	16.7	22.1	16.5	14.9	15.5	21.1	15.7	–	10.6	15.1	18.7	1.9	4.0	14.9
	GPT-3$_{6.7B}$	(32-shot)	8.3	4.6	6.4	4.4	2.1	1.7	0.8	2.5	–	0.2	0.1	0.1	0.1	0.1	2.4
	XGLM$_{7.5B}$	(32-shot)	**19.9**	**10.3**	13.7	5.3	1.4	1.2	10.9	**11.9**	–	2.7	3.2	1.0	2.2	1.4	6.5
	mT5$_{3.7B}$ + SAP	(2-shot)	18.3	10.1	13.7	11.3	**7.9**	**10.1**	**12.6**	7.8	–	**6.3**	7.2	6.6	2.6	4.7	9.2
	mT5$_{3.7B}$ + SAP	(zero-shot)	18.1	10.1	**13.8**	**12.8**	7.8	9.9	11.4	7.6	–	5.5	**8.0**	**6.7**	**8.1**	**8.2**	**9.8**
ar	Supervised		25.5	18.7	25.7	18.9	15.6	17.8	23.8	13.1	13.3	–	15.4	19.4	1.8	0.9	16.1
	GPT-3$_{6.7B}$	(32-shot)	10.5	5.3	9.6	6.0	2.2	2.2	0.9	0.9	0.1	–	0.1	0.1	0.2	0.0	2.9
	XGLM$_{7.5B}$	(32-shot)	**27.7**	**12.2**	17.9	8.8	**8.5**	9.1	**18.4**	**8.9**	0.8	–	7.7	7.8	3.4	3.7	10.4
	mT5$_{3.7B}$ + SAP	(2-shot)	23.7	10.8	17.5	11.0	8.0	12.2	13.8	5.9	7.1	–	10.3	**8.0**	8.0	8.0	11.1
	mT5$_{3.7B}$ + SAP	(zero-shot)	26.9	11.5	**19.8**	**15.9**	7.8	**14.5**	13.6	6.3	**7.6**	–	**11.0**	**8.0**	**8.8**	**9.3**	**12.4**
sw	Supervised		30.4	19.4	26.7	20.1	15.6	17.6	23.8	13.2	12.2	12.0	–	19.2	2.1	4.0	16.6
	GPT-3$_{6.7B}$	(32-shot)	5.0	2.9	3.9	2.8	1.7	1.8	1.3	1.3	0.5	0.5	–	0.4	0.1	0.1	1.7
	XGLM$_{7.5B}$	(32-shot)	**31.6**	13.4	**21.8**	15.4	**10.2**	13.1	15.2	**9.5**	**6.0**	**8.9**	–	**7.6**	3.4	1.0	**12.1**
	mT5$_{3.7B}$ + SAP	(2-shot)	27.0	12.6	19.0	15.1	9.2	12.2	**15.8**	5.9	**6.0**	8.3	–	6.5	**5.4**	6.0	11.5
	mT5$_{3.7B}$ + SAP	(zero-shot)	30.0	**13.5**	20.0	**18.0**	9.5	**14.5**	**15.8**	6.9	5.7	7.7	–	6.5	2.7	**7.0**	**12.1**
hi	Supervised		27.9	19.4	25.9	18.9	15.7	16.9	23.9	13.5	13.9	12.2	16.8	–	2.5	3.8	16.2
	GPT-3$_{6.7B}$	(32-shot)	1.2	0.9	1.4	0.8	0.4	0.4	0.3	0.2	0.1	0.1	0.1	–	0.1	0.2	0.5
	XGLM$_{7.5B}$	(32-shot)	25.2	12.3	15.4	8.8	**9.8**	11.5	11.3	**10.8**	**8.5**	6.1	4.7	–	1.5	1.9	9.8
	mT5$_{3.7B}$ + SAP	(2-shot)	25.7	12.4	17.0	13.0	8.0	12.2	**15.4**	7.2	4.4	7.4	**8.9**	–	9.6	9.0	11.6
	mT5$_{3.7B}$ + SAP	(zero-shot)	**27.1**	**12.6**	**17.3**	**14.3**	9.0	**12.4**	14.5	8.0	6.7	**8.1**	**8.9**	–	**10.2**	**12.8**	**12.5**
my	Supervised		10.0	6.9	10.4	8.5	6.0	6.7	9.5	5.7	6.1	4.6	7.2	9.1	–	2.5	7.2
	GPT-3$_{6.7B}$	(32-shot)	0.5	0.3	0.4	0.4	0.2	0.1	0.2	0.0	0.0	0.0	0.1	0.2	–	0.1	0.2
	XGLM$_{7.5B}$	(32-shot)	14.1	7.6	10.1	3.8	5.7	7.1	8.9	**7.1**	**6.9**	3.6	3.5	**8.9**	–	2.6	6.9
	mT5$_{3.7B}$ + SAP	(2-shot)	**16.8**	8.5	**12.9**	11.0	6.7	6.1	9.2	5.2	2.9	**5.0**	**8.0**	7.0	–	5.7	8.1
	mT5$_{3.7B}$ + SAP	(zero-shot)	16.4	**9.0**	11.9	**11.6**	**6.9**	**8.3**	**10.4**	5.5	3.6	4.8	6.4	7.1	–	**6.2**	**8.3**
ta	Supervised		8.3	4.9	6.8	5.8	5.0	4.7	7.0	2.5	2.3	1.1	5.2	6.9	1.2	–	4.8
	GPT-3$_{6.7B}$	(32-shot)	1.0	0.5	0.8	0.5	0.2	0.3	0.3	0.1	0.2	0.1	0.1	0.2	0.0	–	0.3
	XGLM$_{7.5B}$	(32-shot)	16.3	8.4	10.3	5.1	5.2	8.1	7.6	**8.1**	6.2	5.4	2.8	7.2	0.9	–	7.1
	mT5$_{3.7B}$ + SAP	(2-shot)	18.7	10.4	13.7	10.9	6.3	9.8	11.6	5.2	0.7	6.5	6.0	9.3	1.8	–	8.5
	mT5$_{3.7B}$ + SAP	(zero-shot)	**20.4**	**10.5**	**14.7**	**12.9**	**8.1**	**10.6**	**13.2**	7.0	**6.8**	**6.6**	**8.3**	**10.1**	**2.6**	–	**10.1**
avg	Supervised		26.0	20.2	26.7	20.0	16.7	18.5	24.5	14.1	13.5	11.8	16.3	19.3	2.1	2.5	16.6
	GPT-3$_{6.7B}$	(32-shot)	18.9	9.9	14.2	9.3	4.2	4.8	2.7	4.0	0.6	0.5	0.2	0.3	0.1	0.1	5.0
	XGLM$_{7.5B}$	(32-shot)	**28.5**	14.9	20.6	14.4	**10.9**	12.4	**18.5**	**10.9**	5.9	6.1	8.5	**9.7**	5.8	3.5	12.2
	mT5$_{3.7B}$ + SAP	(2-shot)	25.8	14.1	20.2	15.6	10.0	13.7	16.9	6.9	6.8	7.4	10.5	8.5	**8.1**	7.5	12.3
	mT5$_{3.7B}$ + SAP	(zero-shot)	27.1	**15.0**	**20.9**	**18.2**	10.5	**14.8**	17.1	7.9	**7.5**	**8.0**	**11.5**	9.2	8.0	**9.7**	**13.2**

Table 6: Few-shot and unsupervised zero-shot machine translation results on FLORES-101 devtest (spBLEU). Source language in rows, target language in columns. GPT-3$_{6.7B}$ and XGLM$_{7.5B}$ use 32 examples from the dev set for few-shot learning. mT5$_{3.7B}$ uses 2 examples from the dev set for few-shot learning. Supervised results correspond to the M2M-124 615M model from Goyal et al. (2021). XGLM$_{7.5B}$ results correspond to the model from Lin et al. (2021). Underline denotes better than supervised, bold denotes best of GPT-3, XGLM, and mT5. spBLEU computed using the implementation from Goyal et al. (2021).

B WMT14 Unsupervised Zero-shot Machine Translation Results

		English-French	French-English
GPT-3_{175B}	(self-amplified)	**30.0**	**31.8**
$mT5_{3.7B}$ + SAP	(self-amplified)	29.8	31.4

Table 7: Unsupervised zero-shot machine translation results on WMT14 English-French test set (SacreBLEU) (Bojar et al., 2014; Post, 2018). GPT-3_{175B} (self-amplified) results correspond to the unsupervised zero-shot "GPT-3 (self-amplified)" results from Han et al. (2021) prior to performing distillation, initial backtranslation, and iterative backtranslation which involved unsupervised weight updates. $mT5_{3.7B}$ (self-amplified) is our fully unsupervised zero-shot approach outlined in Section 4 with a 16 prompt ensemble. The SacreBLEU signature used also follows Han et al. (2021):
`BLEU+case.mixed+numrefs.1+smooth.exp+tok.intl+version.1.2.20)`

C Random Selection vs. mT5Score Filtering and Selection Ablation

	English-Russian	Russian-English
Random Selection	0.0	25.5
mT5Score Filtering and Selection	**20.0**	**26.3**

Table 8: Unsupervised zero-shot machine translation results on FLORES-101 devtest (spBLEU) using $mT5_{3.7B}$ as described in Section 4. In this experiment, we ablate utilizing mT5Score to filter and select the high-quality synthetic examples during bootstrapping over two language pairs, English-Russian and Russian-English. When using random selection, the synthetic parallel examples choosen may be extremely low-quality or non-sensical leading to a 0.0 spBLEU score after self-amplification as shown for the English-Russian language pair.

D Single Prompt vs. Prompt Ensemble Ablation

	English-Russian	Russian-English
Single Prompt	20.0	26.3
4 Prompt Ensemble	**20.9**	27.9
8 Prompt Ensemble	20.7	**28.6**
16 Prompt Ensemble	**20.9**	**28.6**

Table 9: Unsupervised zero-shot machine translation results on FLORES-101 devtest (spBLEU) using $mT5_{3.7B}$ as described in Section 4. In this experiment, we ablate utilizing a single few-shot prompt with two synthetic parallel examples to perform the final translation with utilizing an ensemble of 4, 8, and 16 distinct few-shot prompts each with two synthetic parallel examples that generate 4, 8, and 16 translations respectively from which the best translation (by mT5Score) is selected as the final translation over two language pairs, English-Russian and Russian-English.

E Standard Bootstrap vs. English-centric Bootstrap Ablation

	English-Russian	Russian-Chinese
Standard bootstrap	20.9	5.8
English-centric bootstrap	**21.2**	**8.3**

Table 10: Unsupervised zero-shot machine translation results on FLORES-101 devtest (spBLEU) using $mT5_{3.7B}$ as described in Section 4. In this experiment, we ablate performing the standard bootstrap generally described in Section 4 with the English-centric bootstrap described in Section 4.3 over two language pairs, English-Russian and Russian-Chinese.

F Prompting T5 v1.1 with SAP

Ideally, our experiments on question answering on the SQuAD v1.1 dataset and summarization on the CNN / Daily Mail v3.0.0 dataset would utilize the English-only T5 v1.1 model instead of mT5, since the datasets are English-only and there is no need for multilinguality. We choose to utilize mT5 for all results in this paper due to the observation that T5 v1.1 cannot be prompted as easily as mT5 and underperforms for that reason.

The inputs seen by T5 v1.1 and mT5 during pre-training are of sequence length 512 tokens where multiple spans in the sequence are dropped (Raffel et al., 2020). Therefore, the prompt template we describe in Section 3, would be out-of-distribution from the pre-training inputs since it may have a sequence length shorter or longer than 512 tokens and only contains a single mask instead of multiple masks.

We find that the mT5 model has generalized to sequences shorter and longer than 512 tokens and to sequences that only contain a single mask, while the T5 v1.1 model has not. It is still possible to prompt the T5 v1.1 model with SAP, but requires formulating a prompt that is in-distribution with the pre-training inputs which constrains the length of the prompt.

Due to this complication, we forgo prompting T5 v1.1 altogether in this paper. Since mT5 and T5 v1.1 were trained identically, apart from mT5 being pre-trained on the multilingual mC4 dataset instead of the primarily English C4 dataset, we hypothesize that this difference between T5 v1.1 and mT5 may be an artifact of which checkpoint is selected after pre-training or the length of pre-training (Xue et al., 2021; Raffel et al., 2020).

G Selected Example Generations

Task: Few-shot Machine Translation (Example #1)

Dataset: FLORES-101 (Arabic → English)

Prompt Template:

```
Translate Arabic to English.
{{examples}}
Arabic:  {{source_text}}
English:
```

Ground Truth:

```
The 802.11n standard operates on both the 2.4Ghz and 5.0Ghz frequencies.
```

Generation (mT5$_{3.7B}$ + SAP):

```
The wireless standard 802.11n operates at the frequency of 2.4 GHz and 5 GHz.
```

Generation (mT5$_{3.7B}$):

```
The 802.11n wireless standard operates at 2.4 and 5.0
```

Commentary:

SAP generates a lengthier and more fluent translation and correctly translates the units of the frequencies. Both generations add the word "wireless" which is used correctly and is likely to appear next to the words "802.11n" and "standard" but does not exist in the ground truth translation.

Task: Few-shot Machine Translation (Example #2)

Dataset: FLORES-101 (Russian $\rightarrow$ English)

Prompt Template:

```
Translate Russian to English.
{{examples}}
Russian:  {{source_text}}
English:
```

Ground Truth:

```
In 1956 Słania moved to Sweden, where three years later he began work for the
Swedish Post Office and became their chief engraver.
```

Generation (mT5$_{3.7B}$ + SAP):

```
In 1956, Slania moved to Sweden, where he worked for three years for the Swedish
Post Office and became its chief engraved worker.
```

Generation (mT5$_{3.7B}$):

```
In 1956, Slanya moved to Sweden and became...
```

Commentary:

SAP generates a full length translation and more correctly translates “Słania” to “Slania” instead of “Slanya”. While the translation without SAP only generates a partial translation, the word “became” indicates the direction of translation would be less close to the ground truth translation than the direction of translation taken by SAP. Notably, SAP produces a relatively high-quality translation, but a common failure mode is displayed in this example. SAP translates “chief engraver” to “chief engraved worker” which is an imperfect paraphrase likely due to an imperfect multilingual alignment of the word “engraver” in the embedding space of the model.

Task: Few-shot Question Answering (Example #1)

Dataset: SQuAD v1.1

Prompt Template:

```
Answer the question based on the following passage.

{{examples}}

Passage: {{passage}}
Question: {{question}}
Answer:
```

Passage:

```
In 1874, Tesla evaded being drafted into the Austro-Hungarian Army in Smiljan
by running away to Tomingaj, near Gračac. There, he explored the mountains in
hunter's garb. Tesla said that this contact with nature made him stronger, both
physically and mentally. He read many books while in Tomingaj, and later said that
Mark Twain's works had helped him to miraculously recover from his earlier illness.
```

Question:

```
Why did Tesla avoid by fleeing Smiljan?
```

Ground Truth:

```
being drafted into the Austro-Hungarian Army
```

Generation ($\text{mT5}_{3.7\text{B}}$ + SAP):

```
because he was ill and wanted to avoid being drafted into the Austro-Hungarian Army
```

Generation ($\text{mT5}_{3.7\text{B}}$):

```
because he was ill and could not leave the country
```

Commentary:

In this example, the grammaticality of the question itself ("*Why* did Tesla avoid by fleeing Smiljan?" vs. "*What* did Tesla avoid by fleeing Smiljan?") has issues. This seems to cause both generations to attempt to answer a "why" style question with "because" instead of a "what" style question. Notably, the answer generated by SAP does eventually reach correct answer where as the the answer generated without SAP hallucinates a fact: "he [Tesla] . . . could not leave the country".

Task: Few-shot Question Answering (Example #2)

Dataset: SQuAD v1.1

Prompt Template:

```
Answer the question based on the following passage.

{{examples}}

Passage:  {{passage}}
Question:  {{question}}
Answer:
```

Passage:

```
The Broncos took an early lead in Super Bowl 50 and never trailed.  Newton was
limited by Denver's defense, which sacked him seven times and forced him into
three turnovers, including a fumble which they recovered for a touchdown.  Denver
linebacker Von Miller was named Super Bowl MVP, recording five solo tackles, 2½
sacks, and two forced fumbles.
```

Question:

```
Who won the MVP for the Super Bowl?
```

Ground Truth:

```
Von Miller
```

Generation (mT5$_{3.7B}$ + SAP):

```
Von Miller
```

Generation (mT5$_{3.7B}$):

```
Von Miller, Denver, NFL, NFLPA
```

Commentary:

Without using SAP, the answer generated is correct, but contains hallucinated tokens towards the end of the generation.

Task: Few-shot Summarization (Example #1)

Dataset: CNN / Daily Mail v3.0.0

Prompt Template:

```
Summarize the article.

{{examples}}

Article: {{article}}
Summary:
```

Article:

```
London (CNN)The Hatton Garden heist, as it will surely come to be known, was every
safe deposit box holder's nightmare, every movie director's dream. Thieves using
heavy cutting equipment and rappelling gear broke into the vault of an esteemed
60-year-old safe deposit company in the heart of London over the past holiday
weekend, possibly taking advantage of as many as four days to rifle through an
uncounted number of safe deposit boxes. And they reportedly got away with hundreds
of thousands of pounds worth of gems and cash - even, in the educated guess of one
former police official, as much as 200 million pounds, or...[truncated]
```

Summary:

```
Robbers may have taken advantage of a four-day holiday weekend. Estimates of the
value of the items taken rage from hundreds of thousands of pounds to 200 million
pounds. The heist took place in a historic heart of London's jewelry business.
```

Generation (mT5$_{3.7B}$ + SAP):

```
The Hatton Garden heist was a nightmare for jewelers and their customers. They
were 'treated like criminals', while officers said they would double bail for anyone
who was arrested before. They were arrested while on the sidewalk .
```

Generation (mT5$_{3.7B}$):

```
The latest jewel heist in London took place over the weekend.
```

Commentary:

SAP generates a lengthier generation with multiple sentences summarizing the article. Some of the facts presented in the generation, however, are not coherent with the article.

Task: Few-shot Summarization (Example #2)

Dataset: CNN / Daily Mail v3.0.0

Prompt Template:

```
Summarize the article.

{{examples}}

Article:  {{article}}
Summary:
```

Article:

```
Sanaa, Yemen (CNN)Al Qaeda fighters attacked a prison in the coastal Yemeni city
of Al Mukallah early Thursday, freeing at least 270 prisoners, a third of whom have
al Qaeda links, a senior Defense Ministry official has told CNN. Khaled Batarfi, a
senior al Qaeda figure, was among the escapees, officials said.  Dozens of attackers
took control of government buildings, including the city's Central Prison, Central
Bank and radio station during the assault early Thursday, according to officials.
Government troops arrived early Thursday and clashed with the al Qaeda fighters, and
most of the militants fled,...[truncated]
```

Summary:

```
Al Qaeda fighters attack a prison and other government buildings, freeing many
prisoners.  Government troops clash with the fighters, most of whom flee.  Yemen
is descending into chaos as a Shia-Sunni conflict draws in regional rivals Saudi
Arabia and Iran.
```

Generation (mT5$_{3.7B}$ + SAP):

```
Yemen's al Qaeda fighters escape from prison after clash with government forces.
```

Generation (mT5$_{3.7B}$):

```
Yemen's prisoners escape from al Qaeda fighting in Yemen
```

Commentary:

SAP correctly characterizes the major theme of the article in its summary. Without SAP, the generation devolves in to an incoherent fact ("prisoners escape from al Qaeda fighting") and only a partial summary is generated. The ground truth summary, however, is notably longer and contains multiple sentences, while the summary generated by SAP in this instance is only a single sentence.

H Resources

We provide links and citations to resources used in this paper which provide license information, documentation, and their intended use. Our usage follows the intended usage of all resources.

We utilize the following models:

- mT5 (Xue et al., 2021):
  https://github.com/google-research/multilingual-t5/
- T5 v1.1 (Raffel et al., 2020; Lester et al., 2021):
  https://github.com/google-research/text-to-text-transfer-transformer/
- T5+LM (Raffel et al., 2020; Lester et al., 2021):
  https://github.com/google-research/text-to-text-transfer-transformer/

We utilize the following datasets:

- FLORES-101 (Goyal et al., 2021):
  https://ai.facebook.com/research/publications/the-flores-101-evaluation-benchmark-for-low-resource-and-multilingual-machine-translation
- WMT14 (Bojar et al., 2014):
  https://www.statmt.org/wmt14/translation-task.html
- XQuAD (Artetxe et al., 2020):
  https://github.com/deepmind/xquad
- SQuAD v1.1 (Rajpurkar et al., 2016):
  https://rajpurkar.github.io/SQuAD-explorer/
- CNN / Daily Mail v3.0.0 (Nallapati et al., 2016; See et al., 2017; Hermann et al., 2015):
  https://huggingface.co/datasets/ccdv/cnn_dailymail

We utilize the following software:

- Transformers (Wolf et al., 2019):
  https://github.com/huggingface/transformers
- Datasets (Lhoest et al., 2021):
  https://github.com/huggingface/datasets
- SacreBLEU (Post, 2018; Goyal et al., 2021):
  https://github.com/ngoyal2707/sacrebleu
- ROUGE (Lin, 2004):
  https://github.com/pltrdy/rouge
- BERTScore (Zhang et al., 2019):
  https://github.com/Tiiiger/bert_score/tree/master/bert_score
- langdetect:
  https://pypi.org/project/langdetect/

We estimate the total compute budget and detail computing infrastructure used to run the computational experiments found in this paper below:

- 1x NVIDIA RTX A6000 / 87GB RAM / 4x CPU – 686 hours