Learning to Translate from Soft to Hard LLM Prompts

Anonymous ACL submission

Abstract

Soft prompt tuning is a parameter-efficient method for adapting LLMs to specific tasks, but suffers from a lack of interpretability. Building on recent work on interpreting soft prompts (Ramati et al., 2024), we explore how training a dedicated soft prompt to natural language translation model can yield higher translation quality. In particular, in both quantitative and qualitative comparisons on multiple Datasets of Datasets (DoDs), we demonstrate that our translator produces fluent, accurate verbalizations that outperforms existing training-free methods like InSPEcT. In addition to advancing interpretability, our work suggests a promising downstream application: soft prompts optimized on small, open-source models can be translated into portable text prompts that, when deployed on larger closed-API models, exceed the performance of the original soft prompt and, in some cases, even few-shot learning.

1 Introduction

As an alternative to manual prompt engineering, LLM researchers and practitioners sometimes employ *soft prompt tuning*, whereby a sequence of learnable embeddings is prepended to the input sequence and then fed to an LLM for fine-tuning on some downstream task. Soft prompt tuning matches, and in some cases, surpasses fine-tuning of LLMs while containing significantly fewer parameters (Liu et al., 2022; Lester et al., 2021). Due to their ease of implementation and training, soft prompts are versatile and can be applied to a wide variety of tasks such as text classification (Buchner et al., 2024), relation extraction (Peng et al., 2024), and even drug discovery (Liu et al., 2026). Further, because soft prompts do not modify the model weights themselves, there is reduced risk of catastrophic forgetting (Vu et al., 2022).

In contrast to LLM parameters or adapters, soft prompts share the same embedding space as natural language tokens, and it is thus tempting to hope that they might be more interpretable than other LLM fine-tuning methods. Unfortunately, naive projections of soft prompts into discrete token space are often misleading and yield contradictory results (Khashabi et al., 2022). Some works have found that soft prompts occupy regions in the embedding space that are distinct from those containing natural language (Bailey et al., 2023). Methods designed to optimize soft prompts for interpretability often incur reduced task performance (Patel et al., 2025).

Until recently, it was widely thought that soft prompts, while useful as a light-weight LLM optimization method, were fundamentally not interpretable. However, the InSPEcT technique (Ramati et al., 2024) showed that the same LLM that uses soft prompts for prediction can also be utilized to elicit textual verbalizations of soft prompts. The technique is training-free: it simply relies on the model's in-context learning ability to continue generating tokens after being fed the soft prompt as input. Such verbalizations, it turns out, often provide a glimpse into how the soft prompt operates. Nonetheless, InSPEcT leaves much to be desired in terms of reliable, clear descriptions of what the learned soft prompt means and what behaviors it induces in the LLM using it.

In this paper, we explore whether there are stronger, more effective methods of interpreting soft prompts. In particular, instead of relying on in-context learning, we train a dedicated translation model, using the same architecture (but different model weights) as the LLM on which the soft prompts were trained. We find that our approach is promising and outperforms InSPEcT.

Contributions: (1) We train a soft-to-hard prompt translator by constructing a Dataset of Datasets (DoD) from which many (*soft prompt*, *hard prompt*) pairs can be trained; we then show that the resulting translation model can verbalize soft prompts into natural language that is gram-

matically fluent and has substantial agreement with ground-truth task descriptions. As a baseline, we compare with InSPEcT and observe a clear improvement. (2) Beyond just greater interpretability, the translator's verbalizations also show utility – they can be successfully used as prompts and given to larger LLMs. On a variety of tasks, verbalized prompts achieved superior performance to soft-prompting and even few-shot learning, indicating that the verbalizations are relevant and useful for the intended task. (3) We discuss how the existence of an accurate translator could open the door for novel LLM prompt tuning workflows whereby a small LLM is used to train a soft prompt, which is then translated to a hard prompt and given to a larger more powerful LLM for inference—potentially achieving superior inference performance than the small LLM.

2 Related Works

A number of works have attempted to interpret soft prompts into natural language via obtaining their nearest discrete token projections. However, Khashabi et al. (2022) showed that these naive projections of soft prompts into discrete token space are often misleading, completely uninformative, and in some cases contradictory to the underlying task through a phenomenon they termed *prompt waywardness*. Other works support this finding, showing that soft prompt embeddings occupy regions in the embedding space that are distinct from those containing natural language (Bailey et al., 2023). A number of methods designed to optimize soft prompts for interpretability have found a trade-off between task performance and interpretability (Patel et al., 2025), (Deng et al., 2022).

Closer to the idea of translating a soft prompt into natural language, Ramati et al. (2024) recently proposed InSPEcT, a training-free method of eliciting textual descriptions of soft prompts. InSPEcT works by utilizing the Patchscopes (Ghandeharioun et al., 2024) framework to patch soft prompt activations into a target prompt designed to verbalize descriptions of the task. While InSPEcT can successfully verbalize correct class labels for a given task, we found that verbalizations are often unreliable, sensitive to the specific source and target layers used for patching, and often contain unnatural language. Their experiments were also quite limited and assessed verbalization on only a handful of datasets.

To date, there is no method which can reliably, fluently, and robustly (without any performance-interpretability trade-off) translate soft prompts into natural language. We thus explore whether a more powerful translation model can help.

3 Translating from Soft to Hard Prompts

We explore a simple workflow (see Figure 1) for translating soft prompts to natural language whereby a dedicated translation model is trained with supervised learning on a dataset of (*soft prompt, hard prompt*) pairs. Here, each soft prompt was itself optimized for a specific task – e.g., sentiment classification – on a fixed LLM (typically small and open-source, e.g., Llama-8B) using a dataset of (*input*, *output*) pairs. The trained translation model, given some soft prompt trained on the same LLM, can produce a corresponding "hard prompt", i.e., a natural language translation of the soft prompt. For a sentiment classification soft prompt, the hard prompt might be, "`Examine the input sentence and classify it as positive or negative. A sentence is positive if...`"

Optionally, the generated hard prompt can then be fed to another LLM (typically larger and with a closed API, e.g., GPT-5.5). Given a high-quality verbalization and an arbitrary input example, the larger model possibly delivers higher task accuracy compared to the small LLM on which the soft prompt itself was trained.

4 Experiments

We conduct experiments to investigate the following research questions: (1) Can a dedicated translation model produce more accurate soft prompt translations as compared to just in-context learning methods (such as InSPEcT)? (2) What aspects of the translation model's training set (size, composition, etc.) are important? (3) Are the verbalizations simply plausible-sounding prompts, or are they actually relevant to the task and useful for inference?

4.1 Translator: Architecture & Training Strategy

To train the translator model, we model the task of soft prompt translation as a supervised, sequence-to-sequence language generation task, minimizing cross-entropy of next-token prediction. Conditioned on a sequence of input soft tokens, the translator is tasked with autoregressively generating its

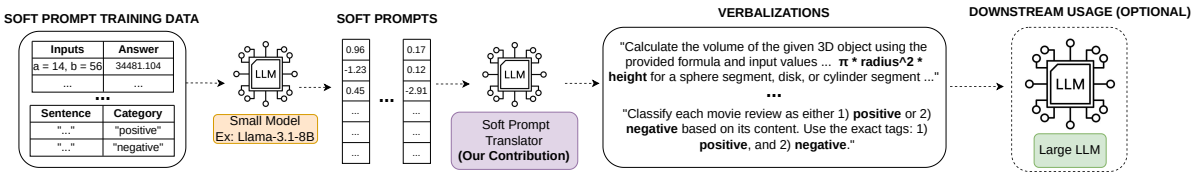

Figure 1: Our proposed hypothetical workflow: From left to right, first a soft prompt is trained on some supervised downstream task on a small-open source LLM. Once trained, the soft prompt is passed to our translator model, which produces a natural language translation. As a possible use case, the translation can then be used on its own as a portable prompt for a large, closed-source LLM for example.

translation in natural language. This follows standard next-token decoding logic, where decoding ends upon seeing an end-of-sequence token. Since the LLM for which the soft prompt was trained clearly has the ability to decode and use it for inference, we decided to construct our translator by fine-tuning the same pre-trained model. Specifically, we used `meta-llama/Llama-3.1-8B-Instruct`.

For training data, we need a labeled dataset of (*soft prompt, hard prompt*) pairs, where the "hard prompt" corresponds to a ground-truth textual description of the task for which the soft prompt was trained. An example of (*soft prompt, hard prompt*) might be $((0.12, -1.94, \ldots)$, `"Classify the news article as: Sports, Business, World, or Culture"`), where the array of numbers is the real-valued vector representing a soft prompt. Each unique soft prompt in the translator's training set also needs to be trained on its own dataset of paired (*input, output*) sequences for some unique downstream task. For example, for a news classification task, the dataset might contain pairs: `("Oil and Economy Cloud...", "Business")`, `(Indians Beat Twins 7-1...", "Sports")`, `("Democratic Senator Urges Energy Reform...", "World")`, etc.

To train a translation model, we thus need a dataset of datasets (DoD). Prior work on deciphering soft prompts (e.g., (Ramati et al., 2024)) evaluated on just 5 datasets and is thus insufficient. Instead, we construct two different DoDs: one synthetic dataset for classification tasks, and another based on the Super-NaturalInstructions dataset (Wang et al., 2022) for natural language tasks such as Text Summarization, Question Generation, Machine Translation, etc.

4.2 Datasets of Datasets (DoDs)

Classification DoD: To examine soft prompts optimized for text classification tasks (like the kind used by (Ramati et al., 2024)), we generated a DoD containing 5500 datasets and 2.75 million sentences in total. Each dataset contains 5 class labels that are tightly semantically clustered. For example, one dataset comprises the labels `music, films, books, art, software`. To generate each label set, we randomly selected a keyword from the Brown Corpus (Francis and Kucera, 1979) as an "abstract concept" and used it to prime an LLM to generate a set of 5 distinguishable related labels. See Appendix A.2 for details.

Within a dataset, each class is associated with 100 sentences (thus, 500 sentences in total per dataset). We generated sentences for each class using an LLM. For class and sentence generation, we used `mistralai/Mistral-Small-3.1-24B-Instruct-2503` as it is a stronger model than our chosen base model. An example of (`input, output`) pairs for one dataset is given in Table 1.

Examples of other label sets in the DoD include `{designer, author, inventor, craftsman, composer}` and `{star, comet, galaxy, planet, nebula}`. During pilot experimentation, we found it was important for the labels within each dataset to be semantically tightly clustered, as otherwise the resulting soft-to-hard prompt translation accuracy suffered (see Appendix A.6).

Input	Output
From lively beats to soothing melodies, there's something for every mood.	**music**
Award winners were celebrated for their exceptional storytelling and artistic direction.	**films**
I'm always on the lookout for limited edition releases signed by the writers.	**books**
The gallery features works from different eras, highlighting the evolution of visual storytelling.	**art**
Some solutions provide comprehensive databases for tracking and retrieving information.	**software**

Table 1: Example of (*input*, *output*) pairs for one of the datasets in the Classification DoD.

Generalized (SuperNatural) DoD: Next, we experimented with more general natural language tasks. We used the Super-NaturalInstructions dataset, which was originally compiled to train and benchmark instruction-following and reasoning ability of LLMs. It contains a very diverse set of 1616 tasks along with labeled datasets for each; it covers 76 distinct task types and 55 different languages (though we used only English-language data in our experiments). It thus seemed a useful basis on which to construct a DoD. We preprocessed the original dataset to create the General DoD containing 539 training and 96 validation tasks, each consisting of 500 (*input, output*) pairs. More details of our preprocessing pipeline can be found in Appendix A.7. To also test scalability of our translator, we performed data augmentation to General DoD, by paraphrasing the original instructions (hard prompts) using `mistralai/Mistral-Small-3.1-24B-Instruct-2503` (see details in Appendix A.8). Example task instructions from the dataset can be found below. There is a large variety of tasks—some more obscure and abstract than others.

> **Example 1**: "In this task you need to indicate the plausibility of reasoning for the pronoun coreference relations...You should answer 'Correct' if the reasoning made sense, otherwise, you should answer 'Wrong'"
>
> **Example 2**: "Generate an overlapping word between the given two sentences. When you find the overlapping words, they don't have to match exactly."

4.3 Training Details

Soft Prompt Training: For each of the two DoDs, we first trained soft prompts: For the Classification DoD, we used an 80/20 train/validation split per dataset and trained a soft prompt consisting of 20 virtual tokens using a learning rate of 1e-4 for 20 epochs (patience of 5 epochs based on the validation set loss), batch size of 32, and the AdamW optimizer. For the General DoD, we used a 90/10 split per dataset, learning rate of 1e-3, 20 epochs (patience of 3 epochs), and batch size of 8.

The trained soft prompts for Classification DoD achieved 73.73% average validation task accuracy (i.e., correct classification rate), while soft prompts for General DoD achieved average validation task ROUGE-L of ~0.17. We note that, although the low ROUGE-L score suggests weak soft prompt performance, the General DoD tasks are often very challenging. Moreover, it is possible that, while the LLM is unable to reliably generate the correct response due to its modest size (8B parameters), the soft prompt itself might nonetheless encode the correct task that the LLM is supposed to be doing. In fact, we show evidence supporting this in Section 5.2 and Appendix A.9. More details on soft prompt training can be found in Appendix A.4.

Translator Training: We train two translator models in total (one for Classification DoD, and one for General DoD). For each, we used a 90/10 train/test split on the (*soft prompt*, *hard prompt*) pairs in each DoD. We used LoRA to fine-tune the base LLM (`meta-llama/Llama-3.1-8B-Instruct`) and the following hyperparameters: learning rate of 1e-4, 5 epochs, batch size of 16, LoRA rank as 4, LoRA dropout as 0.1, with AdamW optimizer with weight decay of 0.1. More details are in Appendix A.5.

For the Classification DoD, we define the target hard prompt for each dataset as a string consisting of a comma separated list of the 5 class labels, e.g., *"nurse, physician, surgeon, pharmacist, radiologist"*. For the General DoD, target hard prompts were already provided by the Super-NaturalInstructions dataset itself.

4.4 Evaluating Soft-to-Hard Prompt Verbalization Quality

We evaluate both the translator and the InSPEcT method on multiple datasets of datasets: Classification DoD, General DoD, as well as the set of 5 datasets used in (Ramati et al., 2024), namely: SST-2 (Socher et al., 2013), SST-5 (Socher et al., 2013), AG-News (Zhang et al., 2015), Subj (Pang and Lee, 2004), and TREC (Voorhees and Tice, 2000); for the remainder of this paper, they will be referred to as InSPEcT datasets.

Metrics: For classification tasks (Classification DoD, and the 5 InSPEcT datasets), we compute the Recall of the ground-truth label set (this was called Class Rate in (Ramati et al., 2024)). We also compute the F1-Score (harmonic mean of Recall and Precision rates) to account for "runaway generations", as it penalizes the presence of irrelevant tokens. For general tasks (General DoD), we use ROUGE-L and cosine similarity of embedded verbalized soft prompts (computed using the sentence transformer `all-MiniLM-L6-v2`) relative to ground-truth hard prompts.

As a baseline, we also compare our translators against InSPEcT (see 5.1). We note that while InSPEcT is training-free, it has an important hy-

perparameter: namely the pairs of layers used by the Patchscopes framework (Ghandeharioun et al., 2024) as part of InSPEcT. We found that InSPEcT is highly sensitive to this hyperparameter, and we thus optimized it on the training set of each DoD for all comparisons. See Appendix A.3 for details.

4.5 Evaluating Relevance and Utility of Verbalizations

Beyond measuring the similarity of the soft prompt's verbalizations to ground-truth hard prompts (Section 4.4), we also evaluated the task performance of verbalizations when used as a prompt by a larger, downstream LLM on new inputs. Specifically, we used the following methodology: (1) Train a soft prompt using a small, open-source LLM for a certain task; (2) Use the translator to verbalize the soft prompt into a hard prompt; (3) pass the verbalized hard prompt, along with an input example, to a large, closed-source LLM for inference on the same task (see Figure 1 for an illustration). If, using the verbalization as a prompt, the LLM performs at least as well as the original soft prompt, then that suggests the verbalizations are relevant to the task and useful for inference. It would also show that the verbalizations are portable hard prompts.

We apply this procedure to tasks within the testing set of the General DoD: Using the associated translator model, we verbalize a predicted hard prompt and pass it to a closed-source LLM (GPT-4o-mini) for inference using the prompt template:

```
{Verbalized prompt} Input: {input}
Output:
```

Control: To verify whether any observed improvement in task performance is due to the translated soft prompt being useful for the task, rather than just due to the larger LLM being inherently more proficient, we added a baseline condition that omits the verbalized prompt itself.

5 Results

5.1 Comparing Prompt Verbalization Quality

InSPEcT Datasets: We compared the verbalizations produced by the Classification DoD translator with the InSPEcT verbalizations on the 5 InSPEcT datasets. The translator outperformed InSPEcT in all datasets in terms of Recall and F1-Score; see Table 2. Qualitative examples of verbalizations produced by the translator and InSPEcT methods are presented in Table 3.

Dataset	Translator		InSPEcT	
	Recall	F1-Score	Recall	F1-Score
SST-2	**1.0**	**1.0**	**1.0**	**1.0**
SST-5	**1.0**	**0.71**	0.6	0.4
Subj	**1.0**	**0.8**	1.0	0.5
AG-News	**1.0**	**0.57**	1.0	0.44
TREC	**0.83**	**0.62**	0.16	0.18

Table 2: Translator vs. InSPEcT on InSPEcT datasets.

Notably, our translator performs substantially better, in terms of Recall, on SST-5 and TREC datasets where InSPEcT struggles due to a claimed dependency on high soft prompt task performance (Ramati et al., 2024). In contrast, the translator's verbalization quality seems more robust to weak soft prompt task accuracy. We further analyze the relationship between interpretability and task accuracy in Section 5.2 and Appendix A.9.

Classification DoD: As shown in Table 4 (top), the translator approach achieved an average verbalization F1-Score of 0.68 and an average Recall of 0.80. In contrast, we found that InSPEcT was completely unusable for this dataset, even after using the optimized layer pair `(27, 0)` on the training portion of the DoD (see Appendix A.3). Examples of verbalizations can be found in Table 5.

General DoD: As shown in Table 4 (bottom), the translator achieved 0.252 ROUGE-L and 0.427 cosine similarity, averaged over all the testing sets in this DoD. It again outperformed InSPEcT, which attained an average of 0.143 ROUGE-L and 0.269 cosine similarity, after finding the optimized layer pair `(22, 1)` on the training portion of the DoD. Verbalization examples are shown in Figure 7.

5.1.1 Size of Training DoD

By subselecting portions of the General DoD, we can estimate the trajectory of how increasing amounts of training data may yield a more accurate translator. In particular, we subselected 12%, 25%, 50%, and 100% of the General DoD; see Table 6.

As expected, training using the full dataset yields the best performance. The results suggest the translator could benefit from training on a larger DoD, though not hugely. Even in fine-tuning regimes, a training set size of only 539 samples is quite small. It is intriguing that in spite of a small training set, the translator can often find a rough correspondence between soft prompts and natural language.

Dataset & Ground-Truth	Translator (trained on Classification DoD)	InSPEcT (Ramati et al., 2024)
SST-2 *{positive, negative}*	"positive, positive, negative, positive, positive, negative, positive, negative, positive, positive"	"positive feedback, negative feedback | Identify the sentiment of a given text: positive, negative, neutral | . . . | Identify the sentiment of a given text: positive, negative,"
SST-5 *{terrible, bad, neutral, good, great}*	"good, great, neutral, bad, terrible, wonderful, excellent, awful, love, wonderful"	"is not a good writer. | Identify the sentiment of a text: positive, negative, neutral | . . . | Identify"
AG-News *{world, sports, business, technology}*	"technology, business, sports, world, contracts, agreements, deals, real, estate, area"	"from the world of business, the world of technology, the world of sports, the world of politics," . . .
Subj *{objective, subjective}*	"subjective, objective, neutral, subjective, subjective, subjective, ..."	":// Identify the tone of a text: formal, informal, objective, subjective The tone of the text is . . ."
TREC *{desc, entity, abbrev, human, loc, num}*	"description, human, entity, location, name, title, date, number, time, amount"	"ainer The final answer is: `$\boxed{6}$`assistant It seems like . . ."

Table 3: Verbalizations produced by our Classification DoD translator versus InSPEcT, based on soft prompts trained on InSPEcT datasets. Repetitive and long sequences are shown as (. . .).

DoD	Translator		InSPEcT	
	Recall	**F1-Score**	**Recall**	**F1-Score**
Classification	0.80	0.68	0	0
	ROUGE	**CosSim**	**ROUGE**	**CosSim**
General	0.252	0.427	0.143	0.269

Table 4: Translator vs. InSPEcT on Classification and General datasets of datasets (DoDs). Results are averaged over all constituent test datasets in each DoD.

5.1.2 Data Augmentation + Simplification

We also explored the effects of data augmentation on the ground-truth hard prompts. Subjectively, we found that the original hard prompts in the Super-NaturalInstructions dataset can be confusing and verbose (some instructions are 400+ words long). It is possible that using these as ground-truth for training the translator could hurt its performance, and thus we experimented with paraphrasing the original instructions to shorten and simplify them as a data augmentation measure. All paraphrased instructions are capped at 300 max tokens. We provide results of translators trained using augmented datasets 1, 10, and 100 times the original size of the training set below. See Appendix A.8.

Results are shown in Table 8 and suggest that, while augmentation of the DoD can improve translation quality benefits, augmenting too much can hurt performance (possibly by diluting the effect of the original hard prompts). Though, subjectively we found the verbalizations were qualitatively more readable. For all remaining experiments in the paper, we thus used the translator trained on the 10x augmented General DoD.

5.1.3 Out-of-Distribution Generalization

As a small out-of-distribution generalization test, we also trained two soft prompts on two real world tasks and analyzed the (General DoD) translator's verbalizations. Since there is no ground-truth hard prompt for these tasks, we are unable to quantitatively analyze them, but qualitatively the verbalizations (see Figure 2) suggest the translator is capable of generalizing. One such task consists of separating between "truthful" vs "deceptive" hotel reviews (Deceptive Spam Corpus) (Li et al., 2014). The other task was to distinguish between "AI generated" vs "Human written" essays (Kaushal, 2023).

Truthful vs. Deceptive Opinion Spam task

`Classify the review as "deceptive" if it is a fake review meant to deceive readers about its authenticity or the provider's quality. Otherwise, classify it as "truthful."`

Human vs. AI Essay Classification task

`Classify the given text as "Original" or "Generated". The text is "Original" if it was created by a human. The text is "Generated" if it was created by a machine`

Figure 2: Examples of translator generalizing out of distribution onto real world tasks.

5.2 Relevance and Utility of Verbalizations

To assess whether verbalized soft prompts are useful for the task or merely plausible sounding

Dataset & Classes	Our Translator Verbalization	InSPEcT Verbalization
Instruments *{drum, piano, violin, flute, guitar}*	"piano, violin, guitar, flute, drum, harp, cello, banjo"	"oga text: Identify the author of a given text: Shakespeare or Marlowe | . . . | Identify the author of a given text: Shakespeare or Marlowe | Identify"
Medical *{analgesic, antifungal, antibiotic, antiviral, antipyretic}*	"antiviral, antibiotic, analgesic, antifungal, antiseptic, antibiotic"	"://www.grammarlytes.com The text is a link to a website that provides a free online grammar . . ."

Table 5: Verbalizations produced using the translator vs. InSPEcT on soft prompts from the Classification DoD.

Training set	Cosine	ROUGE-L
12% of original	0.383	0.195
25% of original	0.399	0.247
50% of original	0.392	0.233
100% of original	**0.427**	**0.252**

Table 6: Performance of the generalized translator model on SuperNaturals instruction DoD. Reported metrics were averaged across 96 tasks in the test set.

prompts, we experiment with passing the verbalizations to a larger LLM for inference (as described in Section 4.5) on a set of input examples. We compare this to (1) giving the large LLM the same inputs *without* the verbalization ("baseline") as well as (2) giving the small LLM (SLM) the soft prompts themselves and with the input sentences.

Results (Figure 3) show that the translated prompt, as used by the larger LLM, outperforms both the original soft prompt itself, as well as the baseline LLM without the prompt. This suggests the verbalization of a soft prompt is also a viable—and portable—hard prompt usable for inference.

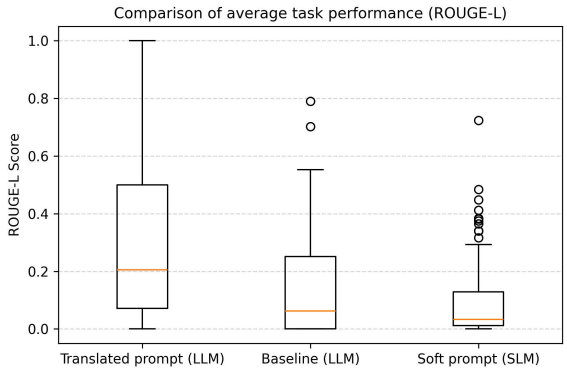

Figure 3: Comparison of translated hard prompts applied on LLM vs Baseline (no prompt) on LLM vs. soft prompt on SLM. Results are based on averaging task ROUGE-L across tasks in the General DoD testing split.

Comparison to Few-Shot Learning: Can translating soft prompts trained on a smaller model into human-readable prompts outperform in-context few-shot learning, whereby the LLM implicitly infers a pattern from the given examples to make a prediction? We investigate this by applying few-shot learning on the same General DoD tasks using the same average task ROUGE-L metric.

While for most of the tasks the few-shot learning approach was more successful in terms of task accuracy (ROUGE-L), we did find a set of 10 tasks (out of 96) where the prompt that was discovered through soft prompt translation outperformed few-shot learning. Many of these tasks were often more obscure and difficult to infer just by looking at a few examples alone. We provide a qualitative example in Fig. 4 of one such task.

Ground truth hard prompt
`"You are given a sentence from a conversation between a human and a virtual assistant. Your task is to identify whether the sentence is a question or not. Answer with Yes or No."`
Prediction from translator
`"Classify the given query as either "yes" if it is a question or "no" if it is a statement."`
Examples
`Input:` "Which city should I search?" `Output:` "Yes"
`Input:` "Sorry your reservation could not be done. Can I help you in any other thing?" `Output:` "No"
Performance

Translator's Prompt on LLM Task ROUGE-L	0.80
Few-shot Learning on LLM Task ROUGE-L	0.40
Soft prompt on SLM Task ROUGE-L	0.01

Figure 4: Qualitative example of an instance where fewshot learning and soft prompting is outperformed by the soft prompt translator's verbalized hard prompt.

We speculate that these 10 tasks represents a class of problems where there are many hypotheses that could explain the input-output relationship between a sentence and its label, thus a large number of examples is needed to identify the true hypothesis. For soft prompt training, this is no problem

Ground-Truth Prompt	Translator (trained on General DoD)	InSPEcT (Ramati et al., 2024)
task1391: "In this task, you are given a question containing a blank (_) and two options. You should pick the best option to answer the question. Please answer with "A" or "B"."	"Identify the correct answer from the given options using the provided context. Return the answer index: "A" or "B"."	"ipel, use the same words to create a new answer. | Instruction: Use the same words to create a new answer. | ... | Instruction: Use the same words to create"
task392: "In this task, you will be given two sentences separated by "because". You should decide whether the first sentence can be the result of the second sentence. If you can see a possible cause and effect relationship, answer with "plausible", otherwise answer with "not plausible".	"Determine if the explanation for the given plot hole or inconsistency is plausible. If it makes sense and fits the context, label it as "plausible". Otherwise, label it as "not plausible". The explanation must address the specific plot hole or inconsistency."	", Instruction: Use the first letter of the input to generate a word. The word should be a noun. | Instruction: Use the first letter of the input to generate a word. The word should be a noun. | Instruction: Use the first letter of the input to generate a word. The"

Table 7: Qualitative examples from General DoD, showing the translator's and InSPEcT's verbalizations of soft prompts for two tasks (task1391 and task392) from SuperNatural DoD.

Training set	Cosine Similarity	ROUGE-L
$\text{Aug}_{1\times}$	0.394	0.165
$\text{Aug}_{10\times}$	**0.410**	**0.174**
$\text{Aug}_{100\times}$	0.384	0.135

Table 8: Effect of augmentation on verbalization quality

(since it can be trained on many examples), but this is very inefficient for few-shot learning.

6 Discussion

Over all our experiments, we found that the dedicated translation models consistently outperformed the training-free InSPEcT technique on the InSPEcT datasets as well as the Classification and General DoDs in terms of quantitative verbalization quality. The translator also showed significant ability to generalize out-of-distribution. Qualitatively, the translator's verbalizations were mixed: some completely captured the goal of the task, whereas others missed the mark (often by a wide margin).

The performance of the translator model, especially its generalization capability, is surprising given the small training set size (539 samples). We speculate a possible explanation: the Rosetta Stone contained only a few thousand words in equivalent English translations, yet it was sufficient to establish a correspondence between ancient Greek and ancient Egyptian. Suppose that soft prompts constitute a language in their own right. It is possible that LLMs already innately understand this language, and obviously they can understand and express natural language. The main difficulty in translating from soft to hard prompts is thus in finding a *correspondence* between these two languages, and perhaps a small dataset of (*soft prompt*, *hard prompt*) pairs is sufficient. This contrasts with, say, the LLM being fine-tuned to translate between two entirely new languages it was never trained on.

Finally, the workflow we tested (far right side of Figure 1), whereby verbalized soft prompts can be passed to a large, more powerful LLM for inference, showed promise. It has an important advantage over in-context few-shot learning that the soft prompt can be trained on a large dataset, whereas few-shot learning is generally limited to a small number of inputs due to token limits.

7 Conclusion and Future Work

Soft prompts are opaque and difficult to interpret. We show that by training a dedicated soft prompt translator model, it is possible to extract a human readable, natural language translation of these soft prompts that is more accurate than the prior training-free InSPEcT approach. Verbalizations obtained from the translator are useful and relevant to the task at hand, which also potentially opens up a new workflow for obtaining portable prompts that can be applied onto closed source models.

With a more powerful soft prompt translator trained on a larger dataset, it may be possible to train soft prompts on tasks with the specific goal of later translating them to learn new knowledge. It would be exciting if, after training a soft prompt for a complex task—like drug discovery (Liu et al., 2026)—insights into the mechanism of how the LLM makes its predictions (e.g., the molecular interactions) could be expressed in natural language.

8 Limitations

We did not explore in this work whether our soft prompt translator model can work with soft prompts trained on a different base model, though given that soft prompts are not normally transferrable between models, it is safe to assume this is not the case. This means unless the soft prompt is already trained using the same base model as ours (Llama-3.1-8B-Instruct), a new translator model would need to be trained.

While testing the soft prompt translation model out of distribution on real-world tasks, we noticed that while verbalizations were truthful and relevant to the task, it was often the case that they were also terse and lacked relevant details. We believe this is an issue with stylistic alignment—where the translator is erroneously choosing to omit relevant task details to produce more fluent verbalizations. See Appendix A.12 for further details.

As is, soft prompt translation is a promising proof of concept. In order for it to become a practical tool, however, a larger dataset is needed along with dedicated stylistic alignment (i.e.: preference optimization or Supervised-Fine Tuning on real world datasets) to ensure verbose, informative verbalizations.

Acknowledgments

Parts of this work utilized AI for coding assistance for miscellaneous tasks such as visualizations and implementations of certain methods, etc. All implementations were checked and tested by humans. No AI assistance was used for the writing of the paper itself. No AI assistance was used for any ideation.

References

Eshaan Agarwal, Joykirat Singh, Vivek Dani, Raghav Magazine, Tanuja Ganu, and Akshay Nambi. 2024. Promptwizard: Task-aware prompt optimization framework. *Preprint*, arXiv:2405.18369.

Luke Bailey, Gustaf Ahdritz, Anat Kleiman, Siddharth Swaroop, Finale Doshi-Velez, and Weiwei Pan. 2023. Soft prompting might be a bug, not a feature. In *Workshop on Deployable Generative AI at the International Conference on Machine Learning*. OpenReview preprint.

Leonard Bereska and Efstratios Gavves. 2024. Mechanistic interpretability for ai safety–a review. *arXiv preprint arXiv:2404.14082*.

Valentin Leonhard Buchner, Lele Cao, Jan-Christoph Kalo, and Vilhelm von Ehrenheim. 2024. Prompt tuned embedding classification for multi-label industry sector allocation. *Preprint*, arXiv:2309.12075.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric P. Xing, and Zhiting Hu. 2022. Rlprompt: Optimizing discrete text prompts with reinforcement learning. *Preprint*, arXiv:2205.12548.

W. N. Francis and H. Kucera. 1979. Brown corpus manual. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island, US.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. 2024. Patchscopes: A unifying framework for inspecting hidden representations of language models. *arXiv preprint arXiv:2401.06102*.

Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. 2021. Gradient-based adversarial attacks against text transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5747–5757, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Navjot Kaushal. 2023. Human vs ai generated essays. https://www.kaggle.com/datasets/navjotkaushal/human-vs-ai-generated-essays. Kaggle dataset. Accessed: 2026-05-22.

Daniel Khashabi, Xinxi Lyu, Sewon Min, Lianhui Qin, Kyle Richardson, Sean Welleck, Hannaneh Hajishirzi, Tushar Khot, Ashish Sabharwal, Sameer Singh, and Yejin Choi. 2022. Prompt waywardness: The curious case of discretized interpretation of continuous prompts. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3631–3643, Seattle, United States. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. *CoRR*, abs/2104.08691.

Jiwei Li, Myle Ott, Claire Cardie, and Eduard Hovy. 2014. Towards a general rule for identifying deceptive opinion spam. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1566–1576, Baltimore, Maryland. Association for Computational Linguistics.

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.

Jack Lindsey. 2025. Emergent introspective awareness in large language models. *Transformer Circuits Thread*.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 61–68, Dublin, Ireland. Association for Computational Linguistics.

Yupeng Liu, Han Zhang, Rui Hu, Hui Zhang, and Xiaochen Zhang. 2026. Multi-task molecular representation learning based on soft prompting of the important subgraph. *Mol. Genet. Genomics*, 301(1).

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 271–278, Barcelona, Spain.

Oam Patel, Jason Wang, Nikhil Shivakumar Nayak, Suraj Srinivas, and Himabindu Lakkaraju. 2025. Towards interpretable soft prompts. *Preprint*, arXiv:2504.02144.

Cheng Peng, Xi Yang, Kaleb E Smith, Zehao Yu, Aokun Chen, Jiang Bian, and Yonghui Wu. 2024. Model tuning or prompt tuning? a study of large language models for clinical concept and relation extraction. *Journal of Biomedical Informatics*, 153:104630.

Aleksandar Petrov, Philip H. S. Torr, and Adel Bibi. 2024. Prompting a pretrained transformer can be a universal approximator. *Preprint*, arXiv:2402.14753.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer. *Preprint*, arXiv:1910.10683.

Dana Ramati, Daniela Gottesman, and Mor Geva. 2024. Eliciting textual descriptions from representations of continuous prompts. *Preprint*, arXiv:2410.11660.

Weijia Shi, Xiaochuang Han, Hila Gonen, Ari Holtzman, Yulia Tsvetkov, and Luke Zettlemoyer. 2022. Toward human readable prompt tuning: Kubrick's the shining is a good movie, and a good prompt too? *Preprint*, arXiv:2212.10539.

Parshin Shojaee, Kazem Meidani, Shashank Gupta, Amir Barati Farimani, and Chandan K. Reddy. 2025. LLM-SR: Scientific equation discovery via programming with large language models. In *The Thirteenth International Conference on Learning Representations*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Ellen M. Voorhees and Dawn M. Tice. 2000. Building a question answering test collection. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, page 200–207, New York, NY, USA. Association for Computing Machinery.

Tu Vu, Aditya Barua, Brian Lester, Daniel Cer, Mohit Iyyer, and Noah Constant. 2022. Overcoming catastrophic forgetting in zero-shot cross-lingual generation. *Preprint*, arXiv:2205.12647.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Anjana Arunkumar, Arjun Ashok, Arut Selvan Dhanasekaran, Atharva Naik, David Stap, and 1 others. 2022. Super-naturalinstructions:generalization via declarative instructions on 1600+ tasks. In *EMNLP*.

Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2023. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. *Preprint*, arXiv:2302.03668.

Yao Xiao, Lu Xu, Jiaxi Li, Wei Lu, and Xiaoli Li. 2023. Decomposed prompt tuning via low-rank reparameterization. *Preprint*, arXiv:2310.10094.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NIPS*.

Mengjie Zhao and Hinrich Schütze. 2021. Discrete and soft prompting for multilingual models. *CoRR*, abs/2109.03630.

A Additional details

A.1 Code Repository Link

All code related to our experiments can be found at the following remote repository link: `https://anonymous.4open.science/r/softprompt_experiments-B072`.

A.2 Synthetic dataset creation details

For Classification-DoD generation, we utilized two separate vLLM pipelines for related keyword generation and sentence generation.

Keyword Generation: To choose keywords for a mini-dataset, we used a vLLM pipeline which sampled 3 random (nouns or adjectives) from the

Brown Corpus (Francis and Kucera, 1979), and primed an LLM with the sampled random keywords as "abstract inspiration" to generate distinct words as classes which are related by concepts, industries, themes, etc. For each dataset, we generated 5 keywords as the dataset's classes. We used `mistralai/Mistral-Small-3.1-24B-Instruct-2503` as the LLM for keyword generation.

Sentence Generation: Using chosen keywords per mini-dataset, we generated 100 sentences per keyword (500 sentences per dataset), using `mistralai/Mistral-Small-3.1-24B-Instruct-2503`. This dataset generation pipeline uses the vLLM library to efficiently batch requests together for higher generation throughput. Each mini-dataset was stored in a single SQLite3 database file, in a relational database manner on 3 tables, for fast and easy retrieval during the soft prompt training phase. For each mini-dataset, we store the generated sentences with their ground-truth class label, which was used to generate the sentences. Due to the constraint that the classes selected must be related to each other, we had instances where some mini-datasets contained classes which are very difficult to differentiate such as: `revive`, `stimulate`, `inspire`, `encourage`, `motivate`, which are generated from using the abstract word `energize`. Due to this, we observed many soft prompts with task accuracy below 50%.

A.3 InSPEcT experiment details

We trained soft prompts of 20 tokens in length on the SST-2, SST-5, AG-News, Subj, and TREC datasets, according to hyperparameter configurations mentioned in Table 4 of the InSPEcT Paper (Ramati et al., 2024). We utilized the `meta-llama/Meta-Llama-3.1-8B-Instruct` model as the base LLM for soft prompt training. For the patching experiments, we utilized the `meta-llama/Meta-Llama-3.1-8B-Instruct` model, which has 32 decoder layers. For determining a specific layer pair hyperparameter: `configuration (source layer, target layer)`, there are 1024 combinations (32 * 32, which excludes the last decoder layer and includes the embedding layer). The existing code for InSPEcT paper uses limited layer pair combinations `(source layer, target layer)`, though our experiments used all layer pair combinations, as the InSPEcT paper / code does not mention the "golden" layer pairs to use.

We also applied InSPEcT to our Classification DoD and General DoD, by optimizing for the best layer pair hyperparameter on the training split of each DoD. For this, we performed a grid search over all 1024 layer pair combinations and computed an evaluation metric. We then picked the layer pair which provided the best evaluation metric average value across all mini-datasets in the DoD. For optimization, we used following evaluation metrics: average of F1-Score and Class Rate for Classification DoD and cosine similarity for General DoD.

Our implementation of InSPEcT is forked from the original InSPEcT GitHub repository: https://github.com/danaramati1/InSPEcT . For InSPEcT datasets, we used the following versions on Hugging Face:

1. **SST-2**: `stanfordnlp/sst2`
2. **SST-5**: `SetFit/sst5`
3. **AG-News**: `fancyzhx/ag_news`
4. **Subj**: `SetFit/subj`
5. **TREC**: `SetFit/TREC-QC`

A.4 Soft Prompt Training Implementation Details

We used the PEFT library for training soft prompts for InSPEcT, whereas we used our own custom implementation for training soft prompts on DoDs. We also reversed the implementations for each method (for example: trained InSPEcT soft prompts using custom implementation) and got worse results for both. We therefore, report the best results above for each method, across all implementations.

A.5 Soft prompt Translator training details

All methods described in this section use soft prompt embeddings as the input sequence and hard prompts as the output sequence for training using Maximum Likelihood Estimation. As described in our paper, we model soft prompt translation as a sequence to sequence, text-generation task where we pass in soft prompt embeddings as input and calculate supervised, next-token loss over the target hard prompt output sequence. We fine-tune `meta-llama/Llama-3.1-8B-Instruct` using LoRA (Hu et al., 2021). For LoRA implementation, we used the PEFT library with the following LoRA configuration:

```
lora_config = LoraConfig(
    r = LORA_RANK,
    lora_alpha = 2 * LORA_RANK,
    target_modules = ["q_proj", "v_proj", "k_proj", "o_proj", "gate_proj", "up_proj", "down_proj"],
    lora_dropout = LORA_DROPOUT,
    bias = "none",
    task_type = TaskType.CAUSAL_LM
)
```

Listing 1: LoRA configuration used with the PEFT library.

A.6 Preliminary Classification DoD vs Classification DoD

An earlier iteration of our Classification DoD — Preliminary Classification DoD —- lacked semantic relationship constraint between keywords within a mini-dataset. In terms of soft prompts trained on InSPEcT datasets, translator trained on Classification DoD outperformed translator trained on Preliminary Classification DoD, not only in terms of Class Rate and F1-score, but also in terms of quality of the verbalizations produced. This can be attributed to the constraint placed on the selection of keywords within each mini dataset in Classification DoD (which we discussed in section A.2), which allowed the mini datasets, and hence the respective trained soft prompts, to be more real-world-like. The comparison of Classification DoD translator and Preliminary Classification DoD can be found in table 9. Hence for further experiments, we use Classification DoD translator.

A.7 Preprocessing Super-NaturalInstructions Dataset

Since the original Super-NaturalInstructions Dataset is in a GitHub repository (https://github.com/allenai/natural-instructions), comprising of multiple JSON files, we decided to parse this into an easy-to-use dataset on Hugging Face. We performed some data analysis and decided to further filter the dataset for training soft prompts and translators on it. Based on our data analysis, we performed the following preprocessing:

1. Filter out any instance sequences (input + output) longer than 400 tokens, as we found that there were certain instances where the sequence length was over 200k tokens, with P99 of 1312 and P90 of 337. To keep at least 90% of the instance data, we picked 400 tokens as the threshold for this filtration.

2. Filter out any tasks with less than 500 instances, as we found that some tasks had a minimum of 29 instances only, which would not be sufficient for training soft prompts. Hence, we decided to use 500 instances as a minimum threshold for selecting tasks for future experiments.

3. Added a column for total tokens of input + output sequences, calculated using tokenizer of `meta-llama/Llama-3.1-8B-Instruct` language model, as we dynamically use it later during soft prompt training, for data collation.

4. Created instances of (input, output[i]) pairs (unwinding), if the output is an array of multiple values for a given input, for all values of i in the output array.

In the end, we were left with 539 tasks for training split and 96 tasks for testing split.

A.8 Data Augmentation on General DoD

In order to test our hypothesis of whether the translator quality improves with more training data, we decided to perform data augmentation to the training split of General DoD. We used a vLLM pipeline, using the `mistralai/Mistral-Small-3.1-24B-Instruct-2503` LLM, to generate multiple paraphrases of the original instructions using the following system prompt:

```
system_prompt = (
    "You are an expert at simplifying and extremely condensing instructions."
    "Your task is to paraphrase the following instruction into a highly concise statement."
    "CRITICAL: You MUST explicitly preserve all specific classes, exact tags, labels, output formats, and special syntax constraints. "
    "CRITICAL: Do NOT oversimplify or remove specific mappings between concepts (e.g., specifying which sentence is the premise, exact sentence counts, positional logic, or structural relationships). "
    "Use a maximum of 3 to 5 short sentences. Strip away ONLY filler words, long narrative examples, and conversational redundant explanations. "
    "Output ONLY the paraphrased instruction text and absolutely nothing else. Do NOT include phrases like 'Here is the paraphrased version'."
```

Dataset	Class Rate (Preliminary Classification DoD translator)	F1-Score (Preliminary Classification-DoD translator)	Class Rate (Classification DoD translator)	F1-Score (Classification DoD translator)
SST-2	1.0	0.33	**1.0**	**1.0**
SST-5	0.4	0.4	**1.0**	**0.71**
AG-News	0.75	0.46	**1.0**	**0.57**
Subj	1.0	0.57	**1.0**	**0.8**
TREC	0.33	0.27	**0.83**	**0.62**

Table 9: Comparison of Classification DoD translator and Preliminary Classification DoD translator on soft prompts trained on the InSPEcT datasets

```
)
```

Listing 2: System Prompt for generating paraphrases with reduced number of tokens

We decided to prompt the LLM here to generate condensed paraphrases, since we wanted to save training time during translator training, as these paraphrases will be used as "hard prompt". We generated two new training splits: (1) with 10 times augmentation applied and (2) with 100 times augmentation applied. These augmented training splits are used in the results section 5.1.

A.9 Correlation between Interpretability and Task Accuracy

While testing Classification DoD translator on the testing split of Classification DoD, we observed an interesting phenomenon: the accuracy of the soft prompts does not need to be high in order to get good verbalizations out of them. For example, most verbalizations of the soft prompts from the testing split of Classification DoD are commendable in terms of Recall and F1-score, despite having low task performance (< 50%). This is contrary to the idea proposed in the InSPEcT paper, where they claim a correlation between quality of verbalizations and the task accuracy of the soft prompts. Some verbalizations for soft prompts with low task accuracy can be found in table 10.

We hypothesize that while the soft prompt applied on the LLM may achieve poor performance, this might not just be because they encode irrelevant task instructions; another valid possibility is that the soft prompt does encode relevant task instructions, but rather our small base model of only 8 billion parameters may be failing to execute them.

This hypothesis is consistent with our finding in the General DoD: we showed that extracting the instructions encoded within the soft prompt using the translator and passing these instructions to a larger, more capable LLM results in better performance—indicating that the soft prompts encode relevant task instructions but the smaller base model partially failed to execute them. Taking these soft prompts and verbalizing them with the translator, we can pass their encoded instructions to a larger, more intelligent LLM for inference on the same task. If the soft prompts do encode correct instructions for a given task, we could then expect that a more intelligent LLM would be able to carry it out. Indeed, this was shown to be the case—in one instance (See: Fig. 4), while the original soft prompt had a task ROUGE-L of only 0.012, verbalizing the soft prompt and passing the instructions to an LLM achieves a task ROUGE-L of 0.8, surpassing even the performance of few-shot learning on the same LLM which achieved a task ROUGE-L of 0.4.

A.10 Task Differentiability Analysis

We also evaluate verbalizations of the General DoD Translator using the Mean Percentile Rank (MPR) of cosine similarities. Although ROUGE-L and cosine similarity capture absolute textual and semantic alignment, MPR provides a relative measure of task discriminability. To calculate MPR, each predicted verbalization's cosine similarity is scored against all ground-truth prompts' similarity (including its true paired hard prompt and all other unrelated hard prompts) in the evaluation set. We then compute its percentile rank and average this across all predicted hard prompts, which evaluates how well the translator distinguishes unique tasks based on verbalized soft prompts alone.

Hard Prompt	Verbalized Soft Prompt	Soft Prompt Accuracy	Recall	F1-score
anchored, permanent, fixed, immovable, stationary	stationary, fixed, anchored, permanent, immovable, sailing, moving, floating, transient	51.67%	1.0	0.71
abundant, profitable, bountiful, prosperous, plentiful	abundant, plentiful, fruitful, bountiful, prosperous, rich, successful, booming	75.0%	0.8	0.62
revival, rejuvenation, restoration, recuperation, regeneration	restoration, regeneration, revival, rejuvenation, restoration, recuperation, revival, regeneration,	57.45%	1.0	1.0
pity, empathy, compassion, mercy, sorrow	compassion, sorrow, mercy, pity, empathy	41%	1.0	1.0
gps, lidar, braille, sonar, radar	model, google, braille, echolocation, sonar	93.13%	0.6	0.50

Table 10: Qualitative examples of verbalizations produced using Classification-DoD-translator on the soft prompts from the testing split of Classification-DoD.

A.11 Augmentation Scaling Experiments

To investigate the scaling behavior of the augmentation training set, we also experimented with varying the number of training samples used in each augmentation regime (1x, 10x, 100x times original train set size). They are generally consistent with the hypothesis that the translator benefits from additional data. Notably, the augmentation made a considerable difference in terms of task discriminability via MPR while ROUGE-L and cosine similarity seems to have suffered, as seen in Table 11.

Training set	MPR	Cosine	ROUGE-L
12% of original	0.657	0.383	0.195
25% of original	0.644	0.399	0.247
50% of original	0.688	0.392	0.233
100% of original	0.704	0.427	0.252
12% of $aug_{1\times}$	0.674	0.338	0.152
25% of $aug_{1\times}$	0.713	0.372	0.160
50% of $aug_{1\times}$	0.740	0.387	0.161
100% of $aug_{1\times}$	0.749	0.394	0.165
12% of $aug_{10\times}$	0.763	0.385	0.164
25% of $aug_{10\times}$	0.752	0.402	**0.177**
50% of $aug_{10\times}$	**0.772**	**0.411**	0.176
100% of $aug_{10\times}$	0.768	0.410	0.174
12% of $aug_{100\times}$	0.739	0.354	0.129
25% of $aug_{100\times}$	0.755	0.366	0.129
50% of $aug_{100\times}$	0.759	0.371	0.130
100% of $aug_{100\times}$	0.768	0.384	0.135

Table 11: Effect of data augmentation on MPR, cosine similarity, and ROUGE-L across hold out set tasks.

A.12 Stylistic Alignment

A soft prompt is a high dimensional object holding many details, some of which are trivial like expected formatting. In order to produce natural-language-like verbalizations, the translator must decide which details need to be omitted. In practice, we found that there were many cases where verbalizations were too concise than ideal (for instance in figure 2). This is where style alignment can be applied to ensure that important details are not omitted.

As a rudimentary method of encouraging a more verbose generation style, we experiment with prefilling text. (i.e.: start autoregressive generation pre-conditioned on [soft tokens; prefill text]). Often, after applying a prefill text, we see additional factual details in verbalizations. We show an example of one such case below in Fig 5.

In this anecdotal example, the translator model omitted relevant task details in its original verbalization, which is only revealed once prefill is applied. What this suggests is that, in such instances, the translator correctly comprehends the semantics of the soft prompt, but incorrectly omitted crucial details as a stylistic choice. This is a clear example of a stylistic misalignment which dedicated preference optimization could potentially address.

To verify that this is not anecdotal, we also investigated the effects of prefill conditioning more rigorously on 550 classification datasets from Classification DoD. Across 550 classification datasets, the average class rate of verbalizations doubled after applying the prefill.

Without Prefill

`Calculate the value of the given equation using the provided formula and variables. Output the result. Use the exact formula and variables as specified.`

With Prefill

`[First, I should] calculate the factored form of the quadratic equation. The equation is given as '`$ax^2+bx+c=0$`'. The factored form is '`$a(x-r_1)(x-r_2)=0$`', where` r_1 `and` r_2 `are the roots of the equation. Second, I should identify the value of the variable '`x`' in the given problem. Third, I should substitute the value of '`x`' into the equation and solve for the answer using the factored form.`

Figure 5: Example showing the effect of prefill conditioning ("First, I should") on verbalizations, in particular In this case a regression task involving quadratics

This was found by applying our general translator model out of distribution onto the Classification DoD. Without prefill conditioning, the general translator model had an average class rate of 1.16% across 550 classification tasks, which is quite poor. We then repeated the same experiment, applying a simple prefill ("Classify the given input into one of the following:"), in which our class rate went up to 2.62%. This is consistent with our speculation that the translator may omit certain details due to stylistic misalignment, suggesting potentially significant improvements through preference optimization.