ICL Markup: Structuring In-Context Learning using Soft-Token Tags

Marc-Etienne Brunet
University of Toronto
Vector Institute
mebrunet@cs.toronto.edu

Ashton Anderson
University of Toronto
Vector Institute
ashton@cs.toronto.edu

Richard Zemel
University of Toronto
Columbia University
Vector Institute
zemel@cs.toronto.edu

Abstract

Large pretrained language models (PLMs) can be rapidly adapted to a wide variety of tasks via a text-to-text approach, where the instruction and input are fed to the model in natural language. Combined with in-context learning (ICL), this paradigm is impressively flexible and powerful. However, it also burdens engineers with an overwhelming amount of choices, many of them arbitrary. Inspired by markup languages like HTML, we contribute a method of using soft-token (a.k.a tunable token) tags to compose prompt templates. This approach reduces arbitrary decisions and streamlines the application of ICL. Our method is a form of meta-learning for ICL; it learns these tags in advance during a parameter-efficient fine-tuning "warm-up" process. The tags can subsequently be used in templates for ICL on new, unseen tasks without any additional fine-tuning. Our experiments with this approach yield promising initial results, improving PLM performance in important enterprise applications such as few-shot and open-world intent detection, as well as text classification in news and legal domains.

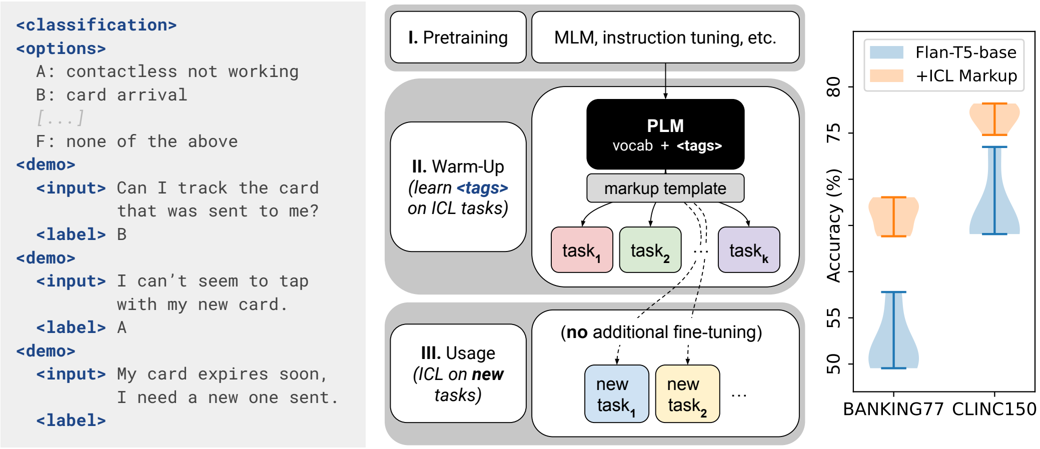
1 Introduction

With the growing size and capabilities of large pretrained language models (PLMs), in-context learning (ICL) has become a popular way to harness their power for new tasks. ICL is an approach to prompting PLMs which includes demonstrations of how to complete the target task in the prompt (Dong et al., 2022). It has significant advantages over traditional fine-tuning, being data-efficient, highly flexible, and user-friendly. A PLM can be adapted to perform effectively on a new task with only a handful of demonstrations (few-shot) and some natural language instructions. This can be done quickly even by someone with little knowledge of machine learning. The PLM can also be encapsulated as a black box and shared across tasks. This allows individuals and organizations to leverage PLMs for new tasks, even if they do not have the computing resources necessary to fine-tune (or even host) such large models.

However, ICL also has several disadvantages. Most PLMs have not been explicitly trained or tuned to perform ICL, and thus have not actually been optimized to approach new tasks in this format (Dong et al., 2022). Like other forms of prompt engineering, ICL suffers from a lack of robustness across the many arbitrary choices that users encounter in the process of setting it up (Chen et al., 2022). There is also evidence to suggest that ICL performs poorly when shown a "none of the above" option (Kadavath et al., 2022), which could hinder its application in practical settings (e.g. open world classification) where the inputs may not always correspond with any option in the label space.

We propose addressing these shortcomings with an approach to ICL inspired by markup languages like HTML. In this paradigm, we structure ICL prompt templates using a dedicated set of soft-token tags that we add to the model's vocabulary. These soft-tokens (a.k.a. tunable tokens) are effectively

R0-FoMo: Workshop on Robustness of Few-shot and Zero-shot Learning in Foundation Models at NeurIPS 2023.



```
<classification>
<options>
  A: contactless not working
  B: card arrival
  [...]
  F: none of the above
<demo>
  <input> Can I track the card
          that was sent to me?
  <label> B
<demo>
  <input> I can't seem to tap
          with my new card.
  <label> A
<demo>
  <input> My card expires soon,
          I need a new one sent.
  <label>
```

Figure 1: (left) Example of an ICL Markup template applied to intent detection. Blue boldface **<tags>** indicate dedicated soft-tokens introduced into the PLM's vocabulary. (center) The soft-token tags are learned in advance during a "warm-up" phase (parameter-efficient fine-tuning). They can then be used on new, unseen tasks without additional fine-tuning. (right) Initial experiments show accuracy improvements over hard-token ICL templates.

"new words": bound to trainable parameters and processed like other tokens. Their weights are learned in advance during a "warm-up" stage (parameter-efficient fine-tuning). They can then be used in the ICL template for new tasks without additional fine-tuning, and can thus also be shared across tasks. The training process is therefore a form of meta-learning for ICL. We show that this approach removes several arbitrary decisions from the design of ICL prompt templates. We also provide initial empirical evidence that it can improve a model's ICL ability on new tasks. Specifically, we show that ICL Markup improves Flan-T5 models on text classification tasks (intent detection, news and legal domains). We show that ICL Markup can reduce Flan-T5's performance variability when compared to prompt engineering, increase classification accuracy, as well as improve out of scope intent detection when the template includes a "none of the above" multiple-choice option.

2 Background and Related Work

In-Context Learning One can think of ICL as the PLM learning the target task "by analogy" from the examples in the prompt (Dong et al., 2022). ICL has also been referred to as prompt augmentation (Liu et al., 2023), since a cloze or completion-style prompt is augmented by prepending answered prompts. The effectiveness of ICL has sparked a great deal of research on the topic. The areas most related to this work include pretraining methods for ICL, for example, MetaICL (Min et al., 2021), In-Context Tuning (Chen et al.), and Symbol Tuning (Wei et al., 2023). Our work differentiates itself by introducing dedicated soft-token tags, and performing the meta-learning in a parameter-efficient way. ICL research also investigates methods for demonstration selection, such as retriever systems that identify the best demonstrations for a particular input from a pool of candidates (Liu et al., 2022). As well as work targeting ICL robustness (or lack thereof) due to choices in prompt template designs (Zhao et al., 2021), or demonstration order (Lu et al., 2021).

Parameter-Efficient Fine-Tuning Prior to the rise of prompt engineering (including ICL), the principal approach to adapting a pretrained model to a downstream task was via fine-tuning (Liu et al., 2023). However, it is not computationally feasible for most individuals or even organizations to fine-tune modern PLMs; there are simply too many parameters (Ding et al., 2023). As result, several parameter-efficient approaches have been developed which enable fine-tuning PLMs using fewer trainable parameters (Lester et al.; Lisa Li and Liang; Hu et al., 2021).

Other approaches to prompt engineering The work by Gu et al. on pretrained prompt tuning (PPT) is perhaps the most related to ours. They pretrain soft-token prompts in a self-supervised manner, then fine-tune them per few-shot task. This improves the reliablity of few-shot prompt tuning. Our work is different because it aims to avoid the second fine-tuning step. Our tokens and templates are designed to be used for ICL on new tasks without further fine-tuning, relying on demonstrations to assist with adaptation. Prompt tuning with rules (Han et al., 2022) involves manually decomposing a task into sub-prompts, then fine-tuning a PLM to optimize the performance of this decomposition.

Few-shot and Open-world Intent Detection Virtual Assistant (VA) and dialogue systems are deployed in many enterprise use cases. A single enterprise platform provider may host over 100,000 customer-specific models (Qian et al., 2023). The intent detection models in these VA systems need to be flexible (adapting to an evolving label space), quick to train (minute-scale), configurable by non-machine learning experts, capable of handling highly multi-class, few-shot, and imbalanced datasets, and able to recognize out-of-scope intents. There has been considerable research into few-shot and open-world intent detection detection. Several works have studied the use of PLMs to augment few-shot datsets with synthetic examples (Lin et al.; Sahu et al., 2022). Others explore methods to identify out-of-scope (OOS) examples (Khosla and Gangadharaiah, 2022; Zhang et al., 2020b; Qian et al., 2023), i.e., inputs having an intent that falls outside of the model's configured label space. However, Zhang et al. (2021) note that OOS detection is considerably more challenging with in-domain OOS (ID-OOS) examples which are semantically related to the in-scope intent classes. They construct datasets in order to examine the robustness of pretrained transformers in this challenging setting.

3 Proposed Method: ICL Markup

We propose using a markup-like language to construct ICL templates in order to reduce the number of arbitrary choices involved and thereby enable easier and more consistent application. Our paradigm is visualized in Figure 1 (left). With this approach, we separate content and form using soft-token tags. For example, rather than including a demonstration in the prompt as `statement:I can't tap my card. class:contactless not working`, it is included as `<input> I can't tap my card. <label> contactless not working`, where `<input>` and `<label>` are soft-tokens that have been learned in advance to indicate the inclusion of a labeled (X, y) pair. This removes many arbitrary choices about presentation, e.g. whether to demarcate the demonstration using "statement:", "input:", "label:" or "category:". Engineers can then focus their energy on the content of the prompt, such as the choice of demonstrations and class descriptors. Such an approach maintains flexibility while reducing arbitrary choices.

There are many possible ways to structure these markup-like tags. Here we explore one option which targets multiple-choice style classification templates and is applicable to a broad set of tasks. It uses the following tags:

`<classification>` instructs the model that the following input is a classification task

`<options>` demarcates the start of YAML-formatted multiple choice options; defines a mapping from capital letter tokens to class descriptors

`<demo>` denotes the start of a labelled example

`<input>` indicates the start of the example's textual input

`<label>` indicates the multiple-choice letter option corresponding to the correct class descriptor

We add these soft-token tags to the vocabulary of a PLM and let the model learn them within our defined template structure during a "warm-up" process. This is done by composing one or several templates with the tags, then using the templates as prompts to solve ICL classification tasks. This is a parameter-efficient fine-tuning process in the sense that only the parameters of the tags are updated. The tags can then be used to perform ICL on new tasks without further fine-tuning. We note that a tag need not correspond to a single soft-token, but could be an ordered set of soft-tokens. The size of the sets is chosen so that the tags house the representational capacity required for the model to use them.

4 Experiments

We use Flan-T5 models for our experiments (Raffel et al., 2020; Chung et al., 2022). We train ICL Markup tags for three model sizes: base (250M parameters), large (780M), and XL (3B). Throughout all experiments, we use the same form of multiple-choice style ICL Markup template: having a variable number of answer options, followed by a variable number of (labelled) demonstrations. See Figure 1 (left) for an example. We use an ordered set of between 9 and 18 soft-tokens to represent the <classification> tag, an ordered set of 2 soft-tokens for <options>, and 1 soft-token for each of <demo>, <input>, and <label>. With the exception of our open-world evaluations, we use an unconstrained greedy decoding to generate the multiple choice response.[1]

Relationship between training and target tasks We focus on the use of ICL Markup in few-shot text classification. We assume only a limited number of labelled examples are available at inference for adaptation to the target task (which is done *through ICL only, not parameter updates*). However, we assume access to labelled data from related tasks can be used to learn the ICL Markup tags in advance. These training tasks may differ (shift) from the target task in a few ways. We consider three types of shifts in our experiments. *I. Categories* (Section 4.1): the target task consists of new categories (classes) from within the same base dataset. *II. Datasets* (Section 4.2): the target task is an entirely separate dataset, with different classes and different input text characteristics, but shares the same objective, e.g., intent detection. *III. Objectives* (Section 4.3): the target task relates to the training task only in its form, i.e., text classification, but is a new dataset with different objectives. These different shifts are further discussed in Appendix A.

4.1 News headline classification (*shift in categories*)

Dataset and training We begin our investigation of ICL Markup on a Huffington Post News dataset (Misra, 2022), released for few-shot text classification by Bao et al. (2019).The dataset is partitioned along news categories (classes), with 20 categories available for training, 5 used for validation, and the remaining 16 reserved for testing. We first learn the ICL Markup tokens using the training categories. We then assess the performance on the test categories in (5-way, 10-way) x (1-shot, 5-shot) configurations. Further dataset details can be found in Appendix B.1.

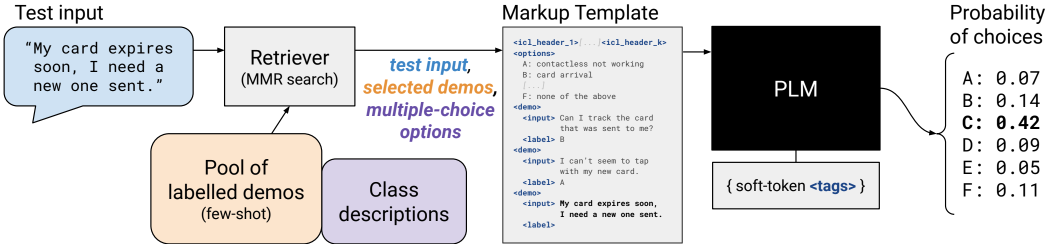
Baselines and comparisons In order to estimate the performance of the Flan-T5 models without ICL Markup we conduct a prompt sweep. We replace the tags in our ICL template with hand-engineered words and phrases, searching over 96 different combinations. We notice a dramatic variation in performance across prompts in the base and large sized models. This variation reduces considerably in the XL and XXL sizes. Further details, including visualizations of this performance variation can be found in Appendix B.2. We further compare our method to the best reported results on this dataset, Prompt-Based Meta-Learning (PBML) (Zhang et al., 2022).

Results We tabulate our results in Table 1. We include additional visualizations in Appendix B.3. For each model size, we report the mean performance of the prompt sweep ± 1 standard deviation. In parentheses, we report the test performance of the best prompt as determined with the validation set. Below each, we report the mean and standard deviation of ICL Markup (ICL-MU), taken over several training runs with different random seeds. In parentheses, we report the test performance of the best training run as determined with the validation set. **We find that ICL Markup improves over the average prompt in every setting.** It also reduces performance variation, especially in the smaller models. With the exception of Flan-T5-large on 5-way classification, we see that the average ICL Markup training run improves over the best prompt as determined by the validation set. **In every test configuration, Flan-T5-XL with ICL Markup outperforms PBML** (which relies on gradient-base parameter updates for few-shot adaption).

[1] When the model output is not a multiple choice option (rarely), we consider it to be "none of the above".

Table 1: Few-shot classification accuracy on Huffington Post dataset. Mean ± stddev (best on val.)

	5-way		**10-way**	
	1-shot	5-shot	1-shot	5-shot
PBML (Zhang et al.)	74.9	78.0	64.6	68.6
Flan-T5-base	53.3±8.4 (64.5)	51.5±9.1 (63.4)	36.7±9.3 (50.6)	35.7±8.8 (49.0)
+ICL-MU	67.8±1.2 (69.7)	68.9±1.9 (70.9)	54.6±1.6 (56.7)	56.1±1.8 (57.9)
Flan-T5-large	65.4±7.8 (74.7)	66.9±7.6 (75.0)	52.9±7.1 (59.4)	53.1±6.9 (59.7)
+ICL-MU	72.1±0.9 (73.6)	73.3±1.0 (74.7)	61.5±0.8 (62.3)	61.5±0.9 (62.4)
Flan-T5-XL	75.7±1.2 (77.8)	76.5±1.5 (78.7)	61.3±1.2 (63.3)	62.1±1.1 (63.9)
+ICL-MU	80.4±0.6 (**81.2**)	82.5±0.2 (**82.7**)	70.3±0.9 (**70.9**)	71.8±0.7 (**72.4**)



```
<icl_header_1>[...]<icl_header_k>
<options>
  A: contactless not working
  B: card arrival
  [...]
  F: none of the above
<demo>
  <input> Can I track the card
          that was sent to me?
  <label> B
<demo>
  <input> I can't seem to tap
          with my new card.
  <label> A
<demo>
  <input> My card expires soon,
          I need a new one sent.
  <label>
```

Figure 2: Our pipeline for using ICL Markup in intent detection.

4.2 Intent detection (*shift in datasets*)

We now consider ICL Markup in few-shot and open-world intent detection tasks. This is an important practical setting that stands to benefit from the flexibility of ICL. Intent categories often change, and data is limited. However, intent detection can be highly multi-class. If the PLM's context window is limited, there may not be enough space in the context window to include a demonstration for each intent class. In extreme cases, i.e., 1000+ intent classes, there may not even be enough context space to enumerate all of the intents. We address this by retrieving the k most relevant demonstrations for an input instance from a (few-shot) candidate pool, similar to Liu et al. (2022). However, rather than selecting the k-nearest-neighbors, we use a maximum marginal relevance (MMR) selection strategy (Carbonell and Goldstein, 1998), encouraging diverse demonstrations as proposed by (Ye et al., 2022). We use the set of labels of the k-selected demonstrations to re-scope the classification for each test instance, narrowing the label space (per test instance) to something that fits in context. This MMR retriever approach requires little overhead beyond a vector database and a lightweight embedding model. Figure 2 illustrates this retriever-controlled ICL Markup pipeline.

Datasets and training We evaluate our approach using the few-shot and open-world intent detection datasets released by (Zhang et al., 2021). We focus on BANKING77 and CLINC150 as well as their open-world variants, BANKING77-OOS, CLINC-Single-Domain-OOS-banking, and CLINC-Single-Domain-OOS-credit-cards. These datasets are particularly challenging because they are designed to contain in-domain out-of-scope (ID-OOS) examples which are semantically similar to the in-scope intent classes. They also contain out-of-domain out-of-scope (OOD-OOS) examples, which are easier to recognize as out of scope. In order to learn the weights for the soft-token tags we build a training set from four of the intent detection datasets. We include HWU64, SNIPS, and ATIS. We also include either BANKING77 or CLINC150 and keep the other (and its OOS variants) as the target task for testing. Further dataset details are described in Appendix C.1. In in both training and testing, we use the MMR retriever to select k-demonstrations per input instance [2] [3]. We trim the set of demonstrations, i.e., decrease k, if it does not fit in the model's context window. With this setup, the true label appears among the multiple choice options in approximately 97% of the examples in the training set. In all intent detection cases, the prompt template includes a “none of the above” option.

Flan-T5 baselines To estimate the ICL performance of the Flan-T5 model on these tasks, we replace the tags in our ICL template with hand-engineered words and phrases, but otherwise use the same pipeline (Figure 2). We focus on Flan-T5 XL for these experiments. We consider 5 sets of (sensible) words and phrases for each task type, i.e., intent detection or legal text classification. These choices are listed in Appendix C.5. We take this approach to isolate the effect of using ICL Markup from the effect of using the MMR retriever. For our few-shot evaluations, we report the mean and standard deviation across the different prompts. For our open-world evaluation, we first choose the best prompt on a warm-up task validation set then report on that (averaging results over few-shot draws).

Few-shot evaluation We test on datasets that are unseen during training. Specifically, few-shot demonstration pools from the target task are made available to the model at test time (for ICL), but the target tasks differ from those used to train the soft-token tags. We evaluate on BANKING77 and CLINC150 using the fixed 5-shot or 10-shot training splits released by Zhang et al. (2020b). The few-shot training data serves as the demonstration pool for the MMR retriever. The results of our

[2] We use $k = 9$ for the intent detection tasks

[3] We use SBERT `all-mpnet-base-v2` embeddings(Reimers and Gurevych, 2019)

Table 2: Few-shot intent detection accuracy

	fine	data	BANKING77		CLINC150	
	tune	aug.	10-shot	5-shot	10-shot	5-shot
ICDA (Lin et al.)	yes	yes	89.8	84.0	94.8	92.6
PTR (Han et al., 2022)	yes	no	86.0	79.7	90.8	90.2
ChatGPT (naive)	no	no	70.9	-	86.5	-
ChatGPT (MMR)	no	no	79.6 ± 0.7	74.6 ± 1.0	86.7 ± 1.1	84.3 ± 1.5
Flan-T5-XL (MMR)	no	no	82.3 ± 0.5	78.5 ± 0.7	89.6 ± 0.2	87.9 ± 0.2
+ICL-MU	no	no	85.5 ± 0.6	82.1 ± 0.4	91.0 ± 0.1	88.8 ± 0.1

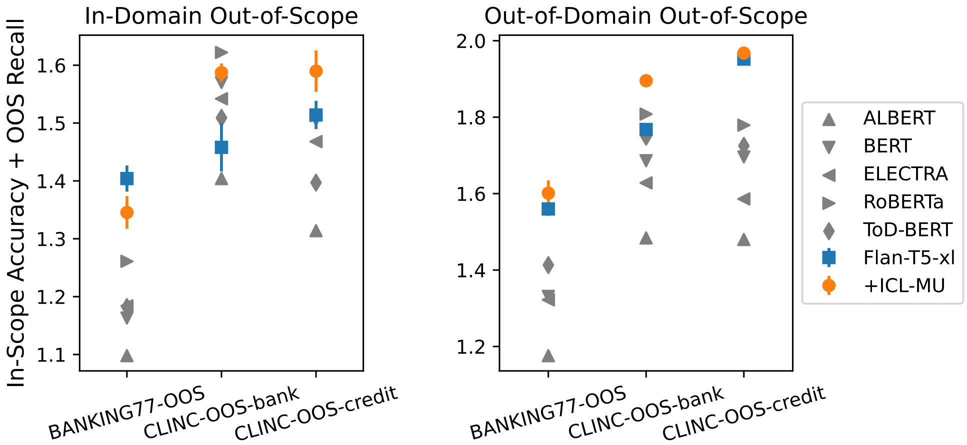


Figure 3: Evaluation on open-world (5-shot) intent detection tasks. Baselines (Zhang et al., 2021) (gray triangles/dimonds) are fine-tuned per task (dataset) on the training set, then use the validation set to tune separate ID-OOS and OOD-OOS thresholds, aiming to maximize accuracy on in-scope intent classes + OOS recall. In contrast, the Flan-T5 baseline and ICL Markup (ICL-MU) do not fine-tune parameters (or the OOD-OOS threshold) using target task data. Points are means over ten 5-shot draws; Flan-T5 and ICL-MU error bars are ±std-dev.

few-shot evaluation are presented in Table 2. We find that ICL Markup improves the Flan-T5-XL model. We compare to the best reported results that we could find on these datasets[4]: In-Context Data-Augmentation Lin et al., as well as the best reported prompt-based baseline: Prompt Tuning with Rules (PTR). We note that both ICDA and PTR fine-tune models on the target task data, while ICDA additionally uses synthetic data augmentation, and PTR involves hand-crafting rule-based prompts from the class names. Thus both methods require considerable effort to configure for a target task. By contrast, ICL keeps the model fixed, requiring only sensible class names and test-time access to the few-shot demonstration pools. We also compare to ChatGPT. First in a naive configuration: where all labels-options are listed and 9 random demonstrations are included for each test instance. We further compare to using ChatGPT in the MMR retriever pipeline we use for Flan-T5 (Figure 2). Flan-T5-XL outperforms both of these ChatGPT baselines. **We find that ICL Markup improves the Flan-T5-XL model[5], pushing its mean performance (over datasets/shots) slightly above PTR.** We found the gains from using ICL Markup to be even greater in the smaller base model. For example, the performance of ICL Markup with Flan-T5-Base on CLINC150 and BANKING77 (10-shot) is shown in Figure 1 (right), where we see a clear and substantial improvement.

Open-world evaluation Open-world classification involves the identification of out-of-scope (OOS) inputs, requiring the PLM to identify inputs that do not belong to any of the predetermined intent classes. It has been shown that large PLMs can be negatively affected both in accuracy and calibration when they are presented with a "none of the above" option (Kadavath et al., 2022), however, we

[4] Literature searches were conducted in July 2023.

[5] Soft-token tags initialized randomly did not significantly improve Flan-T5-XL on 5-shot CLINC150.

explore whether exposure to such an option during the warm-up phase may allow the PLM to select it with more success. For the open-world evaluation we follow a procedure similar to the one used to produce the baselines released with the OOS datasets (Zhang et al., 2020a,b). We consider ID-OOS and OOD-OOS examples separately, and average our results over ten random 5-shot or 10-shot draws from each of the target task's training sets. However, *unlike the previous baselines, we do not use target task validation data to select the OOD-OOS thresholds*. When assessing against OOD-OOS examples, we simply use the model's (greedy) generated prediction, taking "none of the above" to correspond to OOS. However, like the previous baselines, when assessing against the challenging ID-OOS examples, we increase the OOS sensitivity by tuning a threshold using the target task's validation set. If our model has assigned a probability above this threshold to the "none of the above" multiple choice option, we interpret the prediction as OOS. The results of the 5-shot open world experiment are shown in Figure 3. The 5 baselines are from work by Zhang et al. (2021). They were optimized to maximize the in-scope accuracy plus out-of-scope recall. Separate models are trained for each dataset-shot combination, and OOD-OOS and ID-OOS thresholds were tuned separately on the validation sets. **ICL Markup outperforms all previous baselines in 5 of the 6 5-shot configurations** (7 of 12 total). It also outperforms our Flan-T5-XL baseline in 5 of the 6 5-shot configurations (10 of 12 total). See Appendix C.3 for complete results.

4.3 Legal text classification (*shift in objectives*)

To assess whether ICL Markup tags learned for one task objective can add value in a completely different objective, we evaluate the ICL Markup tags learned for intent detection on a version of LEDGAR (Tuggener et al., 2020), a 100-way text classification dataset in the legal domain. We follow the same approach as for our few-shot evaluation in Section 4.2, setting k=7 (since the input texts are longer). However, this task is not few-shot; all training examples are available to the MMR retriever as candidate demonstrations. The results are presented in Table 3. They show that the soft token tags, which were *trained on only intent detection tasks*, can nonetheless help the model improve its ICL ability with this task.

Table 3: Accuracy on LEDGAR legal text classification dataset

MMR Retriever	17.9	(guess on multiple-choice)
kNN	77.7	(using SBert embeddings)
Flan-T5-XL	77.6 ± 2.0	
+ICL-MU	79.6 ± 0.2	(p-value 0.03)

5 Discussion

In-context learning offers great flexibility for application-developers wishing to leverage PLMs, but it also lacks robustness and leads to arbitrary decisions which may affect the system's performance. Our markup-inspired proposal offers structure to help minimize these situations, and our experimental results are promising. When compared to hand-crafted prompts, ICL Markup can reduce variability and improve performance. We believe this is especially interesting in an application area like intent detection which stands to gain from the increased flexibility of ICL. Our few-shot HuffPost classification results, and open-world intent detection results are noteworthy in and of themselves. They indicate that the benchmarks on these datasets can be outperformed using ICL for few-shot adaptation, rather than parameter updates. ICL can then be further improved with Markup templates.

Limitations Our experiments are limited, especially in the models examined. The PLMs considered (250M, 780M and 3B parameters) are small compared to the state of the art. Additionally, our experimental scope is limited to classification tasks, but ICL has much broader applications.

Acknowledgments and Disclosure of Funding

Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through NSERC and CIFAR, and companies sponsoring the Vector Institute[6]. We would like to thank the University of Toronto computing and administrative staff for their support, the anonymous reviewers for their insightful feedback, and Tom Zollo, Zhun Deng, and Lillio Mok for their helpful comments.

References

Y. Bao, M. Wu, S. Chang, and R. Barzilay. Few-shot Text Classification with Distributional Signatures. 8 2019. URL `http://arxiv.org/abs/1908.06039`.

J. Carbonell and J. Goldstein. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. In SIGIR 1998 - Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 1998. doi: 10.1145/290941.291025.

I. Chalkidis, A. Jana, D. Hartung, M. Bommarito, I. Androutsopoulos, D. M. Katz, and N. Aletras. LexGLUE: A Benchmark Dataset for Legal Language Understanding in English. 10 2021. URL `http://arxiv.org/abs/2110.00976`.

Y. Chen, R. Zhong, S. Zha, G. Karypis, and H. He. Meta-learning via Language Model In-context Tuning. Technical report.

Y. Chen, C. Zhao, Z. Yu, K. McKeown, and H. He. On the Relation between Sensitivity and Accuracy in In-context Learning. 9 2022. URL `http://arxiv.org/abs/2209.07661`.

H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, A. Castro-Ros, M. Pellat, K. Robinson, D. Valter, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, and J. Wei. Scaling Instruction-Finetuned Language Models. 10 2022. URL `http://arxiv.org/abs/2210.11416`.

N. Ding, Y. Qin, G. Yang, F. Wei, Z. Yang, Y. Su, S. Hu, Y. Chen, C. M. Chan, W. Chen, J. Yi, W. Zhao, X. Wang, Z. Liu, H. T. Zheng, J. Chen, Y. Liu, J. Tang, J. Li, and M. Sun. Parameter-efficient fine-tuning of large-scale pre-trained language models. Nature Machine Intelligence, 5 (3):220–235, 3 2023. ISSN 25225839. doi: 10.1038/s42256-023-00626-4.

Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, L. Li, and Z. Sui. A Survey on In-context Learning. 12 2022. URL `http://arxiv.org/abs/2301.00234`.

Y. Gu, X. Han, Z. Liu, and M. Huang. PPT: Pre-trained Prompt Tuning for Few-shot Learning. Technical report.

X. Han, W. Zhao, N. Ding, Z. Liu, and M. Sun. PTR: Prompt Tuning with Rules for Text Classification. AI Open, 3:182–192, 1 2022. ISSN 26666510. doi: 10.1016/j.aiopen.2022.11.003.

E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. LoRA: Low-Rank Adaptation of Large Language Models. 6 2021. URL `http://arxiv.org/abs/2106.09685`.

S. Kadavath, T. Conerly, A. Askell, T. Henighan, D. Drain, E. Perez, N. Schiefer, Z. Hatfield-Dodds, N. DasSarma, E. Tran-Johnson, S. Johnston, S. El-Showk, A. Jones, N. Elhage, T. Hume, A. Chen, Y. Bai, S. Bowman, S. Fort, D. Ganguli, D. Hernandez, J. Jacobson, J. Kernion, S. Kravec, L. Lovitt, K. Ndousse, C. Olsson, S. Ringer, D. Amodei, T. Brown, J. Clark, N. Joseph, B. Mann, S. McCandlish, C. Olah, and J. Kaplan. Language Models (Mostly) Know What They Know. 7 2022. URL `http://arxiv.org/abs/2207.05221`.

S. Khosla and R. Gangadharaiah. Evaluating the Practical Utility of Confidence-score based Techniques for Unsupervised Open-world Intent Classification. Technical report, 2022. URL `https://huggingface.co/roberta-base`.

[6] `https://vectorinstitute.ai`

B. Lester, R. Al-Rfou, N. Constant, and G. Research. The Power of Scale for Parameter-Efficient Prompt Tuning. Technical report. URL https://github.com/google-research/.

Y.-T. Lin, A. Papangelis, S. Kim, S. Lee, D. Hazarika, M. Namazifar, D. Jin, Y. Liu, and D. Hakkani-Tur. Selective In-Context Data Augmentation for Intent Detection using Pointwise V-Information. Technical report. URL https://huggingface.co/docs/transformers/main_.

X. Lisa Li and P. Liang. Prefix-Tuning: Optimizing Continuous Prompts for Generation. Technical report.

J. Liu, D. Shen, Y. Zhang, B. Dolan, L. Carin, and W. Chen. What Makes Good In-Context Examples for GPT-3? Technical report, 2022.

P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. ACM Computing Surveys, 55(9), 1 2023. ISSN 15577341. doi: 10.1145/3560815.

Y. Lu, M. Bartolo, A. Moore, S. Riedel, and P. Stenetorp. Fantastically Ordered Prompts and Where to Find Them: Overcoming Few-Shot Prompt Order Sensitivity. 4 2021. URL http://arxiv.org/abs/2104.08786.

S. Min, M. Lewis, L. Zettlemoyer, and H. Hajishirzi. MetaICL: Learning to Learn In Context. 10 2021. URL http://arxiv.org/abs/2110.15943.

R. Misra. News Category Dataset. 9 2022. URL http://arxiv.org/abs/2209.11429.

C. Qian, H. Qi, G. Wang, L. Kunc, and S. Potdar. Distinguish Sense from Nonsense: Out-of-Scope Detection for Virtual Assistants. 1 2023. URL http://arxiv.org/abs/2301.06544.

C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. Technical report, 2020. URL http://jmlr.org/papers/v21/20-074.html.

N. Reimers and I. Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. 8 2019. URL http://arxiv.org/abs/1908.10084.

G. Sahu, P. Rodriguez, I. H. Laradji, P. Atighehchian, D. Vazquez, and D. Bahdanau. Data Augmentation for Intent Classification with Off-the-shelf Large Language Models. 4 2022. URL http://arxiv.org/abs/2204.01959.

D. Tuggener, P. Von Däniken, T. Peetz, and M. Cieliebak. LEDGAR: A Large-Scale Multilabel Corpus for Text Classification of Legal Provisions in Contracts. Technical report, 2020. URL https://drive.switch.ch/index.

J. Wei, L. Hou, A. Lampinen, X. Chen, D. Huang, Y. Tay, X. Chen, Y. Lu, D. Zhou, T. Ma, and Q. V. Le. Symbol tuning improves in-context learning in language models. 5 2023. URL http://arxiv.org/abs/2305.08298.

X. Ye, S. Iyer, A. Celikyilmaz, V. Stoyanov, G. Durrett, and R. Pasunuru. Complementary Explanations for Effective In-Context Learning. 11 2022. URL http://arxiv.org/abs/2211.13892.

H. Zhang, A. Sneyd, and M. Stevenson. Robustness and Reliability of Gender Bias Assessment in Word Embeddings: The Role of Base Pairs. In Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 759–769, 2020a. URL https://aclanthology.org/2020.aacl-main.76.pdf.

H. Zhang, X. Zhang, H. Huang, and L. Yu. Prompt-Based Meta-Learning For Few-shot Text Classification. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 1342–1357, Stroudsburg, PA, USA, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.87. URL https://aclanthology.org/2022.emnlp-main.87.

J. Zhang, K. Hashimoto, Y. Wan, Z. Liu, Y. Liu, C. Xiong, and P. S. Yu. Are Pretrained Transformers Robust in Intent Classification? A Missing Ingredient in Evaluation of Out-of-Scope Intent Detection. 6 2021. URL `http://arxiv.org/abs/2106.04564`.

J.-G. Zhang, K. Hashimoto, W. Liu, C.-S. Wu, Y. Wan, P. S. Yu, R. Socher, and C. Xiong. Discriminative Nearest Neighbor Few-Shot Intent Detection by Transferring Natural Language Inference. 10 2020b. URL `http://arxiv.org/abs/2010.13009`.

T. Z. Zhao, E. Wallace, S. Feng, D. Klein, and S. Singh. Calibrate Before Use: Improving Few-Shot Performance of Language Models. Technical report, 2021. URL `https://www.github.com/tonyzhaozh/few-shot-learning`.

A Relationship between training ("warm-up") and target tasks

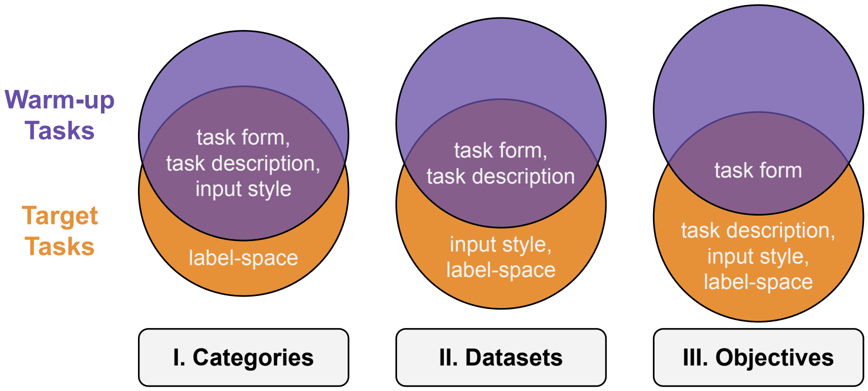

Warm-up Tasks
Target Tasks
task form, task description, input style
label-space
task form, task description
input style, label-space
task form
task description, input style, label-space
I. Categories
II. Datasets
III. Objectives

Figure 4: Relationship between warm-up and target tasks.

In order to learn the weights for the soft-token tags, we first construct a prompt template using ICL Markup that is appropriate for the *form* of the target task. Throughout this work, the template we use is a multiple-choice question with a variable number of options, followed by a variable number of demonstrations. See Figure 1 (left) as an example. We then fine-tune the soft-tokens in the template using a collection of related training tasks. These warmp-up tasks may differ from the target task along a few axes, e.g.,

Label-space: The set of classes and their definitions, determined by $p(Y = y|X = x)$.

Input style: The characteristics of the textual input, e.g., text length, writing-style, determined by the empirical input distribution, $p(X)$, defined by a dataset.

Task objective: The description of the objective of the task, in the sense of the instructions one would prompt a PLM with, e.g., "categorize these news articles", or "determine the legal purpose of these contract provisions".

We consider three types of shifts in our experiments in Section 4:

I. Categories: (Section 4.1) Changes are primarily in the label space. The target task consists of new categories (classes) from within the same dataset as the one used for training.

II. Datasets: (Section 4.2) Changes in both the label space and input style. The target task is an entirely separate dataset, with different classes and different input text characteristics, but sharing the same objective, e.g., intent detection.

III. Objectives: (Section 4.3) Changes in everything except the form of the task. The target task relates to the training task only in that is still constitute text classification, but this is on a new dataset with different objectives.

These different shifts are depicted in Figure 4.

B Huffington Post Experiments

B.1 Dataset details

We experiment using a Huffington Post News dataset (Misra, 2022) processed and released for meta-learning and few-shot learning by Bao et al. (2019)[7]. The goal is to classify news headlines into their corresponding news category, e.g. "World News", "Arts & Culture". There are 41 such categories that have been divided up in 20 categories for training, 5 for validation, and 16 for testing. We build training, validation, and test sets using a data loader released by Zhang et al. (2022). The

[7] The few-shot dataset is released with pretokenized text. Rather that use it in this way, we map the tokenized headlines back to the original dataset and use the original headlines instead, preserving capitalization and punctuation

data loader samples random few-shot episodes, first choosing 5 or 10 classes (5-way or 10-way), then choosing 1 query example for each class, then choosing 1 or 5 supporting examples per class available as demonstrations for ICL (1-shot or 5-shot). Each of our test sets is composed of 10,000 query examples (which is 2000 episodes in 5-way testing, and 1000 episodes in 10-way testing). The baseline Flan-T5 and Flan-T5+ICL-MU models are tested on the same query examples. We learn the ICL Markup tokens using only the training categories, sampling 5000 episodes from each of the (5-way, 10-way) by (1-shot, 5-shot) configurations. We use this combined training set for all ICL-Markup training runs, i.e., our tokens are not way/shot specific.

Table 4: HuffPost dataset composition

Text Length (avg.)	Example/Class	Train Classes	Val. Classes	Test Classes
11	900	20	5	16

B.2 Decomposing prompt variation

The 96 prompts considered in the sweep were created with the cartesian product of: {Intructions} x {Options Header} x {Demo separator} x {Input Indicator} x {Label Indicator} x {Puctuator}. This roughly captures the different axes of arbitrary decisions that must be made when desiging a multiple-choice style prompt. To better understand the effect of these choices on prompt performance, we consider the distribution of performance (on a HuffPost validation set) "sliced" along each axis, thus showing the individual effect of each componente. This is shown in Figure B.2. We see that the choice of Demo Separator (from within the two options we tried) seems to have a pretty negligible effect, while the choice of Input Indicator or Punctuator is much more pronounced. However, the clearest trend is that the larger Flan-T5 models are considerably less susceptible to performance variations due to any of these choices.

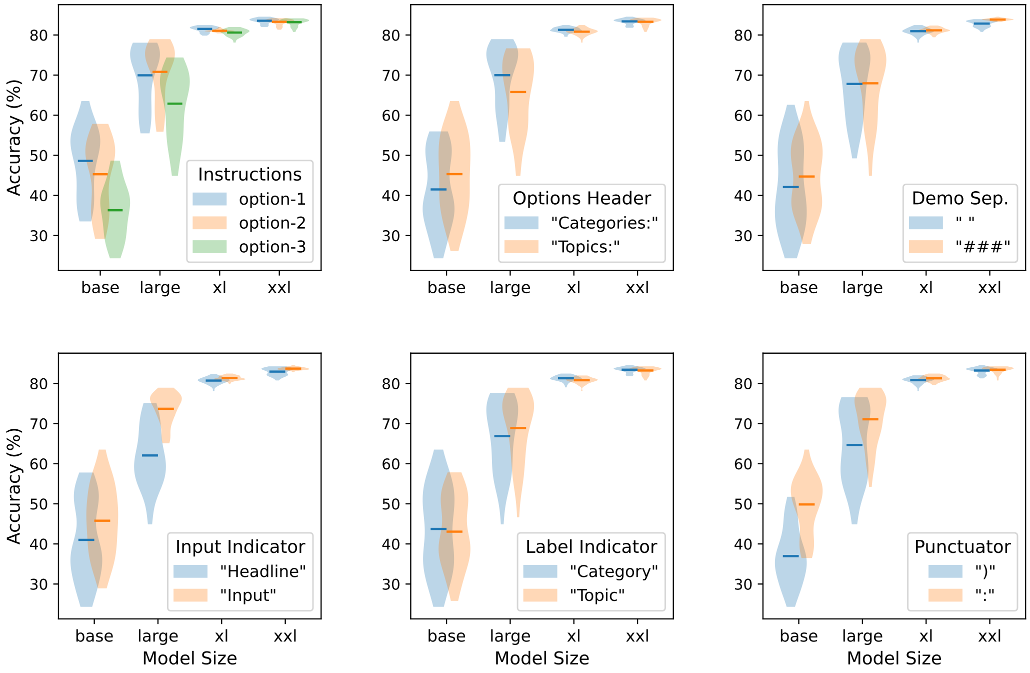


Figure 5: Decomposition of prompt performance along different choices. Violin plots show variation in performance when one component of the prompt is fixed to a particular option. Solid horizontal lines within the violins depict the mean. In the top-left plot, the instruction options are option-1: "Categorize the following news headlines according to their topic." option-2: "Classify these headlines based on the type of news." option-3: "Identify the type of news based on following headlines."

B.3 Additional figures

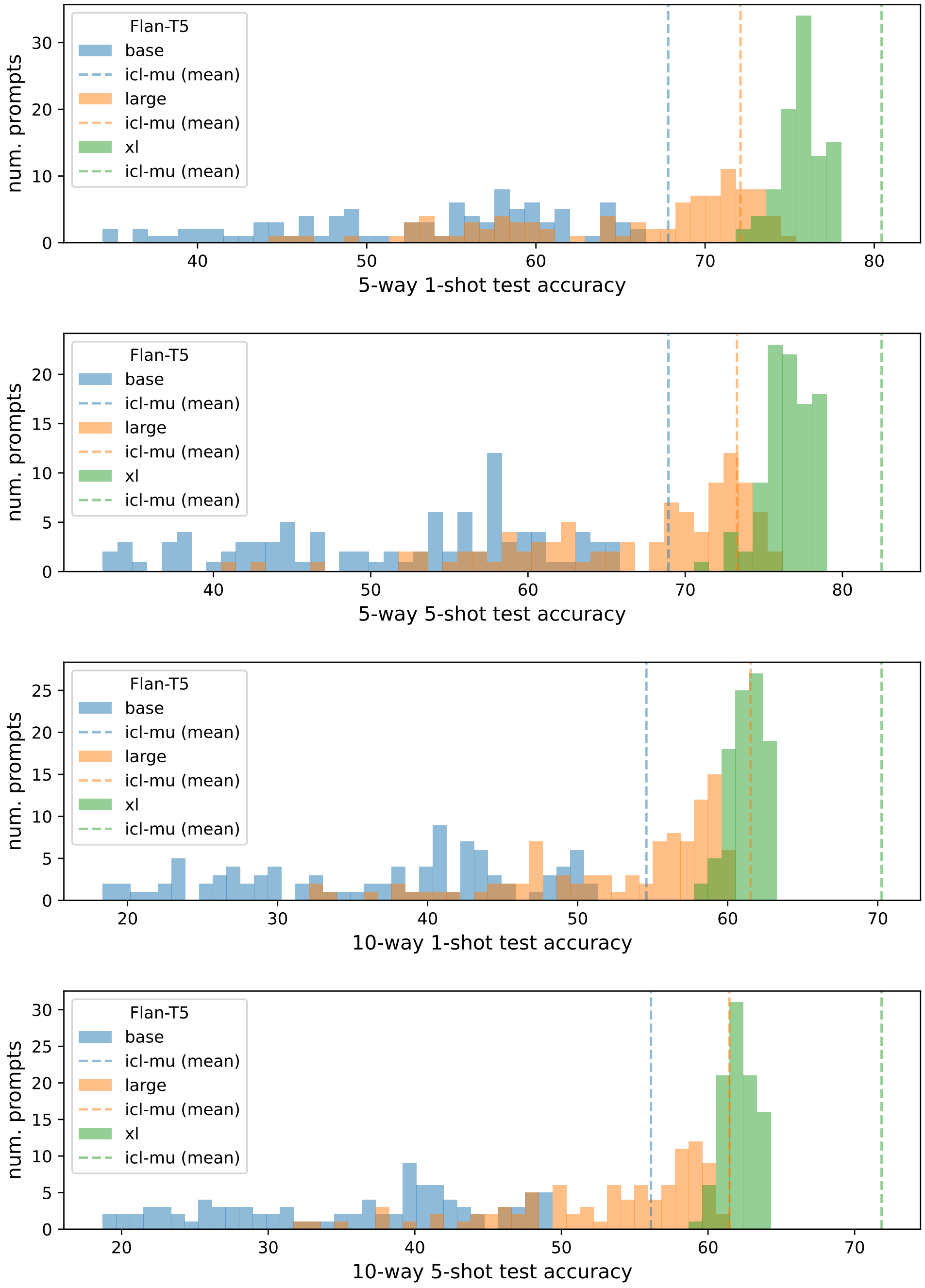


Figure 6: Visual depictions of the results in Table 1. Histograms depict the test performance of a prompt sweep over 96 prompts, across 3 Flan-T5 model sizes. Dotted vertical lines depict the mean ICL-MU performance at each model size.

C Intent Detection Experiements

C.1 Training set composition

In order to learn the weights for the soft-token tags we built a training set from three intent detection datasets: HWU64, SNIPS, and ATIS. We further include either BANKING77 or CLINC150, the one that is *not* the target for testing. These datasets are described in Table 5. Specifically, ICL Markup tags tested on BANKING77 and BANKING77-OSS are trained with dataset splits marked with an 'B', and hyper-parameters and the ID-OOS threshold are tuned with those marked with a 'b'. Tokens tested on CLINC150 and CLINC-Single-Domain-OOS are developed in the same way with the splits marked 'C' and 'c'. Dataset splits that have been crossed out were unused by our ICL Markup models. During the "warm-up" process where the tokens are learned, we draw on examples from the training sets to serve as inputs, and use the validation sets to serve as the demonstration pools. Note there are two CLINC-Single-Domain-OOS datasets: banking and credit-cards. They have the same number of intents classes and the same number of instances in each split. Also note that only few-shot sub-samples of the training sets were used for hyperparameter tuning and threshold choices.

Table 5: Intent detection datasets

	Intents	Train	Valid	Test
CLINC150	150	15,000^{B}	3,000^{B}	4,500^{b}
BANKING77	77	8,622^{C}	1,540^{C}	3,080^{c}
HWU64	64	8,954^{B,C}	1,076^{B,C}	~~1,076~~
SNIPS	7	13,084^{B,C}	700^{B,C}	~~700~~
ATIS	17	4478^{B,C}	500^{B,C}	~~893~~
CLINC-SD-OOS (x2)	10	500^{c}	500^{c}	500
id-oos		-	400^{c}	350
ood-oos		-	~~200~~	1,000
BANKING77-OOS	50	5,905^{b}	1,506^{b}	2,000
id-oos		-	530^{b}	1,080
ood-oos		-	~~200~~	1,000

C.2 Intent detection dataset preprocessing

All intent detection datasets are lower-cased. For each dataset, the descriptive class names (which form the multiple-choice options in the filled prompt template) are derived from the included label names. The processing is very simple, limited mostly to changing underscores to spaces. The exceptions were:

- removing the substring "atis_" preceeding all ATIS labels
- changing "flight no" to "flight number" (also ATIS)
- separating the words in the seven SNIPS labels, e.g., "addtoplaylist" becomes "add to play list"

C.3 Additional open-world results

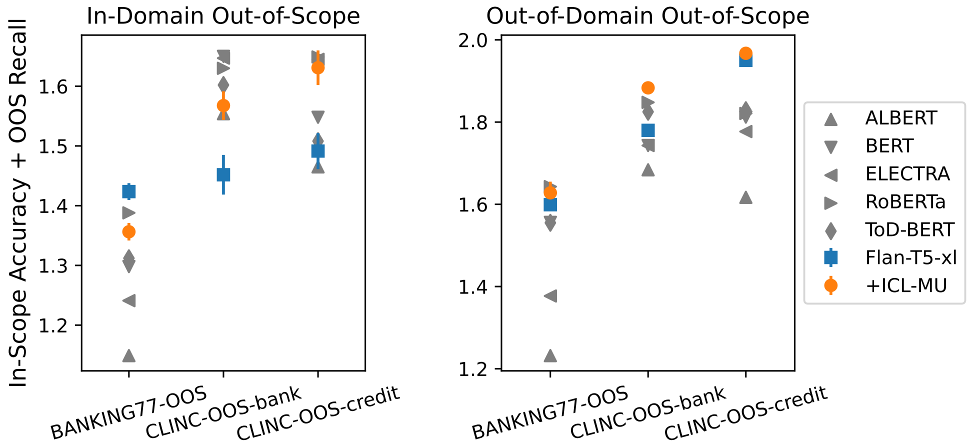


Figure 7: Evaluation on open-world (10-shot) intent detection tasks. Baselines (Zhang et al., 2021) (gray triangles/dimonds) are fine-tuned per task (dataset) on the training set, then use the validation set to tune separate ID-OOS and OOD-OOS thresholds, aiming to maximize accuracy on in-scope intent classes + OOS recall. In contrast, the Flan-T5 baseline and ICL Markup (ICL-MU) do not fine-tune parameters (or OOD-OOS thresholds) using target task data. Points are means over ten 10-shot draws; Flan-T5 and ICL-MU error bars are ±std-dev.

		data_source	BANKING77-OOS				CLINC-OOS-banking				CLINC-OOS-credit-cards			
				IS-Acc	OOS-Rcl	OOS-Prc		IS-Acc	OOS-Rcl	OOS-Prc		IS-Acc	OOS-Rcl	OOS-Prc
5-shot	id-oos	ALBERT		20.3	89.5	39.8		54.1	86.3	57.9		55.5	75.9	55.8
		BERT		25.4	90.9	41.3		75.2	81.8	70.8		74.1	76.5	68.1
		ELECTRA		30.9	87.5	43.0		64.8	89.4	65.1		71.0	75.8	67.1
		RoBERTa		43.0	83.1	46.3	*	83.8	78.4	78.6		64.5	86.8	63.3
		ToD-BERT		35.5	82.7	43.8		75.1	75.8	69.4		67.4	72.3	61.3
		Flan-T5-xl	*	59.0	81.4	61.1		68.4	77.4	67.3		74.9	76.5	68.3
		+ICL-MU		56.6	77.9	56.6		78.7	80.0	74.7	*	82.1	76.9	75.4
	ood-oos	ALBERT		20.3	97.3	39.9		63.1	85.3	83.4		55.5	92.5	81.5
		BERT		39.0	94.1	49.0		75.2	93.4	88.8		74.1	95.5	88.4
		ELECTRA		39.1	93.1	48.7		75.5	87.3	88.8		71.0	87.6	87.0
		RoBERTa		62.1	93.9	68.7		83.8	97.0	92.9		81.2	96.7	91.4
		ToD-BERT		52.9	88.4	66.0		83.0	91.9	92.8		75.8	96.7	89.6
		Flan-T5-xl		73.8	82.2	97.0		92.6	84.2	100.0		99.1	96.1	99.8
		+ICL-MU	*	73.0	87.1	86.3	*	95.0	94.6	99.1	*	98.4	98.3	99.6
10-shot	id-oos	ALBERT		27.3	87.6	42.4		77.8	77.6	72.2		66.7	79.8	64.0
		BERT		52.5	77.3	50.8	*	77.5	87.5	73.8		80.3	74.5	73.1
		ELECTRA		40.1	84.0	46.1		79.5	85.2	75.4		78.0	86.5	73.3
		RoBERTa		59.7	79.1	55.8		76.6	86.4	72.7	*	81.0	83.9	75.8
		ToD-BERT		54.3	76.9	52.1		80.7	79.5	75.4		80.6	70.2	71.9
		Flan-T5-xl	*	61.3	81.0	62.2		69.9	75.3	67.6		74.3	74.9	67.3
		+ICL-MU		55.6	80.0	55.0		79.6	77.2	74.8		84.2	78.9	78.2
	ood-oos	ALBERT		30.5	92.7	47.1		77.8	90.6	89.8		66.7	95.0	85.7
		BERT		64.2	91.4	68.9		77.5	96.8	90.0		90.1	91.1	95.5
		ELECTRA		40.1	97.6	47.9		79.5	94.8	90.7		88.6	89.1	94.2
		RoBERTa	*	70.3	94.0	73.3		89.2	95.6	95.4		87.5	94.6	94.0
		ToD-BERT		60.6	94.9	63.3		86.5	96.0	94.2		86.5	96.4	93.7
		Flan-T5-xl		77.2	82.7	97.2		93.3	84.7	100.0		99.1	95.9	99.8
		+ICL-MU		76.9	85.9	89.8	*	95.3	93.0	99.2	*	98.8	97.9	99.8

Table 6: Detailed open-world (few-shot) test performance. Baselines (Zhang et al., 2021) are fine-tuned per task/shot on the training set, then use the validation set to tune separate ID-OOS and OOD-OOS thresholds, aiming to maximize accuracy on in-scope intent classes (IS-Acc) + OOS recall (OOS-Rcl). In contrast, ICL-MU does not fine-tune parameters (or the OOD-OOS threshold) using target task data. For each of the 12 dataset/shot/OOS-type configurations the best model is indicated by a star.

C.4 Tuning OOS Thresholds

As mentioned in Section 4, when assessing against out-of-domain out-of-scope (OOD-OOS) examples, we simply use the model's (greedy) generated prediction, taking "none of the above" to correspond to OOS. However, when assessing against the challenging ID-OOS examples, we increase the OOS sensitivity by tuning a threshold using the target task's validation set. If the model has assigned a probability above this threshold to the "none of the above" multiple choice option, we interpret the prediction as OOS. In Figure 8 we show the performance of ICL Markup vs. different threshold values. We notice that initializing tokens randomly leads to more robustness with regards to threshold choice. We use this initialization strategy for the open-world evaluation.

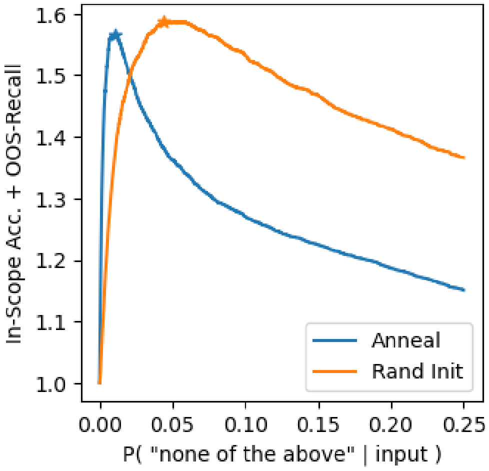


Figure 8: ID-OOS threshold sweep on validation set

C.5 Hand written ICL templates

Here we show the handwritten ICL templates that were used in to create the Flan-T5 baselines for intent detection and LEDGAR. These were also used to initialize the soft token tags for the initialization strategy "anneal".

C.5.1 Intent Detection

1. **icl header** Categorize the following user statements according to their intent.
   options header category options:
   demo indicator example
   input indicator statement:
   label indicator category:
2. **icl header** Classify the user inquiries below according to their intent.
   options header possible classes:
   demo indicator demonstration
   input indicator inquiry:
   label indicator class:
3. **icl header** Label these user requests based on their intent type.
   options header label options:
   demo indicator example
   input indicator request:
   label indicator label:
4. **icl header** Classify these user utterances based on their principal intent.
   options header possible classes:

   demo indicator ###
   input indicator utterance:
   label indicator class:

5. **icl header** Determine the intent of the following incoming user requests.
   options header intent options:
   demo indicator e.g.
   input indicator request:
   label indicator intent:

C.5.2 Legal Text Classification

1. **icl header** Categorize the following contract provisions according to their main topic.
   options header category options:
   demo indicator example
   input indicator provision:
   label indicator category:
2. **icl header** Classify the contract provisions below according to their main topic.
   options header possible classes:
   demo indicator demonstration
   input indicator provision:
   label indicator class:
3. **icl header** Label these contract provisions based on their main topic.
   options header label options:
   demo indicator example
   input indicator provision:
   label indicator label:
4. **icl header** Classify these contract provisions based on their primary topic.
   options header possible classes:
   demo indicator ###
   input indicator provision:
   label indicator class:
5. **icl header** Determine the main topic of the following contract provisions.
   options header topic options:
   demo indicator e.g.
   input indicator provision:
   label indicator topic:

D LEDGAR Experiments

To understand whether the learned soft token tags have value on tasks outside of intent detection, we evaluate them on a version of LEDGAR (Tuggener et al., 2020), a legal text classification dataset. The objective is to classify provisions in legal contracts. We start with a version included in LexGLUE (Chalkidis et al., 2021) that has been trimmed down to 100 different classes, then further remove all provisions longer than 75 tokens. T5's context window is only 512 tokens, and we wanted to ensure that at least a few demonstrations fit per test instance.

Because this task is not few-shot, the advantage of using an LLM is limited. In Table 3 we see that Flan-T5-XL slightly under-performs the SBERT (all-mpnet-base-v2) kNN model. However, we find that ICL-Markup still improves Flan-T5, pushing its performance beyond this baseline.