Style-Unified Meta-In-Context Learning: Improving In-Context Learning Ability by Learning to Unify Output Styles

Anonymous ACL submission

Abstract

This paper proposes a style-unified meta-in-context learning that enhances In-Context Learning (ICL) ability for language models by learning to unify the output styles. Meta-training for ICL (MetaICL), a method that learns ICL ability for enhancing to follow a few in-context examples, has been proposed. However, the language models trained with MetaICL might not be able to consider information obtained from in-context examples at inference because it has been reported that the performance is unaffected when random or flipped outputs are used in a few in-context examples. Our key idea for using in-context information is explicitly giving a relationship between outputs in context and a target output by unifying the output style. Specifically, arbitrary symbols (e.g., integer or word) are inserted into the outputs in context, and we expect the model to focus on examples by learning to output the same symbols at the same positions. To evaluate the proposed method, we create a Japanese dataset containing multiple examples per task. Experiments using a 0.6B Japanese language model demonstrate that the proposed method outperforms the conventional method.

1 Introduction

In-Context Learning (ICL) improves the ability to solve unseen tasks at inference in language models by conditioning on a few in-context examples (Brown et al., 2020). However, since the language models do not learn to solve tasks conditioned on the few in-context examples during pre-training, there is a gap between pre-training and using ICL (Chen et al., 2022). To fill the gap, Meta-training for ICL (MetaICL), a method to learn how to do ICL, has been proposed. In MetaICL, the language models are fine-tuned on a large set of tasks that contain multiple input-output examples. This method is expected to improve ICL capability and solve unseen tasks with high performance.

However, it has been reported that task performance is unaffected when random or flipped outputs are used in a few in-context examples in MetaICL (Min et al., 2022b; Wei et al., 2023b). This suggests the model may not consider the information from a few examples but only the instruction and input to solve the task. Therefore, symbol tuning has been proposed to encourage language models to use in-context information (Wei et al., 2023a). In symbol tuning, by learning input and arbitrary symbol (e.g., "foo," "bar") mapping instead of input and natural language labels (e.g., "positive," "negative"), the model is expected to figure out tasks using in-context information. This approach has enabled improvements in ICL ability and following flipped outputs in context. Unfortunately, it treats only classification tasks and cannot deal with generation tasks.

To mitigate this problem, we consider a method to learn the ICL ability while using in-context information, which can be used for generation tasks. Our idea for considering a few in-context examples more explicitly is to give a strong relationship between the outputs of a few examples and a target output by unifying these output styles. Specifically, arbitrary symbols are inserted into the outputs of a few examples, and the same symbol is inserted at the same position for the target output. With this approach, we expect language models to solve tasks by following a few in-context examples because they need to access examples to predict the symbol and its position. In addition, this approach can be applied to both classification and generation tasks easily because symbols are not replaced with the output but only added to the output.

In this paper, we propose *style-unified meta-in-context learning* as a method to learn ICL ability. In the proposed method, a language model is fine-tuned using training data inserted the same symbol into the outputs in context and target output at the same position. Specifically, a symbol is selected

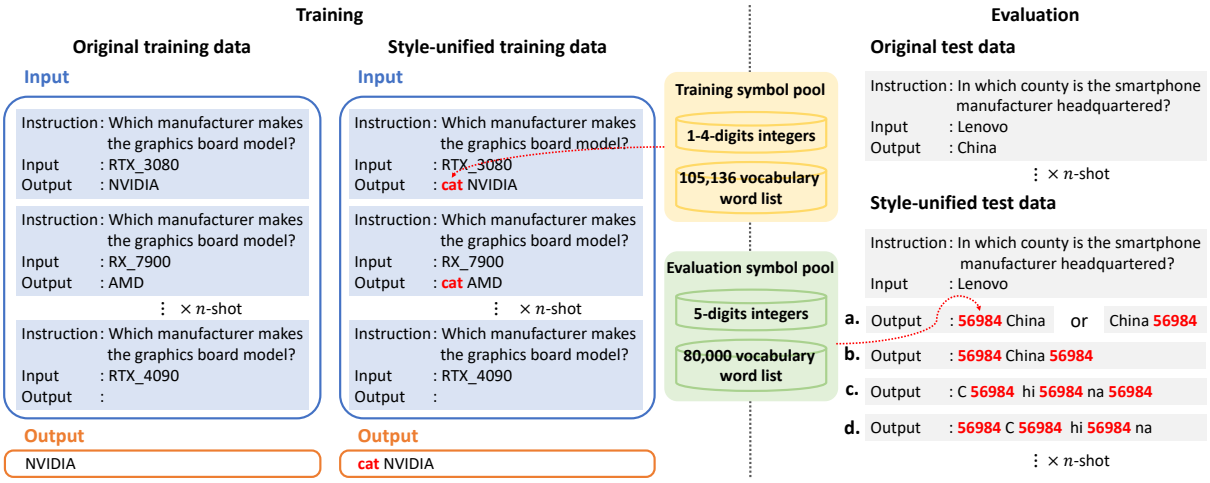

Figure 1: Overview of style-unified meta-in-context learning. In style-unified data, symbols are inserted into outputs to unify the output style in context. We prepare four position patterns and a different symbol pool in test data to investigate the ability to follow a few in-context examples.

from the symbol pool (e.g., integers and words), and it is inserted at a randomly selected position either before or after the output (e.g., instruction, input, output → instruction, input, *symbol* + output, or output + *symbol*). To evaluate the proposed method, we create a Japanese dataset that involves 3,231 Japanese generation tasks (e.g., question answering). Experiments using our dataset and a 0.6B language model demonstrate that the proposed method outperforms MetaICL.

2 Related Work

ICL can improve the ability of language models that solve unseen tasks at inference time by conditioning on a few in-context examples (Brown et al., 2020). However, it has been reported that the performance of ICL strongly relies on the demonstration surface, including the demonstration format and the order of demonstration examples (Zhao et al., 2021; Perez et al., 2021). To this end, several studies have examined demonstration design strategies, such as demonstration selection (Liu et al., 2022; Rubin et al., 2022), ordering (Lu et al., 2022), and formatting (Zhou et al., 2023; Wei et al., 2022). There are also studies on how to learn the ICL ability, as language models are not explicitly trained to learn in context (Chen et al., 2022). For example, methods to learn ICL ability by learning target tasks with a few in-context examples have been proposed (Min et al., 2022a; Wei et al., 2023a), as have methods that pre-train language models by automatically building data containing few examples from a general plaintext corpus to maintain generalization performance (Gu et al., 2023). The proposed method is regarded as an advanced method for learning target tasks with a few in-context examples to improve the ICL ability. The key advance is that it can treat generation tasks and leans to unify the output styles for using in-context information.

3 Style-Unified Meta-In-Context Learning

In this paper, we propose a novel method to learn ICL ability, *style-unified meta-in-context learning*, which can be used for classification and generation tasks. In the proposed method, as shown in Fig. 1, a language model is trained using a dataset containing multiple in-context examples (an original dataset in this paper). The proposed method is an advanced method of MetaICL that trains language models using the original dataset. In contrast to MetaICL, the proposed method trains language models using both the original and a style-unified dataset. The style-unified dataset is created by inserting the same symbol into the output of a few examples and the target output at the same position, as shown in Fig. 1. In this dataset, the arbitrary symbols are selected randomly from a symbol pool containing integers and words[1][2]. Also, the insertion position is randomly selected before or after the output. In summary, the proposed method is trained using the style-unified dataset where arbi-

[1]In our evaluation of the Japanese task in Sec. 4, we used 185,136 vocabularies from the Japanese vocabulary list (Maekawa et al., 2014) as a word set.

[2]The integer and word categories of the symbol pool were determined by referencing symbol-tuning (Wei et al., 2023a).

trary symbols are inserted randomly before or after the output. The model learns to output the same symbol at the same position as the outputs of a few in-context examples; in other words, it learns to unify the output style in context. We expect language models to focus on a few in-context examples by using the proposed method because they need to access the output of examples to output the same symbol at the same position. In addition, the proposed method can be applied to any task because symbols are added to unify the output style.

4 Experiment

Dataset: Training data with multiple examples per task is required to evaluate the proposed method, but currently, there are few such datasets in Japanese. We therefore create a Japanese dataset containing multiple examples per task using crowdsourcing. First, we give 13 Japanese workers 15 categories[3]. Next, the workers select one category and create the appropriate instruction, input, and output for that category as questions with uniquely defined answers. The workers create ten samples per instruction, and the categories should be selected so that the number of instructions is not biased. Then, we instruct the workers that the instruction and output should be sentence and word, respectively, and the input could be sentence or word. In this dataset, some tasks have duplicate outputs because the same output may be produced for different inputs. We show examples of data in the "liberal arts" and "living" categories below.

- Instruction: Which country matches the capital? \n Input: Tokyo \n Output: Japan
- Instruction: What is the number of days in the month? \n Input: April \n Output: 30

Also, we divide the dataset into training, validation, and test sets to avoid task overlap. Table 1 shows the breakdown. In the evaluation, unseen tasks are evaluated because the tasks in the test set are not included in the training set. The training and validation sets randomly contain 2–4 examples per task. In the proposed method, the style-unified training data is created using a symbol pool with 1–4-digit integers and 105,136 words. The test set contains 1–4 examples per task. Note that we use only a prompt of the above example format in this dataset.

[3]Internet/ Entertainment/ Smart Devices/ News/ Economy/ Living Things/ Manners/ Liberal Arts/ Health/ School/ Food/ Interpersonal relationships/ Community/ Living/ Other

	No. of tasks	No. of samples
Train	3,231	32,310
Validation	175	1,750
Test	178	1,780

Table 1: Number of tasks and samples in train, validation, and test sets in our created dataset.

Evaluation: In our experiment, the methods are evaluated using the original dataset. Also, if the model can learn to follow a few in-context examples, it should be able to generate the output in the same style as the examples by using in-context learning. Thus, to investigate the ability to follow a few in-context examples, we create four patterns of style-unified test data, as shown in Fig. 1: a) *inserting symbols before or after the output randomly* (trained), b) *inserting symbols before and after the output* (not trained), c) *inserting symbols for the right side of each token in output* (not trained), and d) *inserting symbols for the left side of each token in output* (not trained). The symbols are selected from a symbol pool with 5-digit integers and 80,000 words, and none of them are duplicated in the training set. In the evaluation with style-unified test data, an answer is considered correct when the same symbol is output in the same position as the examples in context and incorrect even if only the answer is correct.

Models: For these evaluations, we used the small Japanese language model we created. Small language models have the advantage of requiring fewer machine resources and faster inference speed, making them suitable for commercial deployment. Our language model is constructed as a Transformer encoder-decoder model (Vaswani et al., 2017) with 24 encoder layers and 12 decoder layers, with 0.6B parameters. In pre-training, we used Japanese plaintext corpus such as MC4, Oscar, and Wikipedia, and our language model was trained using the UL2 (Tay et al., 2022). We evaluate the proposed method by comparing it with a baseline and a conventional method.

- Baseline: Our pre-trained language model.
- Conventional method: Fine-tuned language model with MetaICL (Min et al., 2022a) using the original training data constructed in Sec. 4.

	No. of tasks	Original test data In-context learning 1-shot	2-shot	3-shot	4-shot	Style-unified test data 4-shot in-context learning a	b	c	d
Baseline	–	33.5	42.5	42.3	42.0	42.6	41.6	35.2	33.4
Conventional	1,500	45.8	49.0	50.1	50.9	33.6	23.1	38.3	17.4
Proposed		47.2	50.9	52.4	53.0	52.5	51.9	42.5	42.4
Conventional	3,231	49.7	51.9	52.5	53.5	34.8	25.4	38.7	19.1
Proposed		**50.5**	**53.7**	**54.9**	**55.2**	**54.3**	**52.4**	**43.7**	**43.3**

Table 2: Results of in-context learning using original and style-unified test data.

	In-context learning 1-shot	2-shot	3-shot	4-shot
Baseline	11.2	16.5	17.9	18.3
Conventional	8.7	11.7	14.0	15.4
Proposed	11.2	14.4	17.0	18.1

Table 3: Results of in-context learning using original test data in which outputs are replaced with arbitrary symbols.

- Proposed method: Fine-tuned language model with style-unified meta-in-context learning using the original and style-unified training data described in Sec. 3.

For the fine-tuning, we used the RAdam optimizer (Liu et al., 2019) and label smoothing (Lukasik et al., 2020) with a smoothing parameter of 0.1. We set the mini-batch size to 32, and the dropout rate in each Transformer block to 0.1. The tokenizer is our trained SentencePiece (Kudo and Richardson, 2018) that has 30K tokens. It takes less than a day to finish fine-tuning on a single A100 80G GPU.

5 Results

Main result: Table 2 lists the results of ICL using the original and the style-unified test data described in Sec. 3. In the table, the value represents the accuracy and is calculated as the exact match rate between the generated result and the reference. Lines 2–3 show the results using 1,500 tasks in the training set, and lines 4–5 show the results using all tasks. The accuracy of lines 2–5 are the averages calculated from models with three different parameters. For the original test data results, the proposed method outperformed the conventional method. The reason is that notational distortion often occurred between the outputs in context and generated output in the conventional method, but this was improved in the proposed method. For the style-unified test data results, the performance of the proposed method is approximately 10% better than the baseline for all patterns a–d. On the other hand, since the conventional method failed to generate symbols for most outputs, its performance was significantly worse than the baseline. These results indicate that the proposed method forces the models to use in-context information by unifying output styles and can improve the ICL ability.

Evaluation of ability to follow examples: Wei et al. (2023b) showed that while language models could follow flipped outputs presented in context to some extent, MetaICL degraded this ability. To evaluate whether the proposed method improves the ability to follow examples in context, we created test data by replacing the outputs in the original test data with randomly selected symbols prepared in Sec. 3. These symbols were completely unrelated to the tasks, and the same output word was replaced with the same symbol. Table 3 lists the results. As shown, the conventional method reduced the ability to follow arbitrary symbols in context, but the proposed method recovered to the same level as during pre-training. This indicates that the proposed method improves the ability to solve tasks by following the examples in context. Since this data contains many generation tasks, predicting symbols not present in context is challenging, and the accuracy range was small.

6 Conclusion

In this paper, we proposed style-unified meta-in-context learning, a method to improve ICL ability by learning to unify the output styles. In the proposed method, when the model learns ICL, the same arbitrary symbols (e.g., words or integers) are inserted into outputs of examples and a target output at the same position. Since language models need to focus on examples in context in order to output symbols, we expect the proposed method to improve the ICL ability. To investigate the effectiveness of the proposed method, we created a Japanese dataset containing ten examples per task. Experimental results using a 0.6B Japanese language model showed that the proposed method outperformed the conventional method.

Limitations

In this paper, our evaluation had three limitations. First, we only used a single small language model in our experiment. Since the results may differ depending on the sizes and kinds of language models, evaluating the method on different sizes and kinds of language models would be worthwhile. On the other hand, large language models are not easy to use because they require large computational resources and take a lot of time for training. Also, evaluating other language datasets and models would be worthwhile because the proposed method does not rely on language.

Next, in the proposed method, the style-unified dataset was created by inserting arbitrary symbols into the outputs of a few examples in context at the same position. Although these symbols were selected from a symbol pool containing integers and words, there are various ways to create a symbol pool, such as combining integers and words, creating long symbols, selecting a set of other words, and using only numbers or words. Thus, it would be worthwhile to select symbols from a vast pool. In addition, it would be worthwhile investigating the performance when there is an increase or decrease in patterns of position, although only two patterns were used concerning the position at which the symbols are inserted.

Finally, this study dealt with a single prompt format in the dataset; however, studies have shown that language models are not robust to prompts (Brown et al., 2020; Reynolds and McDonell, 2021). If more variations of prompt formats are added during training and evaluation, it should be possible to show that the proposed format is not dependent on the prompt format.

References

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

Mingda Chen, Jingfei Du, Ramakanth Pasunuru, Todor Mihaylov, Srini Iyer, Veselin Stoyanov, and Zornitsa Kozareva. 2022. Improving in-context few-shot learning via self-supervised training. In *Proc. the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 3558–3573.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Pre-training to learn in context. In *Proc. the Association for Computational Linguistics (ACL)*, pages 4849–4870.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 66–71.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proc. Deep Learning Inside Out: The Workshop on Knowledge Extraction and Integration for Deep Learning Architectures (DeeLIO)*, pages 100–114.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2019. On the variance of the adaptive learning rate and beyond. In *Proc. International Conference on Learning Representations (ICLR)*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proc. the Association for Computational Linguistics (ACL)*, pages 8086–8098.

Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. 2020. Does label smoothing mitigate label noise? In *Proc. the International Conference on Machine Learning (ICML)*, pages 6448–6458.

Kikuo Maekawa, Makoto Yamazaki, Toshinobu Ogiso, Takehiko Maruyama, Hideki Ogura, Wakako Kashino, Hanae Koiso, Masaya Yamaguchi, Makiro Tanaka, and Yasuharu Den. 2014. Balanced corpus of contemporary written japanese. *Language Resources and Evaluation*, pages 345–371.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022a. MetaICL: Learning to learn in context. In *Proc. the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 2791–2809.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022b. Rethinking the role of demonstrations: What makes in-context learning work? In *Proc. the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 11048–11064.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. *Advances in neural information processing systems*, pages 11054–11070.

Laria Reynolds and Kyle McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. In *Proc. Conference on Human Factors in Computing Systems (CHI)*, page 1–7.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proc. the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 2655–2671.

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier García, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. 2022. Ul2: Unifying language learning paradigms. In *Proc. International Conference on Learning Representations (ICLR)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proc. Advances in neural information processing systems (NIPS)*, pages 5998–6008.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, pages 24824–24837.

Jerry Wei, Le Hou, Andrew Lampinen, Xiangning Chen, Da Huang, Yi Tay, Xinyun Chen, Yifeng Lu, Denny Zhou, Tengyu Ma, and Quoc Le. 2023a. Symbol tuning improves in-context learning in language models. In *Proc. Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 968–979.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and Tengyu Ma. 2023b. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proc. the International Conference on Machine Learning (ICML)*, pages 12697–12706.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2023. Large language models are human-level prompt engineers. *arXiv preprint arXiv:2211.01910*.