PICLe: Pseudo-annotations for In-Context Learning in Low-Resource Named Entity Detection

Anonymous ACL submission

Abstract

In-context learning (ICL) enables Large Language Models (LLMs) to perform tasks using few demonstrations, facilitating task adaptation when labeled examples are hard to come by. However, ICL is sensitive to the choice of demonstrations, and it remains unclear which demonstration attributes enable in-context generalization. In this work, we conduct a perturbation study of in-context demonstrations for low-resource Named Entity Detection (NED). Our surprising finding is that in-context demonstrations with partially-correct annotated entity mentions can be as effective for task transfer as fully correct demonstrations.

Based off our findings, we propose **P**seudo-annotated **I**n-**C**ontext **Le**arning (PICLe), a framework for in-context learning with noisy, pseudo-annotated demonstrations. PICLe leverages LLMs to annotate large quantities of demonstrations in a zero-shot first pass. We then cluster these synthetic demonstrations, sample specific sets of in-context demonstrations from each cluster, and predict entity mentions using each set independently. Finally, we use self-verification to select the final set of entity mentions. We extensively evaluate PICLe on five biomedical NED datasets and show that, with zero human annotation, PICLe outperforms ICL in low-resource settings where few gold examples can be used as in-context demonstrations.

1 Introduction

With in-context learning (ICL), Large Language Models (LLMs) can be adapted to perform many tasks using few demonstrations (Brown et al., 2020; Dong et al., 2022; Srivastava et al., 2023; Ye et al., 2023). This emergent property of LLMs is particularly beneficial in tasks where limited supervision data is available for fine-tuning models, such as in specialized domains where only expensive expert annotations can be relied upon to produce quality data (e.g., biomedical, clinical, legal domains, among many others), and in situations where in-house proprietary datasets must be compiled with few available experts to perform the annotation.

Despite its promise in these settings, ICL is highly sensitive to the choice of the demonstrations (Wang et al., 2024; Li and Qiu, 2023; Liu et al., 2021), and it remains unclear which characteristics of demonstrations are critical for successful task adaptation. Consequently, prior work has explored which demonstration characteristics lead to successful task adaptation in ICL (Min et al., 2022; Yoo et al., 2022; Wei et al., 2023), but these studies have focused on tasks with a focus on scalar outputs such as classification tasks, with a limited and pre-defined label space. As a result, demonstration characteristics that maximize performance are still unclear for tasks that require structured, open-ended prediction such as Named Entity Detection (NED), where the label space is effectively bounded only by the number of domain entities.

In this work, we focus on NED given its high number of use cases, particularly in specialized domains where effective annotation is challenging, as (1) it requires considerable domain expertise, and (2) entities can change over time, introducing distribution shifts in supervised datasets over time.

We conduct a thorough analysis of demonstration properties that impact in-context adaptation in NED. First, we analyze the importance of the context-label correspondence of in-context demonstrations, corrupting the demonstrations to add noise to the context-label mapping, with different perturbations retaining different dimensions of information about this mapping. Second, we investigate the degree of *partial correctness* of demonstrations. Indeed, contrary to single-label classification tasks, answers to open-ended token-level tasks such as NED can be partially correct. We experiment with various perturbation schemes to produce demonstrations with differing levels of correctness. We find that ICL is much less sensitive to corrup-

tions that retain even a weak semantic mapping between the input context and the label set. Moreover, our analysis reveals that in-context learning is surprisingly resilient to partially-incorrect annotations so long as a high number of entity annotations remain in the demonstration labels.

Following this analysis, we introduce **P**seudo-annotated **I**n-**C**ontext **Le**arning (PICLe), a framework for in-context NED with pseudo-annotated demonstrations that requires no human labeling effort. First, we exploit a pool of unlabeled samples to obtain pseudo-annotations through zero-shot prediction from LLMs, followed by a self-verification step in which the model is prompted to verify the type of individual entities. Then, the noisy pseudo-annotated samples are clustered, and demonstration sets are sampled from each cluster individually. These cluster-specific demonstrations are used to predict the entities mentioned in the test query. Predictions from all clusters are consolidated to obtain the final set of entity mentions. In our evaluations with multiple LLMs across five biomedical entity detection datasets (Taboureau et al., 2010; Li et al., 2016; Smith et al., 2008), we show that PICLe is as effective as, and on average outperforms, standard ICL that uses gold-labeled demonstrations.

In summary:

1. We conduct a perturbation study to identify the demonstration attributes that make in-context learning work in low-resource NED. We find that above a surprisingly low correctness threshold, partially-correct entity mention annotations can be as effective for in-context learning as demonstrations with fully correct gold annotations, particularly in scarce annotation settings.

2. We propose PICLe, a novel framework for in-context learning that uses pseudo-annotated demonstrations as in-context examples. We show that with no human-annotation effort, PICLe competes and even outperforms ICL with gold-labeled demonstrations in resource-scarce settings.

2 Related works

What matters in in-context learning? In-context learning is remarkably effective for performing various NLP tasks with only a few task demonstrations appended to the prompt (Brown et al., 2020). However, despite a large body of work on designing novel in-context learning methods (e.g., Gao et al., 2021; Sorensen et al., 2022; Mishra et al., 2022), it is not yet fully understood what makes in-context learning effective, with multiple works demonstrating surprising variables, such as the impact of the demonstration order (Lu et al., 2022), the term frequencies of test examples in pretraining data (Razeghi et al., 2022), and basic output calibration (Zhao et al., 2021; Fei et al., 2023; Jiang et al., 2023b). Consequently, recent works explore *how* demonstration components might be separately responsible for in-context transfer. Min et al. (2022) show that in-context demonstrations serve to show the label space of demonstrations, the distribution of their input text, and their overall format. However, Yoo et al. (2022) perform quantifiable analysis on the impact of ground-truth label demonstrations on multiple tasks and datasets and find that ground-truth labels have substantial impacts on ICL performance. Wei et al. (2023) continue this line of work and show that the degree to which the label mapping influences task transfer depends on the scale of the model, and that smaller models are more capable of ignoring misaligned label mappings. Wang et al. (2023a) show similar results for CoT reasoning, finding that CoT is also possible without valid demonstrations, and that demonstrations that are relevant to the query and have the correct order of reasoning steps are more important for effective transfer.

However, these works focus on classification tasks, which have no notion of "partial correctness": a label is either correct or not. In token-level tasks such as NED, the list of gold-annotated entities can be partially correct. For NED, we show that partially correct demonstrations can be as effective as fully correct ones, which these works do not show. Furthermore, in contrast with Min et al. (2022)'s findings for classification, we actually show that ICL demonstrations with fully incorrect labels are not effective in NED.

Pseudo-annotation. Pseudo-annotation is a popular semi-supervised learning method in many domains (Yang et al., 2022). It has recently been used for various NLP tasks to generate demonstrations for ICL (Wan et al., 2023a,b) and fine-tuning LLMs (Huang et al., 2023; Honovich et al., 2023; Wang et al., 2023b). Demonstrations are either random (*e.g.* Z-ICL, Lyu et al., 2023, for classification tasks) or partially correct. In particular, COSP (Wan et al., 2023a) selects and builds a demonstration pool from an LLM's zero-shot outputs via multiple rounds of prediction with high

temperature and exceeds few-shot baselines for a range of reasoning tasks. Most similar to our work is Self-ICL (Chen et al., 2023b), which uses zero-shot models to generate in-context demonstrations for text classification. In our work, we construct a pipeline for leveraging zero-shot predicted labels for real test examples in named entity detection, but ground our pseudo-annotation method in analysis of how demonstration noise influences downstream in-context learning performance.

Information extraction with in-context learning. Although LLMs have achieved SOTA performance in many NLP tasks, their performance in extraction tasks is still significantly below supervised baselines (Ma et al., 2023). Recent works have designed dedicated prompting techniques to improve in-context NER for LLMs (Lee et al., 2022; Shen et al., 2023; Chen et al., 2023a). PromptNER (Shen et al., 2023) provides the entity definition to the model, and prompts it to output a list of potential entities with an explanation justifying the compatibility of each entity with the provided definition, achieving considerable improvement compared to vanilla prompting, but requiring further human effort to annotate examples that may not be available in many settings. In our work, we adopt a similar task formulation as Prompt-NER, but do not require labeled examples or explanations, as we use pseudo-annotation to produce in-context learning examples.

3 Experimental setup

The task of Named Entity Detection (NED) requires detecting all mentions of entities in a text.

We formulate the task such that the language model is given a passage of text as part of a prompt and must predict the list of entities that are mentioned in the passage. Optionally, in few-shot settings (i.e., in-context learning), the prompt also contains several demonstrations, which each include an example passage and a corresponding list of mentioned entities in the passage.

Datasets. We consider five biomedical NED datasets with rich and comprehensive collections of diverse specialized entity types. **ChemProt** (Taboureau et al., 2010) contains annotations for extracting chemical compounds (drugs) and gene and protein-related objects (GPRO). Originally, each sample of this dataset is a paragraph, but we split these paragraphs into sentences. We construct two datasets from ChemProt: *ChemProt-Chem* and *ChemProt-Gene*, for detecting chemicals and genes, respectively. **BC5CDR** (Li et al., 2016) contains biomedical abstracts annotated for chemical and disease extraction. Similar to ChemProt, we conduct our experiments on two sub-portions, *BC5-Chem* and *BC5-Disease*. Finally, **BC2GM** (Smith et al., 2008) contains biomedical abstracts annotated for the extraction of genes, proteins, and related entities. We summarize statistics for these datasets in Table 1.

Models. We use three LLMs in our experiments: the proprietary GPT-3.5-Turbo, and the open-source Mistral-7b-instruct (Jiang et al., 2023a) and Llama-2-7b-Chat (Touvron et al., 2023). In the remainder of the paper, we refer to them as Mistral and Llama2, respectively.

Metrics. Using each dataset's original IOB2 annotation scheme, we compute the micro-averaged Precision, Recall, and F1 to measure entity mention detection performance.[2] We strictly evaluate using exact spans: longer or shorter predicted spans than the gold span are marked as incorrect.

4 Do we need gold demonstrations?

We conduct an exploratory analysis of which components of in-context demonstrations are critical for task transfer by studying the effect of fully incorrect and partially incorrect demonstrations in the in-context prompt. Our analysis demonstrates that while random annotations considerably underperform compared to zero-shot performance, partially correct annotations can be as effective as gold annotations (Figure 2) for in-context transfer for named entity detection.

4.1 Investigating the input-output correspondence of in-context demonstrations

Prior research demonstrates that correct demonstrations are not imperative for priming models in classification tasks (Lyu et al., 2023); incorrect demonstrations are sufficient to show desired in-context transfer behavior, including domain relevance and annotation format.

In this analysis, we investigate essential demonstration attributes for successful in-context task transfer by designing various demonstration corruption schemes, each targeting specific demon-

[1] https://huggingface.co/bigbio

[2] We use sequeval (https://github.com/chakki-works/seqeval/) a widely-used Python library for sequence labeling evaluation.

Source	Name	Entity type	#Train	#Test	Avg # words per entity	Ratio null samples (%)
ChemProt (Taboureau et al., 2010)	ChemProt-Chem	chemical	10 732	8 431	1.39	41.3
	ChemProt-Gene	gene/protein			1.62	45.0
BC5CDR (Li et al., 2016)	BC5-Chem	chemical	4 560	4 797	1.36	35.3
	BC5-Disease	disease/illness			1.70	41.7
BC2GM (Smith et al., 2008)		gene/protein	12 575	5 039	2.45	48.9

Table 1: **Datasets description and statistic**: number of samples (sentences) in train and test splits, average number of words per entity and null samples ratio (ratio of samples with no labeled entities) in train split. We use the versions available in the HuggingFace library.[1]

stration aspects (see Table 2 for examples). We then compare performance under these corruptions to zero-shot prediction (**`No Demo`**) and standard in-context learning (**`Gold Label`**). For each setting, the prediction example remains unchanged and the model receives the same instruction prepended to the prompt.

`Random ID Label`: We replace ground-truth entity labels with random in-domain entities. For each input sentence, every entity in the ground-truth annotation is replaced by an in-distribution (ID) entity randomly sampled from all labels in training examples of the dataset.

`Random OOD Label`: We replace entity labels in the ground-truth demonstrations with a random out-of-distribution (OOD) English word.[3]

`Corrupted OOD Text`: We replace the entity mentions in the text with random out-of-distribution (OOD) English words.[4] For **`Corrupted OOD Text and Label`**, we replace ground-truth labels as well, such that the entities in the text and label match.

`Corrupted and Shuffled OOD Text`, `Corrupted and Shuffled OOD Text and Label`: Same as their non-shuffled counterpart, but with randomly shuffling the words of the sentence.

Results Figure 1 shows Mistral's performance averaged over all datasets. For more corruption schemes, detailed results per dataset, and the performance of GPT-3.5-Turbo (similar to Mistral), see Appendix, Section B. As expected, demonstrations with gold annotations consistently improve the performance over no demonstrations. However, corrupting demonstrations downgrades the performance, particularly in cases like `Random ID Label` and `Random OOD Label`, which are notably worse than zero-shot prediction. This observation differs from the findings of Min et al. (2022) for in-context text classification and multiple choice question answering tasks, as well as Wang et al. (2023a)'s observations for question answering with chain-of-thought reasoning, likely due to the more open nature of the prediction task (i.e., predicting multiple labels from a broad label space). In both of these corruption schemes, the contextual and semantic correspondence between the input sentence and the gold entities is lost. Indeed, the labels are either not in the gold label domain (`Random OOD Labels`), or present in the gold label domain but decorrelated from the target entities (`Random ID Label`). The model learns spurious text-label correspondence through these demonstrations and under-performs compared to zero-shot prediction.

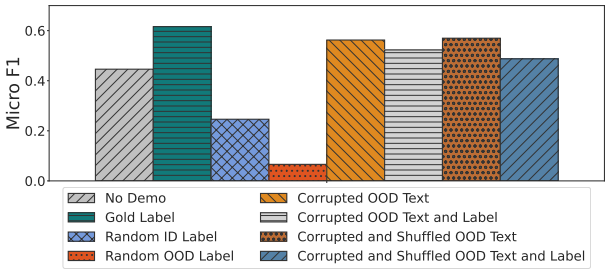

Figure 1: **10-shot ICL performance using various demonstration corruption schemes**, with Mistral and *k*NN demonstration retrieval. We compare to zero-shot and gold demonstrations, averaging over all datasets.

Interestingly, both shuffled and unshuffled text schemes (`Corrupted (and Shuffled) OOD Text`) exhibit no significant performance drops, maintaining an edge over zero-shot prompting. This is intriguing, especially since the input prompt is the same for all corruptions in Table 2. We hypothesize that the model relies less on word order in the

[3]OOD words are randomly sampled from the English vocabulary in the `NLTK` library (Bird et al., 2009).

[4]OOD words are randomly sampled from the English vocabulary in the `NLTK` library (Bird et al., 2009)

Text	This pretreatment had no effect on the inhibition of GABA-T or the elevation of brain GABA levels produced by VIG .
Gold Labels	[GABA, GABA, VIG]
`Random ID Labels`	[dacarbazine, DTIC]
`Random OOD Labels (from nltk)`	[unmeliorated, suddy, vista]
`Corrupted OOD Text`	This pretreatment had no effect on the inhibition of unmeliorated or the elevation of brain suddy levels produced by vista .
`Corrupted and Shuffled OOD Text`	of had by produced on elevation no . levels or effect the vista of the This unmeliorated pretreatment brain inhibition suddy

Table 2: **Examples of different text and labels corruption schemes**. Source: ChemProt-Chem.

demonstrations to adapt to NED. Similar to how previous work showed that models no longer represent local word order in long contexts (Sun et al., 2021), we infer that the model does not need to represent explicit word order in exemplars to use them for transfer for a non-shuffled test sample. Moreover, despite label corruption in (`Corrupted (and Shuffled) OOD Text and Labels`) cases, performance slightly decreases compared to schemes with intact labels, yet still outperforms the zero-shot setting. This finding suggests the model potentially can induce label presence from the global context as ICL with these demonstrations still outperforms zero-shot predictions by up to 10%.

Based on these findings, we conclude that for effective in-context task transfer in NED, the demonstrations must retain a degree of semantic correspondence between the input text and the extracted entities, but that the model's ability to adapt in-context is robust to noise in the demonstrations.

4.2 Partially correct in-context demonstrations

Our first analysis showed that in-context task adaptation was robust to noise in the demonstrations, so long as there remained a context-label correspondence that could be exploited by the model. To investigate this finding further, we perform a second study where we perturb demonstrations by modifying the context-label correspondence in a controlled manner. Specifically, we vary the correctness of the gold labels by applying different heuristic perturbations to the gold entity labels according to a perturbation factor $p \in \{0.1, 0.2, ...0.9\}$:

`Deletion`: each entity in the ground-truth annotation is deleted with probability p.

`Substitution`: each entity in the ground-truth annotation is substituted with a random entity chosen from the dataset's label space with probability p.

`Addition and Substitution`: for each entity in the ground-truth, an entity chosen randomly from the dataset's label space is added with probability p; additionally, each ground-truth entity is substituted with a random entity from the same label space with probability p.

`Deletion and Substitution`: each entity in the ground-truth is removed with probability p. The remaining entities are substituted with a random entity from the dataset with probability p.

Following these perturbations, we report the precision, recall, and F1 score of the perturbed demonstrations (evaluated based off the initial gold demonstration labels) against the F1 score of downstream predictions for test samples that contain at least one entity in their gold annotations.[5]

Results Demonstrations subject to different perturbations may exhibit similar demonstration F1 scores, but result in considerably different prediction F1 scores (Figure 2). Specifically, we note that for a fixed demonstration F1 score, the perturbed demonstrations that retain a higher number of entities in the demonstration achieve much greater performance (i.e., `Substitution` and `Addition and Substitution`). In fact, even with heavily perturbed demonstration labels, the prediction F1 stays above zero-shot performance so long as some of the gold entities remain in the demonstration labels, and even remains close to the performance of 10-shot in-context learning with gold labels. Based off our findings, we hypothesize that noisily labeled demonstrations (such as those predicted by a zero-shot model) could provide in-context learning benefits for named entity detection.

[5] Further results about the number of entities in the demonstration and perturbation factor, along with a comparison of the precision and recall of demonstrations against predictions, can be found in Appendix Figures 6 and 7.

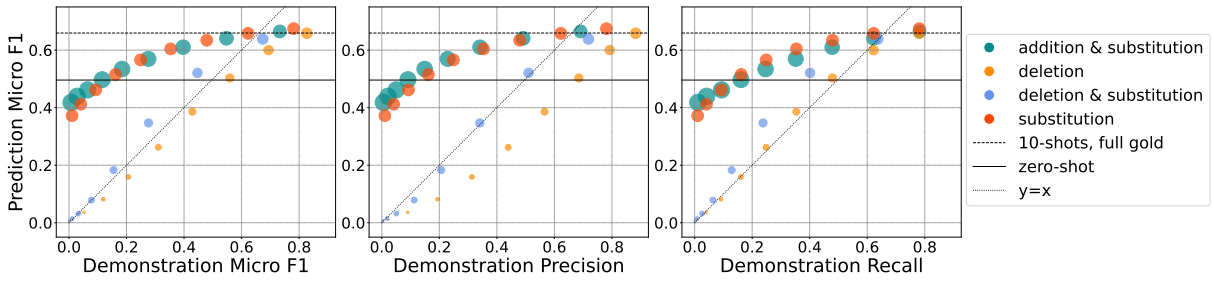

Figure 2: **10-shot ICL performance with perturbed demonstrations** using different perturbation schemes and *k*NN demonstration retrieval. We report the prediction F1 as a function of the precision, recall, and F1 of the perturbed demonstration label sets (relative to the gold demonstrations) averaged over all datasets. The size of the points shows the average number of entities in the label sets of the perturbed demonstrations.

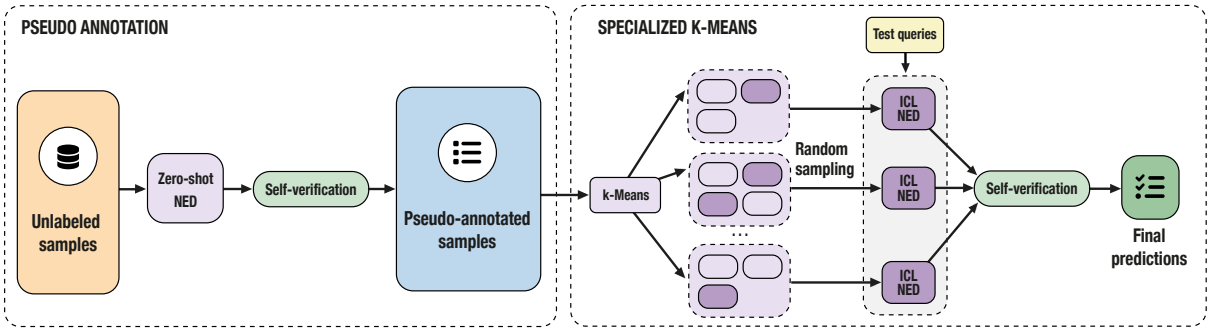

Figure 3: **PICLe pipeline**: Unlabeled samples are pseudo-annotated through a zero-shot prediction and self-verification. Subsequently, they are clustered and cluster-specific sets of in-context demonstrations are chosen at random from each group. Each set is independently used to find entity mentions in the query, and the final set of entity mentions is obtained by aggregating these independent sets and asking the model to verify each predicted entity.

5 In-context NED with pseudo-annotated demonstrations

In this section, we propose PICLe, a framework for pseudo-annotating unlabeled samples that can be leveraged for in-context learning. PICLe, depicted in Figure 3, consists of two stages. In the first stage, we start with a set of unlabeled samples and prompt the model in a zero-shot pass to extract the mentions of entities in each sample. Then we further improve the quality of pseudo-annotations by prompting the model to verify each predicted entity (a process referred to as self-verification; Weng et al., 2023), and filter entities that are not of the correct entity type. We use *k*-means clustering to group the remaining pseudo-annotated samples into K clusters based on the embedding of their text and pseudo-annotations.[6]. Each cluster is used as an individual pool of demonstrations for the downstream NED task. In the second stage, we prompt the model K times, each time choosing the demonstrations from one cluster of pseudo-annotated samples (a sampling method we refer to as Sp-*k*-means, i.e., Specialized *k*-means). Then, for each entity in the K lists of predictions, we perform a self-verification step to verify if the entity has the correct type or not, and retain the extracted entities that have the correct entity type. In all of our experiments, we pseudo-annotate 1000 samples from the training set of the datasets using greedy decoding.

PICLe performance We evaluate PICLe on the same five biomedical NED datasets used for our analysis in Section 4 and compare PICLe's performance with standard ICL using gold demonstrations sampled from different demonstration pool sizes, representing various degrees of annotation scarcity. For baselines that use gold annotations as in-context examples, we initially sample demonstration pools of size of N from the full training set of each dataset, which range in size from 4.5K to 12.5K examples (see Table 1). In scarce annotation settings, we then sample demonstrations from these pools for gold in-context learning using *k*NN

[6]We embed the text and entities of samples using a Sentence-BERT (Reimers and Gurevych, 2019) model trained on PubMed corpus (https://huggingface.co/pritamdeka/S-PubMedBert-MS-MARCO)

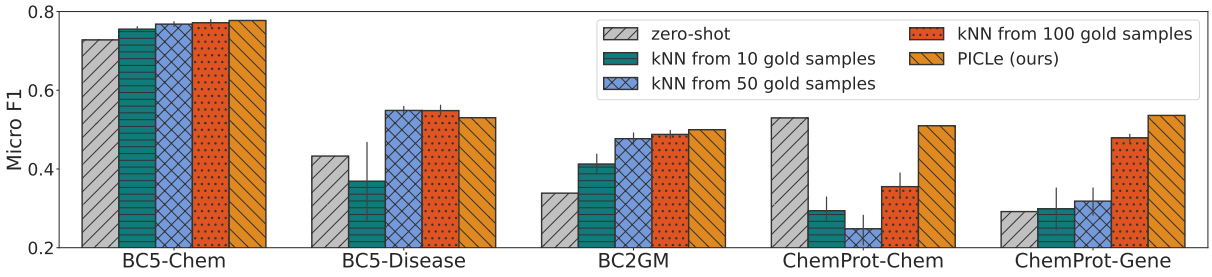

Figure 4: **Performance of PICLe, zero-shot, and ICL with gold demonstrations** selected from $10, 50, 100$ gold examples using Mistral. The error bars show the variance across five seeds for sampling subsets of gold examples. All methods are followed by self-verification unless otherwise specified.

(following ablation study in Figure 8 in Appendix). We experiment with $N \in [10, 50, 100]$, reporting results for a diverse set of annotation budgets. We repeat all experiments with 5 seeds and report the average performance across these runs along with standard deviations.

Our results in Figure 4 show that across most datasets (with the exception of *ChemProt-Chem*), PICLe significantly outperforms the zero-shot baseline by an average of 10.7% (57.1% compared to 46.4%). Furthermore, PICLe also matches or outperforms in-context learning with gold demonstrations in resource-scarce settings, even beating an in-context learning baseline that has access to 100 human-annotated demonstrations (57.1% vs. 52.8%). We note that the dataset with the highest performance, BC5chem (77.7% average F1 score for PICLe), contains entity annotations whose surface forms generally contain fewer tokens (see Table 1). On the contrary, the dataset with the lowest performance, BC2GM (50% F1 for PICLe), has entity annotations that contain longer surface forms, making it more difficult to match the exact span in a generative manner.

We also compare PICLe with a supervised baseline, fine-tuning a domain-specific language encoder, `BiomedNLP-BiomedBERT-large`, on various numbers of gold annotations (see Table 9 in Appendix). While the performance of fine-tuning on 10 gold samples is low and shows high variance between datasets, the performance with 50 gold samples already outperforms all LLM baselines. However, we note that the sequence labeling formulation of the task for the supervised baseline differs from the generative formulation for LLMs, providing the supervised baseline with a simpler format for predicting entity spans, more adapted to our strict exact match evaluation.

Ablation study In Table 3, we ablate each step of the PICLe pipeline to evaluate the importance of each component (see Appendix Table 5 for a detailed version including precision and recall). For pseudo-annotation, similarly to Wan et al. (2023a), we experiment with running zero-shot prediction 10 times with high temperature ($T = 0.7$) and filtering the 10 sets of extracted entities using self-verification or merging (i.e., prompting the LLM to aggregate the entities lists). Both lead to slightly lower F1, while being much more computationally expensive than a single round of zero-shot prediction. We also find that self-verification helps with validating pseudo-annotations (row #3 vs. row #8).

For demonstration retrieval, we compare random, kNN, vanilla k-means, and specialized k-means (Sp-k-means) as in-context example sampling methods. kNN ($k = 10$), known for being sensitive to noisy demonstrations (Zhang et al., 2022), scores the lowest (row #4). For random retrieval, we sample demonstrations using 5 different seeds; the predicted entity lists are merged and post-processed using self-verification (row #5). Similarly, for k-means (row #6), we randomly sample one demonstration per cluster, increasing the intra-run diversity. Conversely, in Sp-k-means, demonstrations in each round are all sampled from the same cluster, maximizing inter-run diversity. We sample demonstrations using 5 different seeds, leading to 5 inference runs. The predicted entity lists are merged and self-verified again. The diversity of demonstrations for k-means leads to a higher recall than random (48.6 vs. 40%), but not as high as the one from Sp-k-means (53.5%), which benefits greatly from having separate "expert" clusters that lead to more varied predictions. Self-verification improves performance during inference (rows #7 vs. #8), especially in terms of precision (+20%).

	Pseudo-annotation		Inference		F1
	Runs	Post-processing	Demo retrieval	SV	
1	10	merging		✓	55.7
2	10	SV	Sp-k-means	✓	55.1
3	1	none		✓	51.8
4	1		kNN	✓	42.7
5	1	SV	random	✓	47.9
6	1		k-means	✓	55.1
7	1	SV	Sp-k-means	×	49.2
8	1			✓	**57.1**
9			Zero-shot	×	44.6
10	NA	NA	Zero-shot	✓	46.4
11			10 Zero-shot	✓	50.6

Table 3: **Ablation of each component of PICLe**, averaged over all 5 datasets, using Mistral. SV refers to the use of self-verification.

Demonstration retrieval	Inference	F1
Llama2 + PICLe	Llama2	51.9
Zero-shot		48.3
10 gold samples		45.5
100 gold samples + kNN		53.6
Mistral + PICLe	Mistral	57.1
Zero-shot		46.4
10 gold samples		42.6
100 gold samples + kNN		52.8
Full train set + kNN (oracle)		63.2
GPT-3.5-Turbo + PICLe	Mistral	56.5

Table 4: **Performance of PICLe using different LLMs for pseudo-annotation and prediction**, compared with zero-shot and ICL with gold annotations.

6 Conclusion

In this work, we study the demonstration attributes that enable in-context generalization for named entity detection. We find that the context-label semantic correspondence is crucial for effective in-context NED, and without this correspondence, in-context examples hurt performance, pushing it below zero-shot NED. However, our analysis demonstrates that partially-correct demonstration label sets are just as effective as gold label sets, provided a sufficient number of correct label mappings are found in the demonstration. Based on these findings, we design an ICL framework, PICLe, for named entity detection that leverages LLMs to produce pseudo-annotated examples that can be used for in-context transfer. Our results on five biomedical NED datasets demonstrate that PICLe is more effective than zero-shot prediction and outperforms gold in-context learning in simulated real-world settings where gold demonstrations are scarce due to the effort and expertise required for annotation.

7 Limitations

Single Task. This work introduces a method to alleviate annotation effort for the NED task while achieving comparable performance to few-shot NED with human-labeled annotations. While this pipeline can be generalized to other tasks with an open-ended nature similar to NED, the experiments presented in this paper are limited to the NED task. However, we demonstrate its effectiveness over a broad set of entity types. Similarly, further work is needed to generalize our conclusions on the partial correctness of demonstrations to all structured output tasks.

Sensitive applications. We apply our system to documents from the biomedical domain. The evaluation sets are drawn from abstracts from published articles. However, the tools we develop can be used to extract the same type of entities in more sensitive documents, such as extracting diseases from patient records. Our tools were not tested for these applications, and practitioners should be aware that performance on such different types of documents is not guaranteed to transfer.

Annotation bias. Annotated data can contain various forms of annotation bias, which lead trained models to make biased predictions when labeling entities based on the knowledge and beliefs of the annotators. This bias is usually alleviated following common annotation practices such as computing inter-rater agreement and having detailed annotation guidelines discussed with the annotators. However PICLe only uses models' pseudo-annotations, since we focus on domains for which expert annotation is challenging to obtain. Consequently, given the lack of interpretability and training data openness of the used LLMs, we cannot assess the reliability and fairness of the demonstrations.

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Jiawei Chen, Yaojie Lu, Hongyu Lin, Jie Lou, Wei Jia, Dai Dai, Hua Wu, Boxi Cao, Xianpei Han, and Le Sun. 2023a. Learning in-context learning for named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13661–13675, Toronto, Canada. Association for Computational Linguistics.

Wei-Lin Chen, Cheng-Kuang Wu, Yun-Nung Chen, and Hsin-Hsi Chen. 2023b. Self-ICL: Zero-shot in-context learning with self-generated demonstrations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15651–15662, Singapore. Association for Computational Linguistics.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv*, abs/2301.00234.

Yu Fei, Yifan Hou, Zeming Chen, and Antoine Bosselut. 2023. Mitigating label biases for in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14014–14031, Toronto, Canada. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428, Toronto, Canada. Association for Computational Linguistics.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023. Large language models can self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1051–1068, Singapore. Association for Computational Linguistics.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023a. Mistral 7b. *ArXiv*, abs/2310.06825.

Zhongtao Jiang, Yuanzhe Zhang, Cao Liu, Jun Zhao, and Kang Liu. 2023b. Generative calibration for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2312–2333, Singapore. Association for Computational Linguistics.

Dong-Ho Lee, Akshen Kadakia, Kangmin Tan, Mahak Agarwal, Xinyu Feng, Takashi Shibuya, Ryosuke Mitani, Toshiyuki Sekiya, Jay Pujara, and Xiang Ren. 2022. Good examples make a faster learner: Simple demonstration-based learning for low-resource NER. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2687–2700, Dublin, Ireland. Association for Computational Linguistics.

Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. 2016. Biocreative V CDR task corpus: a resource for chemical disease relation extraction. *Database: The Journal of Biological Databases and Curation*.

Xiaonan Li and Xipeng Qiu. 2023. MoT: Memory-of-thought enables ChatGPT to self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6354–6374, Singapore. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? In *Workshop on Knowledge Extraction and Integration for Deep Learning Architectures; Deep Learning Inside Out*.

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 *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Xinxi Lyu, Sewon Min, Iz Beltagy, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. Z-ICL: Zero-shot in-context learning with pseudo-demonstrations. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2304–2317, Toronto, Canada. Association for Computational Linguistics.

Yubo Ma, Yixin Cao, Yong Hong, and Aixin Sun. 2023. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10572–10601, Singapore. Association for Computational Linguistics.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2022. Reframing instructional prompts to GPTk's language. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 589–612, Dublin, Ireland. Association for Computational Linguistics.

Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. 2022. Impact of pretraining term frequencies on few-shot numerical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 840–854, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Yongliang Shen, Zeqi Tan, Shuhui Wu, Wenqi Zhang, Rongsheng Zhang, Yadong Xi, Weiming Lu, and Yueting Zhuang. 2023. PromptNER: Prompt locating and typing for named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12492–12507, Toronto, Canada. Association for Computational Linguistics.

Larry L. Smith, Lorraine K. Tanabe, Rie Ando, Cheng-Ju Kuo, I-Fang Chung, Chun-Nan Hsu, Yu-Shi Lin, Roman Klinger, C. Friedrich, Kuzman Ganchev, Manabu Torii, Hongfang Liu, Barry Haddow, Craig A. Struble, Richard J. Povinelli, Andreas Vlachos, William A. Baumgartner, Lawrence E. Hunter, Bob Carpenter, Richard Tzong-Han Tsai, Hong-Jie Dai, Feng Liu, Yifei Chen, Chengjie Sun, Sophia Katrenko, Pieter W. Adriaans, Christian Blaschke, Rafael Torres, Mariana L. Neves, Preslav Nakov, Anna Divoli, Manuel Maña-López, Jacinto Mata, and W. John Wilbur. 2008. Overview of biocreative ii gene mention recognition. *Genome Biology*, 9:S2 – S2.

Taylor Sorensen, Joshua Robinson, Christopher Rytting, Alexander Shaw, Kyle Rogers, Alexia Delorey, Mahmoud Khalil, Nancy Fulda, and David Wingate. 2022. An information-theoretic approach to prompt engineering without ground truth labels. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 819–862, Dublin, Ireland. Association for Computational Linguistics.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Simeng Sun, Kalpesh Krishna, Andrew Mattarella-Micke, and Mohit Iyyer. 2021. Do long-range language models actually use long-range context? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 807–822, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Olivier Taboureau, Sonny Kim Nielsen, Karine Audouze, Nils Weinhold, Daniel Edsgärd, Francisco S. Roque, Irene Kouskoumvekaki, Alina Bora, Ramona Curpan, Thomas Skøt Jensen, Søren Brunak, and Tudor I. Oprea. 2010. Chemprot: a disease chemical biology database. *Nucleic Acids Research*, 39:D367 – D372.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Xingchen Wan, Ruoxi Sun, Hanjun Dai, Sercan Arik, and Tomas Pfister. 2023a. Better zero-shot reasoning with self-adaptive prompting. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3493–3514, Toronto, Canada. Association for Computational Linguistics.

Xingchen Wan, Ruoxi Sun, Hootan Nakhost, Hanjun Dai, Julian Eisenschlos, Sercan Arik, and Tomas

Pfister. 2023b. Universal self-adaptive prompting. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7437–7462, Singapore. Association for Computational Linguistics.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2023a. Towards understanding chain-of-thought prompting: An empirical study of what matters. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2717–2739, Toronto, Canada. Association for Computational Linguistics.

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. 2024. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning. *Advances in Neural Information Processing Systems*, 36.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

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

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. 2023. Large language models are better reasoners with self-verification. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2550–2575, Singapore. Association for Computational Linguistics.

Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. 2022. A survey on deep semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):8934–8954.

Shaokai Ye, Jessy Lauer, Mu Zhou, Alexander Mathis, and Mackenzie W Mathis. 2023. Amadeusgpt: a natural language interface for interactive animal behavioral analysis. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Kang Min Yoo, Junyeob Kim, Hyuhng Joon Kim, Hyunsoo Cho, Hwiyeol Jo, Sang-Woo Lee, Sang-goo Lee, and Taeuk Kim. 2022. Ground-truth labels matter: A deeper look into input-label demonstrations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2422–2437, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*.

Tony Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International Conference on Machine Learning*.

A Reproducibility statement

Code. We provide all the prompts used in our study in the Appendix D.1, and the full code will be released upon publication. Note that small portions of the code were developed with the assistance of GitHub Copilot. We also provide the random seeds used for random sampling of demonstration in the Appendix C.1. For our experiment, we use default parameters, unless specified (*e.g.* temperature for pseudo-annotation and inference). All models were used for inference only, on a single NVIDIA A100 GPU with 32 GB Memory, each inference run taking between 5 and 20 minutes depending on the dataset.

Data. The datasets we use are publicly available on the `Huggingface` platform.[7]

Models. As described in Section 3, we use two open-source models for our studies whose checkpoints can be found in `Huggingface`: Mistral-7b-instruct[8] and Llama-2-7b-Chat.[9] We also conduct experiments using a proprietary LLM from OpenAI, `gpt-3.5-turbo-0125`,[10] which unfortunately is subject to be updated (or removed from the API entirely) at any moment, limiting the long-term reproducibility of the results obtained with this tool. For supervised fine-tuning, we use the text encoder `BiomedNLP-BiomedBERT-large`.[11]

B Additional analysis for ICL demonstration

B.1 Corrupted random demonstrations

On top of the perturbation schemes defined in the main body of the paper, we define two additional methods:

`Random OOD Label from Text`: We replace ground-truth entity labels with words randomly selected from the sample's text that are not included in the ground-truth annotation (i.e., not a target entity).

`Swapped ID Labels`: We swap entity labels in the ground-truth demonstrations with the entity labels of a randomly chosen sample in the training split. Contrary to `Random ID Label` where the *number* of entities is preserved, the number of entities in each ground-truth annotation changes compared to the original ground-truth.

Figure 5 shows results per dataset for all corruption schemes, with GPT-3.5-Turbo and Mistral models.

B.2 Partially correct demonstrations

Figure 6 shows the evolution of the downstream F1 depending on the number of entities in the demonstrations and the perturbation factor. As expected, an increased perturbation factor leads to a lower demonstration F1 and a lower downstream F1 (right side of the figure). Similarly, adding or removing entities in the demonstration labels leads to a lower downstream F1. However, with the same perturbation factor, perturbations that do not decrease the number of entities in the demonstration (addition&substitution, substitution) lead to a much softer rate of performance loss. Similarly, to reach the same downstream performance as zero-shot (around 0.5 on average), removing one entity is enough, while at least two entities need to be added. This result supports the hypothesis that a way to increase downstream performance is to give preference to a higher recall and number of entities in the demonstration set.

Figure 7 compares the precision and recall of demonstrations against the precision and recall of predictions.

C Experimental setup details

C.1 Random Experiment Seeds

We repeat all of our experiments that involve randomization with 5 times with the following seeds: $[12345, 24690, 37035, 49380, 61725]$

D Additional results for PICLe

D.1 Prompts

In this section, we provide examples of prompts used in our experiments.

[7] https://huggingface.co/datasets/bigbio/

[8] https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1

[9] https://huggingface.co/meta-llama/Llama-2-7b-chat-hf

[10] https://platform.openai.com/docs/models/gpt-3-5-turbo

[11] https://huggingface.co/microsoft/BiomedNLP-BiomedBERT-large-uncased-abstract

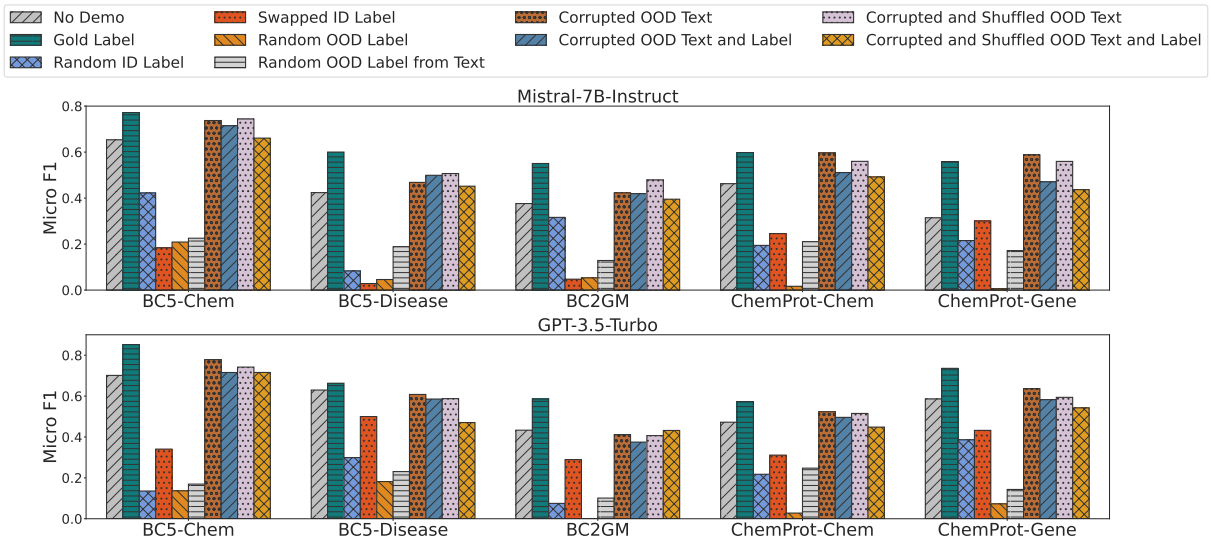

Figure 5: **10-shots ICL performance using various demonstration corruption schemes**, compared with zero-shot and ICL with gold annotations, for each dataset. We use Mistral (top) and GPT-3.5-Turbo (bottom).

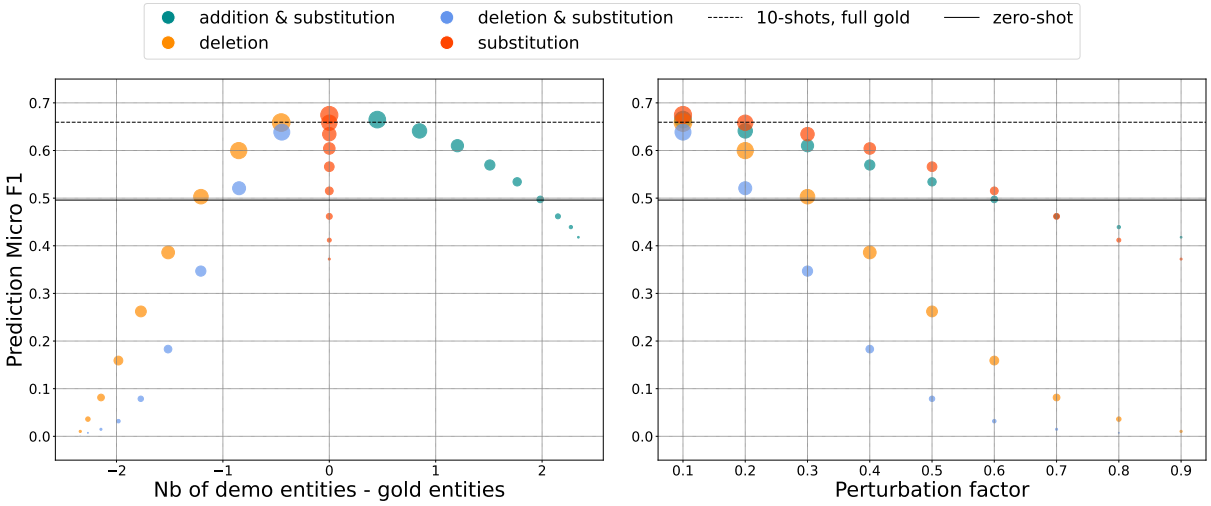

Figure 6: **10-shot ICL performance with partially correct demonstrations using different perturbation schemes** and *k*NN demonstration retrieval. We observe the impact on Prediction F1 of the perturbation factor and the number of entities in the demonstrations for different perturbation types, averaged over all datasets. The size of the points shows the demonstrations' F1 score.

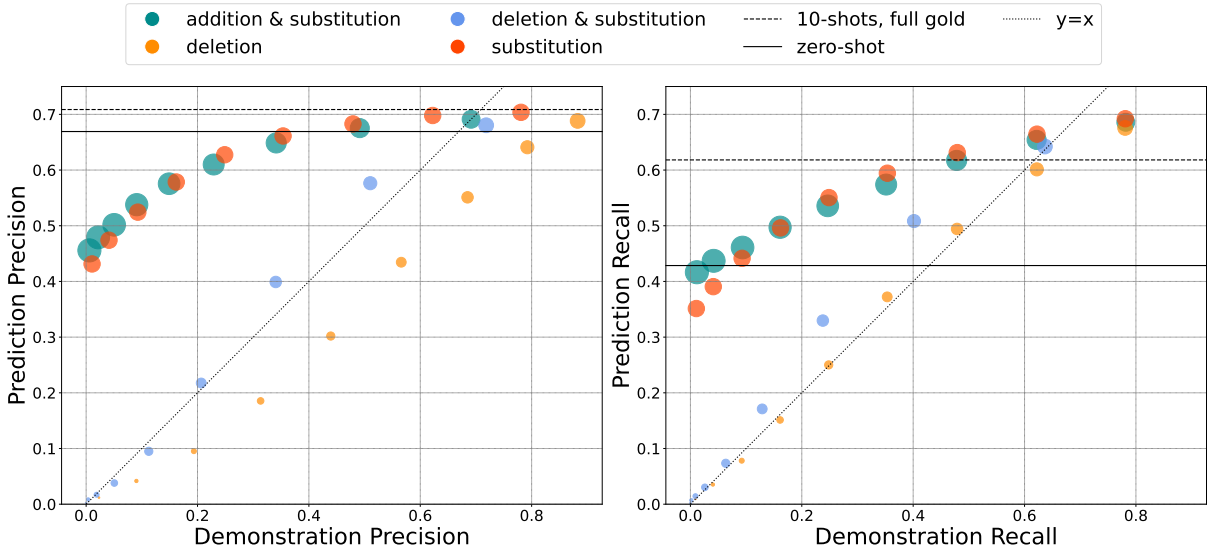

Figure 7: **10-shot ICL performance with partially correct demonstrations using different perturbation schemes** and *k*NN demonstration retrieval. We observe the impact of the demonstration precision and recall on the downstream prediction precision and recall, respectively, averaged over all datasets. The size of the points shows the demonstrations' number of entities in the annotations.

Examples of prompt used for self-verification with the BC5-Chem dataset.

user: Given the context and definition of chemical entity, answer the following question. Please reason about your answer and include YES or NO in your response. YES if the given phrase is a chemical entity, and NO if it is not. If you are not sure, you can say I don't know.
assistant: Understood.
user: Context: p75NTR expression in rat urinary bladder sensory neurons and spinal cord with cyclophosphamide - induced cystitis.
chemical definition: Chemical refers to any substance having a distinct molecular composition that is produced by or used in a chemical process. Chemicals can be elements or compounds, and they can exist in various forms—solid, liquid, or gas.
Based on this context and definition, does cyclophosphamide correspond to the name of a chemical entity?

Example of prompt used for NED with the BC5-Chem dataset.

user: Your task is to extract all of the chemicals mentioned in a given abstract published in PubMed. Chemical refers to any substance having a distinct molecular composition that is produced by or used in a chemical process. Chemicals can be elements or compounds, and they can exist in various forms—solid, liquid, or gas.
Please extract all of the entities corresponding to chemicals from the following paragraph, the same way as they are marked in the examples. Make sure to include all and only the chemicals mentioned in the text, but not the genes and proteins, and diseases. If there are no chemical entities in the text output 'None'. You will be penalized if you include an entity more or less than the number of times it appears in the text.
assistant: understood! let's get started!

user: ## Here are some examples: input: cTnT found in rats after 12 mg / kg were significantly greater than that found after 7 . 5 mg / kg DOX . chemical entities:
assistant: DOX | yes | because it is a chemical

user: input: MMP - 2 , MMP - 9 , ADAM - 10 and ADAM - 17 mRNA levels were increased in CaCl (2) - treated segments (all p < 0 . 01) , with trends of elevation in CaCl (2) - untreated segments , as compared with NaCl - treated segments . chemical entities:
assistant:
CaCl (2) | yes | because it is a chemical
CaCl (2) | yes | because it is a chemical
NaCl | yes | because it is a chemical

user: input: The risk of bladder cancer doubled for every 10 g increment in cyclophosphamide (OR = 2 . 0 , 95 chemical entities:
assistant: cyclophosphamide | yes | because it is a chemical

user: input: Capsaicin - induced muscle pain alters the excitability of the human jaw - stretch reflex . chemical entities:
assistant:
Capsaicin | yes | because it is a chemical

user: input: The normalized reflex amplitude was significantly higher during pain , but only at faster stretches in the painful muscle . chemical entities:
assistant:

user: input: No statistically significant changes in behavior or receptor binding were found in PD males with the exception of increased (3) H - MK - 801 binding in cortex . chemical entities:
assistant:
H | yes | because it is a chemical
MK - 801 | yes | because it is a chemical

user: input: NRA0160 has negligible affinity for the human cloned dopamine D3 receptor (Ki = 39 nM) , rat serotonin (5 - HT) 2A receptors (Ki = 180 nM) and rat alpha1 adrenoceptor (Ki = 237 nM) . chemical entities:
assistant:
NRA0160 | yes | because it is a chemical
dopamine | yes | because it is a chemical
serotonin | yes | because it is a chemical
5 - HT | yes | because it is a chemical

user: input: p75NTR expression in rat urinary bladder sensory neurons and spinal cord with cyclophosphamide - induced cystitis . chemical entities:

D.2 Ablation study

D.3 Results with different models

D.4 Inference with gold demonstrations

Here, we compare random, k-means, and kNN demonstration retrieval methods for gold demonstrations with and without the self-verification step. (Figure 8).

D.5 PICLe Pseudo-annotation Ablation

D.6 Fine-tuning results

We report the performance of fine-tuned PubMedBERT in Table 9.

	Pseudo-annotation		Inference		Precision	Recall	F1
	Runs	**Post-processing**	**Demo retrieval**	**Self-verif**			
1	10	LLM-merging	Sp-*k*-means	✓	56.7	55.1	55.7
2	10	self-verif	Sp-*k*-means	✓	56.3	54.2	55.1
3	1	none	Sp-*k*-means	✓	55.2	49.4	51.8
4	1	self-verif	*k*NN	✓	72.5	32.8	42.7
5	1	self-verif	random	✓	68.1	39.8	47.9
6	1	self-verif	*k*-means	✓	64.8	48.6	55.1
7	1	self-verif	Sp-*k*-means	×	41.8	60.7	49.2
8	1	self-verif	Sp-*k*-means	✓	61.8	53.5	**57.1**

Table 5: Ablation of each component of PICLe, averaged over all datasets, using Mistral for pseudo-annotation and inference.

Demonstration pool	**Demo retrieval**	**Inference model**	**Precision**	**Recall**	**F1**
Llama2 PICLe	Sp-*k*-means	Llama2	47.0	59.9	51.9
	zero-shot		59.5	40.8	48.3
10 gold samples			59.2	38.6	45.5
100 gold samples	*k*NN		60.0	48.7	53.6
Mistral PICLe	Sp-*k*-means	Mistral	61.8	53.5	57.1
	zero-shot		68.7	37.7	46.4
10 gold samples			65.7	34.9	42.6
100 gold samples	*k*NN		73.6	42.5	52.8
GPT-3.5-Turbo PICLe	Sp-*k*-means	Mistral	65.2	50.1	56.5

Table 6: Performance of PICLe compared with using 10 and 100 annotated gold samples, with different models used for pseudo-annotation and prediction.

Dataset	**SV**	**Precision**	**Recall**	**Micro F1**
BC2GM	×	51.9	27.7	36.1
	✓	58.4	22.2	32.2
BC5-Chem	×	60.2	71.7	65.4
	✓	81.5	65.6	72.7
BC5-Disease	×	57.1	33.3	42.1
	✓	69.9	31.3	43.2
ChemProt-Chem	×	34.6	54.7	42.4
	✓	53.1	49.9	51.5
ChemProt-Gene	×	69.1	20.7	31.9
	✓	77.6	18.3	29.6
Average	×	54.6	41.6	43.6
	✓	68.1	37.5	45.8

Table 7: **Evaluation of pseudo-annotated samples with and without self verification.** The pseudo-annotations obtained via zero-shot with zero temperature. SV refers to the use of self-verification.

Dataset	**SV**	**Precision**	**Recall**	**Micro F1**
BC2GM	×	53.0	29.2	37.6
	✓	59.7	23.6	33.9
BC5-Chem	×	60.4	71.2	65.3
	✓	82.3	65.2	72.8
BC5-Disease	×	53.9	34.9	42.4
	✓	67.3	31.9	43.3
ChemProt-Chem	×	39.4	56.1	46.3
	✓	56.7	49.7	53.0
ChemProt-Gene	×	68.8	20.4	31.4
	✓	77.4	18.0	29.2
Average	×	55.1	42.4	44.6
	✓	68.7	37.7	46.4

Table 8: **Evaluation of zero-shot inference with and without self verification.** The temperature is set to zero. SV refers to the use of self-verification.

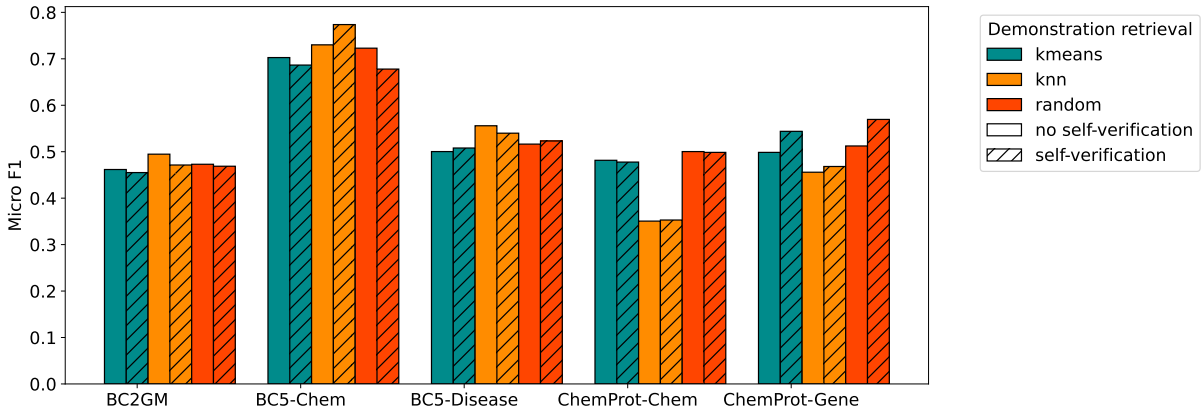

Figure 8: **Ablation study of baselines with gold annotations.**

Train set size	BC2GM	BC5-Chem	BC5-Disease	ChemProt-Chem	ChemProt-Gene	Average
10	9.8	59.0	15.1	64.3	50.1	39.7
50	55.7	79.1	53.0	79.6	72.4	67.9
100	63.9	83.4	65.6	83.4	77.6	74.8
Full	87.0	94.3	85.4	90.8	89.8	89.5

Table 9: **Micro-F1 score of PubMedBERT-large fine-tuned on various numbers of gold annotations**. For 10, 50 and 100 gold annotations, random sets are sampled with 5 different seeds, and the fine-tuning performances are averaged.