In-Context Learning on a Budget: A Case Study in Token Classification

Anonymous ACL submission

Abstract

Few shot in-context learning (ICL) typically assumes access to large annotated training sets. However, in many real world scenarios, such as domain adaptation, there is only a limited budget to annotate a small number of samples, with the goal of maximizing downstream performance. We study various methods for selecting samples to annotate within a predefined budget, focusing on token classification tasks, which are expensive to annotate and are relatively less studied in ICL setups. Across different tasks, models and datasets, we find that a relatively small pool of annotated samples can achieve results comparable to the entire training set. Finally, we discover that random selection of samples for annotation yields surprisingly good performance. We hope that future work adopts our realistic paradigm which takes annotation budget into account.

1 Introduction

In-context learning (ICL) has emerged as a highly efficient and robust method for various textual tasks. In this paradigm, a large language model (LLM) is exposed to a small number of annotated samples, termed *demonstration examples*, which are provided as part of the prompt, before the sample which the model is required to annotate, which we will refer to as *inference sample* henceforth. While the reasons for ICL's success are still contested (Min et al., 2022; Liu et al., 2022), it has been observed that ICL prompts commonly outperform zero-shot prompts, where no demonstration examples are provided (Brown et al., 2020).

Furthermore, a recent line of work has found that the choice of demonstration examples can lead to improved results over random demonstration selection. For example, Liu et al. (2022) found that choosing the nearest neighbors of the inference sample in the training set leads to improvements over random demonstration selection on 6 tasks, such as sentiment analysis or question answering. In all of these, the demonstration examples are chosen from large annotated training sets, ranging from 3.5K samples up to 78K samples.

In this work, we address the following research question: *How can we maximize ICL performance on a given annotation budget?* This question is particularly relevant for real-world domain adaptation settings, where a large pool of annotated samples is unavailable for selecting demonstration examples. Instead, there are large sets of unannotated samples (e.g., raw text in the target domain), and a fixed budget to annotate only a small portion of them. As depicted in Figure 1, we define the task as *pool selection*, i.e., selecting a small pool of k examples out of a large corpus of raw texts. These samples are annotated and serve as the available pool for demonstration examples.

We implement several methods for pool selection, e.g., clustering the train set and selecting a representative example from each cluster, and test them on three token classification tasks: named entity recognition, dependency parsing, and part-of-speech tagging. We select these tasks as they are relatively understudied in the context of ICL and are expensive to annotate due to the need for linguistic expertise and domain-specific knowledge (Chen et al., 2015; Zhang et al., 2017).

We evaluate several state-of-the-art LLMs on token classification benchmarks. We find that a relatively small pool ($\sim$200 samples) allows LLMs to perform over 88% as when demonstrations are selected from the full training set. Additionally, we find that in some scenarios *randomly selecting* samples for the annotation pool performs similarly to more carefully designed methods.

We hope that our paradigm is adopted in future work in order to report more realistic ICL performance, and explore new methods for sample selection for other tasks and domains.

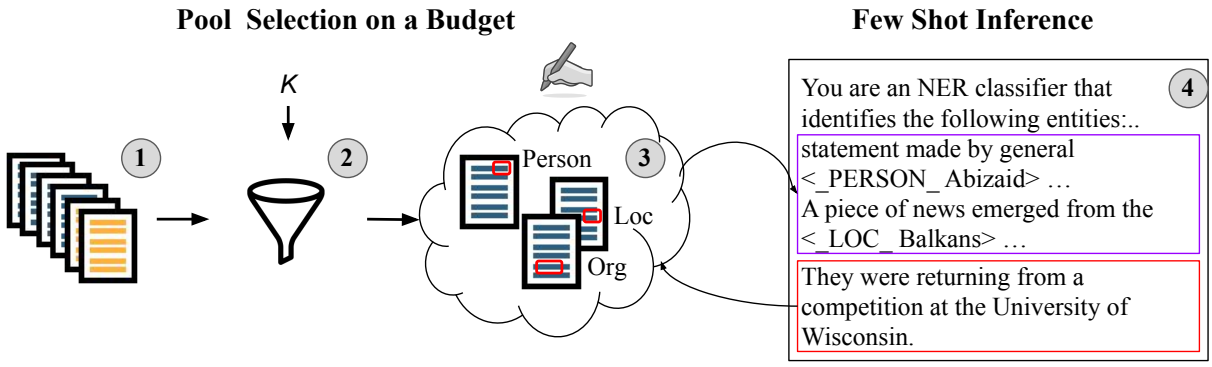

Figure 1: Our proposed approach for ICL on a budget, illustrated in four steps: (1) we assume a large pool of *raw* train and potentially test texts in the target domain; (2) a *pool selection strategy* chooses a subset of k train texts to maximize downstream ICL performance; (3) annotations are collected for the selected pool; (4) an inference prompt is instantiated by choosing the nearest examples in the pool to the inference sample. We focus on step (2), experimenting with various pool selection strategies.

2 ICL on a Budget

Here we propose a conceptual framework for ICL in a realistic domain adaptation setting (depicted in Figure 1), where there are no apriori annotated datasets for the target task. Instead, we assume that there is a large corpus of raw texts in the target domain, and a fixed budget for annotating a small portion of them, such that they can serve as potential demonstration examples during inference. Intuitively, the goal of the annotation process is to maximize downstream ICL performance.

Below we formalize the task of pool selection, and describe 4 selection strategies, which aim to maximize different aspects, e.g., coverage of the training set versus coverage of the test set. In the following section we evaluate these approaches for token classification tasks.

2.1 Pool Selection: Task Definition

Formally, a *pool selection strategy* is a function:

$$S_k : \mathcal{P}(\mathcal{D}) \mapsto \mathcal{D}^k \quad (1)$$

Where $\mathcal{D}$ represents an unannotated distribution (e.g., all texts in a certain domain), $\mathcal{P}(\mathcal{D})$ is the power set of $\mathcal{D}$, and $k \in \mathbb{N}_+$ is the annotation budget, i.e., the number of samples to annotate. Intuitively, S maps raw sample sets to k train samples (the pool), which are then annotated. In all that follows we denote the input samples set by $D \in \mathcal{P}(\mathcal{D})$. Typically, $k << |D|$, indicating that the annotation budget for a new domain can only annotate a small portion of its available texts.

Furthermore, we assume a similarity function ϕ:

$$\phi : \left(\mathcal{D} \times \mathcal{D}\right) \mapsto \mathbb{R} \quad (2)$$

In the scope of this work (similar to previous work (Liu et al., 2022)) samples are embedded into $\mathbb{R}^m$, where m is the text embedding dimension, and ϕ is the cosine similarity of these vectors. In particular, we use a sentence transformer (Reimers and Gurevych, 2019) trained over MPNet (Song et al., 2020).

2.2 Pool Selection Strategies

Below we describe 4 pool selection strategies which follow the definition in Equation 1.

Central. Select the k samples from D that are closest to the Euclidean center of D.

Cluster. Cluster D into k clusters and for each cluster center choose the most similar sample from D. This method aims to maximize the coverage of the expected training distribution. This strategy was proposed in Chang et al. (2021) for selecting examples for fine-tuning.

Vote-k (Su et al., 2022). Selects a k-sized subset of D such that the samples are diverse and dissimilar from one another, through a two-step process which uses the LLM's confidence to bucket the different samples. See Su et al. (2022) for more details.

Random. Randomly select k samples from D.

3 Evaluation

We focus on token classification tasks, as they are both understudied in the context of ICL, while traditionally expensive to annotate at scale. Importantly, as is commonly done in active learning scenarios (e.g., Shen et al., 2018; Liu et al., 2020), we

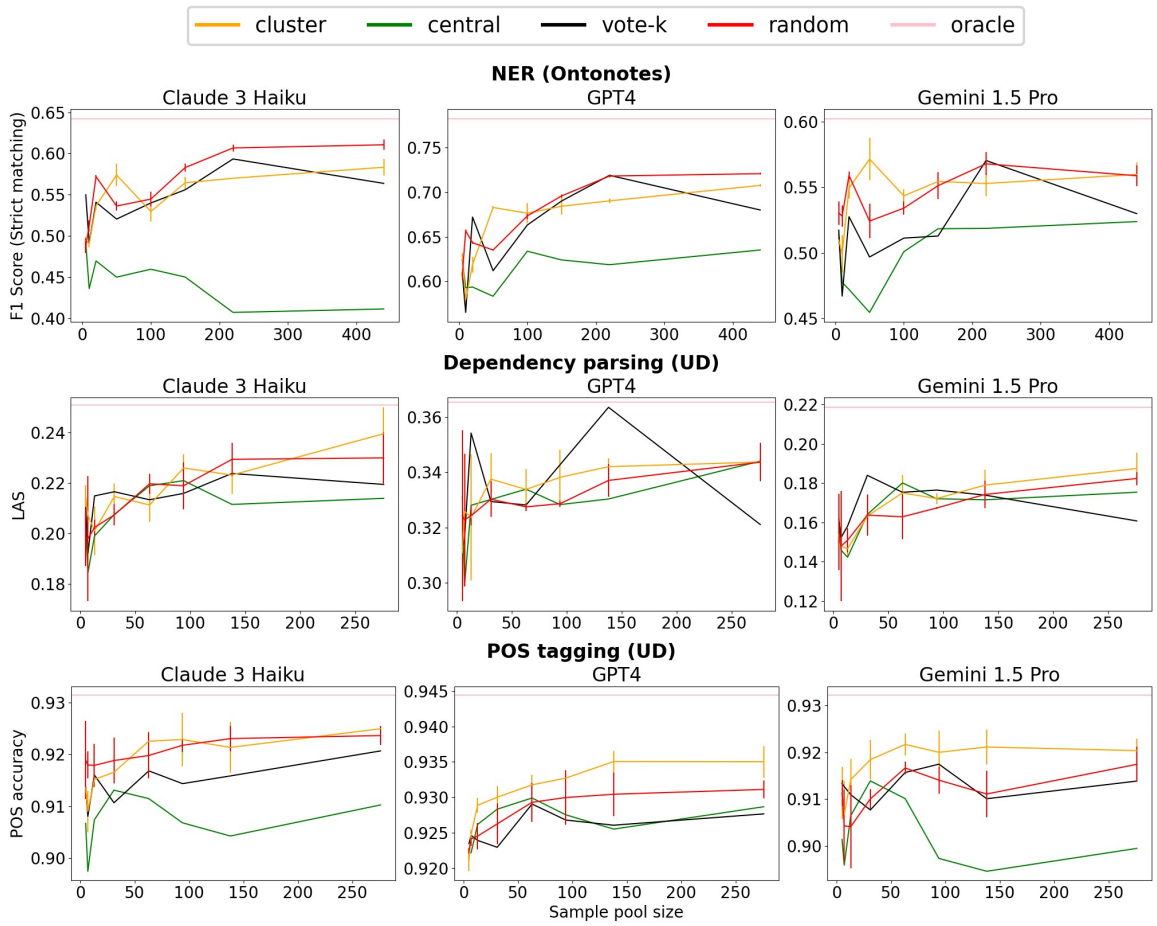

Figure 2: Results for NER (top), dependency parsing (center), and POS tagging (bottom) using Claude 3 Haiku (left), GPT4 (middle) and Gemini 1.5 Pro (right), as a function of the size of sample pool. In methods with a random component we run 3 trials and plot error bars showing the standard deviation. Oracle: using the full training set as the pool.

use existing datasets to *simulate* the process shown in Figure 1, where instead of collecting manual annotations, we sample from the existing annotations.

3.1 Experimental Setup

Tasks, datasets, and models. We test three tasks using two English datasets: Ontonotes 5.0 for NER (Hovy et al., 2006), and the Universal Dependencies corpus for dependency parsing and POS tagging (UD; Nivre et al., 2016).[1] Both datasets contain texts from various domains, including news, conversational, weblogs, web forums, and more. Due to budget constrains we limit the size of each test set by randomly sampling 1000 samples. We experiment with a diverse set of 9 different LLMs, ranging from open to closed models in various parameter sizes: Llama-2 (Touvron et al., 2023), Mistral (Jiang et al., 2023), Starling (Zhu et al., 2023), Vicuna (Chiang et al., 2023), Mixtral (Jiang et al., 2024), phi-2 (Javaheripi et al., 2023), GPT-4 (Achiam et al., 2023), Claude 3 Haiku,[2] and Gemini 1.5 Pro (Reid et al., 2024).

Prompt and evaluation metric. For each task, we prompt the models by describing the task and the expected output format, followed by 5 demonstration examples and the current inference sample. Since POS tagging serves as a precursor to dependency parsing, both tasks are handled using a shared prompt. Following Zhao et al. (2021), we sort the examples such that the most similar example is the last one, based on the similarity function ϕ described in Equation 2. See Appendix A for the full prompt templates.

For NER evaluation we use *strict* matching (Segura-Bedmar et al., 2013), where a predicted

[1] We use the HuggingFace versions for both datasets with the *english_v12* configuration for Ontonotes and the *en_ewt* configuration for UD.

[2] www.anthropic.com/news/claude-3-family

entity is considered correct if it matches both exact span boundaries and entity type. For dependency parsing we use *labeled attachment score (LAS)*, which measures the accuracy of both the correct head and dependency label for each token. For POS tagging, we use the *POS accuracy*.

Pool sizes. We experiment with sample pools of size 0.1% - 10% of the maximum pool size, defined as the number of unique samples used when considering the full training set as the sample pool.

Oracle. To better understand the performance of different pool selection strategies, we report the performance of using the full training set for each dataset and model, compared to our budget-constrained approach.

3.2 Results

Results for the three best-performing models (Claude 3 Haiku, GPT-4, and Gemini 1.5 Pro) on all tasks are presented in Figure 2. The other smaller models we test are not able to produce outputs of the requested format in more than 50% of the cases, and hence cannot be meaningfully compared against these models, which adhere to the correct format in roughly 97% of the cases. We now discuss findings reflected in these results and conduct further analysis.

The choice of few-shot examples matter in token classification tasks. We observe a large variation in performance when selecting different demonstration examples in all configurations. While this was observed in other tasks (Zhang et al., 2022), to the best of our knowledge, this is the first time this was shown for token classification tasks.

Most pool selection methods perform similarly, random is surprisingly good. None of the four methods consistently outperform the others. Surprisingly, we note that random performs similarly to other methods. We note that this is true even for selection strategies that are allowed access to the test set distribution.

Very small pool sizes can approximate the full training corpus. All sampling methods require a pool of only 220 samples for Ontonotes, and 138 samples for UD, to achieve over 88% of the oracle performance across all configurations.

ICL on a budget lags behind state-of-the-art. We compare the ICL results to state-of-the-art results in Table 1. In NER and dependency parsing, fine-tuned methods vastly outperforms using a limited annotation budget. In POS tagging which is considered an easier task, using 138 samples achieves 96% of the state-of-the-art.

	NER	DP	POS
Cluster train	75	41	96
Cluster test	76	40	96
Largest mean	67	41	96
Vote-k	78		
Random	78	40	96

Table 1: GPT-4 percentage in performance out of the state-of-the-art, when using 5% of the samples used in the fully labeled train set method.

4 Related Work

Similar to pool selection, active learning (Shen et al., 2017) also aims to select samples for annotation rather than assuming all samples are annotated. However, active learning operates during training and relies on access to an oracle or intermediate model results (e.g., confidence scores), whereas pool selection assumes no access to the model during training and only relies on observing the outputs of the model.

Recently, Su et al. (2022) introduced a pool selection method as a method for improving downstream performance, which we evaluate as one of our approaches for annotation pool selection. Our work is conceptually different in that it proposes a realistic paradigm under which to examine ICL performance where there are no annotated samples. Subsequently, we differ from them in that we study three different token level tasks, different pool selection methods, and particularly focus on the effect of the pool size on downstream performance.

5 Conclusion

We proposed the framework of ICL on a budget and studied different methods for pool selection, focusing on token classification tasks. We hope this work will inspire more realistic assumptions on the amount of labeled data used in different ICL settings. In future work we will extend this analysis to additional tasks with high annotation costs.

6 Limitations

We tested our ICL on a budget approach on a single class of tasks (token classification), because it has high annotation cost and it was relatively less studied in the context of ICL. It is possible that other tasks will show different trends, hence we stress that the contribution here is the methodological approach, rather than advocating for one particular sampling strategy.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *ArXiv preprint*, abs/2303.08774.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared 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 M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Ernie Chang, Xiaoyu Shen, Hui-Syuan Yeh, and Vera Demberg. 2021. On training instance selection for few-shot neural text generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 8–13, Online. Association for Computational Linguistics.

Yukun Chen, Thomas A Lasko, Qiaozhu Mei, Joshua C Denny, and Hua Xu. 2015. A study of active learning methods for named entity recognition in clinical text. *Journal of biomedical informatics*, 58:11–18.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60, New York City, USA. Association for Computational Linguistics.

Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, et al. 2023. Phi-2: The surprising power of small language models. *Microsoft Research Blog*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *ArXiv preprint*, abs/2310.06825.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *ArXiv preprint*, abs/2401.04088.

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

Mingyi Liu, Zhiying Tu, Zhongjie Wang, and Xiaofei Xu. 2020. Ltp: a new active learning strategy for bert-crf based named entity recognition. *ArXiv preprint*, abs/2001.02524.

Stephen Mayhew, Terra Blevins, Shuheng Liu, Marek Šuppa, Hila Gonen, Joseph Marvin Imperial, Börje F Karlsson, Peiqin Lin, Nikola Ljubešić, LJ Miranda, et al. 2023. Universal ner: A gold-standard multilingual named entity recognition benchmark. *ArXiv preprint*, abs/2311.09122.

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.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666, Portorož, Slovenia. European Language Resources Association (ELRA).

Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *ArXiv preprint*, abs/2403.05530.

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.

Isabel Segura-Bedmar, Paloma Martínez, and María Herrero-Zazo. 2013. SemEval-2013 task 9 : Extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013). In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 341–350, Atlanta, Georgia, USA. Association for Computational Linguistics.

Yanyao Shen, Hyokun Yun, Zachary Lipton, Yakov Kronrod, and Animashree Anandkumar. 2017. Deep active learning for named entity recognition. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 252–256, Vancouver, Canada. Association for Computational Linguistics.

Yanyao Shen, Hyokun Yun, Zachary C. Lipton, Yakov Kronrod, and Animashree Anandkumar. 2018. Deep active learning for named entity recognition. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. 2022. Selective annotation makes language models better few-shot learners. *ArXiv preprint*, abs/2209.01975.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *ArXiv preprint*, abs/2307.09288.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9134–9148, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yue Zhang, Zhenghua Li, Jun Lang, Qingrong Xia, and Min Zhang. 2017. Dependency parsing with partial annotations: An empirical comparison. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 49–58, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

Banghua Zhu, Evan Frick, Tianhao Wu, Hanlin Zhu, and Jiantao Jiao. 2023. Starling-7b: Improving llm helpfulness & harmlessness with rlaif.

A Prompt Template

In this section we describe the prompt templates we use in our experiments.

A.1 NER task description

You are an NER classifier that identifies the following entities: Person __PER__, Organization __ORG__, Geo-Political __GPE__, Location __LOC__, Facility __FAC__, Work-of-Art __WOA__, Event __EVE__, Product __DUC__, Language __ANG__ use angle brackets to tag in-line, please don't include any additional information other than the annotated sentence and keep original spacing.

A.2 Dependency parsing and POS tagging task description

Your task is to parse the input sentence into a dependency tree by providing a (token, part-of-speech tag, head, deprel) for each token. In the input sentence, tokens are separated by spaces. Possible part-of-speech tags are:

VBN, WDT, GW, NN, TO, IN, JJR, WP, EX, VB, HYPH, JJ, SYM, :, RBR, MD, VBP, JJS, LS, WP$, $, VBD, VBZ, NFP, PRP, NNPS, CC, XX, „ “, NNP, -RRB-, CD, VBG, -LRB-, RP, NNS, PDT, AFX, RB, PRP$, UH, ., WRB, DT, FW, RBS, ADD, POS, ”

Output the parse and nothing else. Here are some examples:

A.3 Prompt design

We first describe the task in question, as outlined above. Next, we add the demonstration examples. For GPT and Claude we add the demonstration examples as follows. For each demonstration example, we use LangChain's[3] HumanMessage class

[3] langchain.com

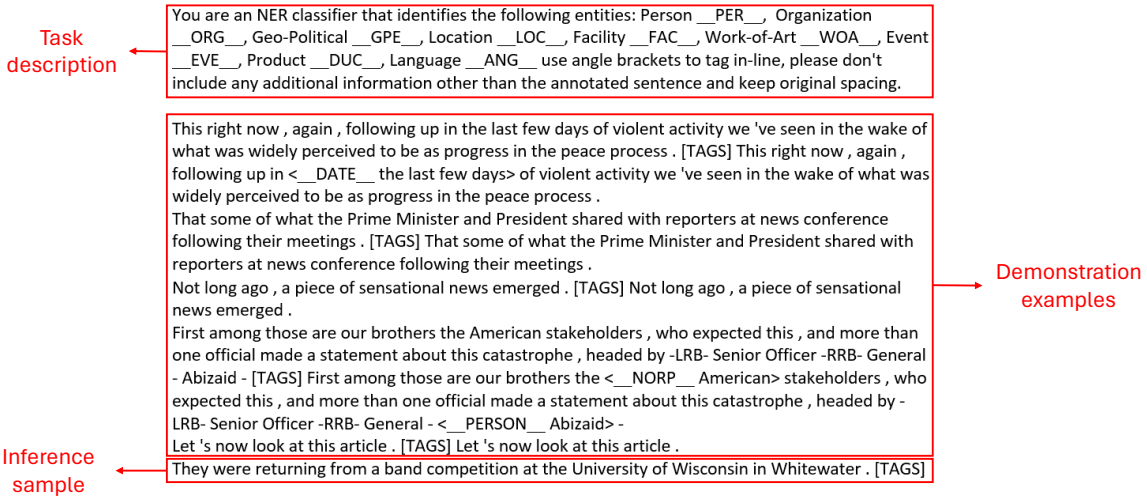

Figure 3: An example NER prompt used in our study.

	GPT	Claude	Gemini
NER	0.54	0.36	0.46
DP	0.64	0.62	0.64
POS	0.53	0.63	0.80

Table 2: Pearson correlation of performance with label diversity in the sample pool, as measured by the entropy of entities. All results are significant.

for the original sentence, followed by an AIMessage for the annotated sentence. Finally, we add the inference sample as a HumanMessage.

In Gemini, for which these classes are not implemented, we choose a different strategy: for each demonstration example, we add the original sentence followed by the separator token ([TAGS] for NER, [PARSE] for dependency parsing and POS tagging), and then the annotated sentence. Finally, we add the inference sample, followed by the separator token. Figures 3 and 4 demonstrates an NER and dependency parsing prompts for Gemini, respectively.

B Performance-Diversity Correlation

In Section 3.2 we computed the Pearson correlation of performance with diversity, as measured by the entropy of counts of entities. Table 2 presents the correlations.

C Additional Results

In the Section we report additional results on NER using a recently introduced dataset, Universal NER (UNER; Mayhew et al., 2023). The input texts are sourced from the UD dataset. We use the same models, prompts and pool sizes as used for Ontonotes.

Figure 5 displays the results, reinforcing our conclusions from the main text.

Task description

Your task is to parse the input sentence into a dependency tree by providing a (token, part-of-speech tag, head, deprel) for each token. In the input sentence, tokens are separated by spaces. Possible part-of-speech tags are: VBN, WDT, GW, NN, TO, IN, JJR, WP, EX, VB, HYPH, JJ, SYM, :, RBR, MD, VBP, JJS, LS, WP$, $, VBD, VBZ, NFP, PRP, NNPS, CC, XX, ,, ``, NNP, -RRB-, CD, VBG, -LRB-, RP, NNS, PDT, AFX, RB, PRP$, UH, ., WRB, DT, FW, RBS, ADD, POS.
Output the parse and nothing else. Here are some examples:

Demonstration examples

[input] (An outside water park !)
[parse] ((, -LRB-, 5, punct), (An, DT, 5, det), (outside, JJ, 5, amod), (water, NN, 5, compound), (park, NN, 0, root), (!, ., 5, punct), (), -RRB-, 5, punct)

[input] Besides the obvious tourist spots which I already have planned for .
[parse] (Besides, IN, 5, case), (the, DT, 5, det), (obvious, JJ, 5, amod), (tourist, NN, 5, compound), (spots, NNS, 0, root), (which, WDT, 10, obl), (I, PRP, 10, nsubj), (already, RB, 10, advmod), (have, VBP, 10, aux), (planned, VBN, 5, acl:relcl), (for, IN, 6, case), (., ., 5, punct)

[input] Maybe we can go drinking too .
[parse] (Maybe, RB, 4, advmod), (we, PRP, 4, nsubj), (can, MD, 4, aux), (go, VB, 0, root), (drinking, VBG, 4, xcomp), (too, RB, 4, advmod), (., ., 4, punct)

[input] To relax and to rejuvenate Disney cruise are the best option .
[parse] (To, TO, 2, mark), (relax, VB, 11, advcl), (and, CC, 5, cc), (to, TO, 5, mark), (rejuvenate, VB, 2, conj), (Disney, NNP, 7, compound), (cruise, NN, 11, nsubj), (are, VBP, 11, cop), (the, DT, 11, det), (best, JJS, 11, amod), (option, NN, 0, root), (., ., 11, punct)

[input] Explore which cruise may fit your needs the best .
[parse] (Explore, VB, 0, root), (which, WDT, 3, det), (cruise, NN, 5, nsubj), (may, MD, 5, aux), (fit, VB, 1, ccomp), (your, PRP$, 7, nmod:poss), (needs, NNS, 5, obj), (the, DT, 9, det), (best, RBS, 5, obl:npmod), (., ., 1, punct)

Inference sample

Parse the following sentence:
[input] Why not put together a bottle of champagne , a picnic and have a date on Treasure Island .
[parse]

Figure 4: An example dependency parsing prompt used in our study.

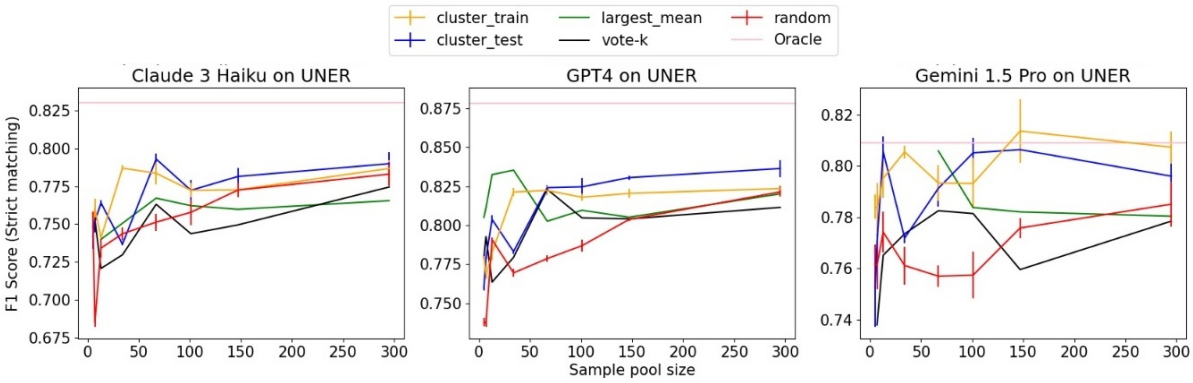

Figure 5: Results for NER using Claude 3 Haiku (left), GPT4 (middle) and Gemini 1.5 Pro (right) on UNER, as a function of the size of sample pool. In methods with a random component we run 3 trials and plot error bars showing the standard deviation. Oracle: using the full training set as the pool.