Efficient Active Learning with Adapters

Anonymous ACL submission

Abstract

One of the main obstacles for deploying Active Learning (AL) in practical NLP tasks is high computational cost of modern deep learning models. This issue can be partially mitigated by applying lightweight models as an acquisition model, but it can lead to the acquisition-successor mismatch (ASM) problem. Previous works show that the ASM problem can be partially alleviated by using distilled versions of a successor models as acquisition ones. However, distilled versions of pretrained models are not always available. Also, the exact pipeline of model distillation that does not lead to the ASM problem is not clear. To address these issues, we propose to use adapters as an alternative to full fine-tuning for acquisition model training. Since adapters are lightweight, this approach reduces the training cost of the model. We provide empirical evidence that it does not cause the ASM problem and can help to deploy active learning in practical NLP tasks.

1 Introduction

Recent progress in the natural language processing (NLP) tasks has become possible due to an abundant range of pre-trained language models. Data annotation is a rather important process, since the performance of model depends greatly on the quality of data it was trained on. Active learning (AL), which is a technique used to annotate data and train models efficiently, has been first introduced in (Cohn et al., 1996). This technique has been widely used to train language models to solve such NLP tasks as text classification (Dor et al., 2020), named entity recognition (Chen et al., 2015) and sequence labeling tasks (Settles and Craven, 2008a).

Active learning helps to reduce annotation costs by employing a specifically designed query strategy which works on sampling the data points that would bring the most substantial information gains for model training. One problem that has been described by (Tsvigun et al., 2022) is acquisition-successor mismatch (ASM). This refers to employing models of different architectures for acquisition (evaluating which samples would be the most beneficial) and successor (retraining with newly acquired samples) negatively impacts the performance. For some popular models, such as BERT, distilled versions can be used as acquisitions to save time and computational resources. We suggest using parameter-efficient fine-tuning methods for those models that do not have a distilled version. The findings of this study indicate that utilizing an adapter model with a successor of identical architecture consistently yields superior outcomes compared to a distilled model with a different architecture.

Our main contributions are the following:

- We show that training an acquisition model with adapters can speed up an AL loop (in comparison with using the full model for acquisition) and does not harm overall performance of AL;
- Our method can be efficiently applied to perform AL in various textual domains of the data;
- We experimentally show that our approach can be used with various types of pretrained encoder models that can be tuned with adapter networks;
- Total time of AL loop can be decreased by 20.84% on average.

2 Related work

In (Shelmanov et al., 2021) it was proposed to accelerate training and data selection steps for AL by leveraging distilled versions of the successor model during AL iterations. A similar approach was introduced in (Nguyen et al., 2022), where it was

proposed to use on-the-fly knowledge distillation of the successor model to the acquisition model. However, model distillation is expensive in terms of both time and computational resources. Furthermore, this approach cannot always be directly used in practice due to the lack of distilled models for several architectures.

In (Tsvigun et al., 2022), it was proposed to use pseudo labeling-based approach to mitigate the ASM problem. However, this approach can also suffer from the lack of distilled/teacher model pairs, especially for some specific domains.

Furthermore, (Jukić and Šnajder, 2023) explores the application of adapters in active learning in low-resource settings. The research concludes that some adapter configurations provide performance gains over full fine-tuning. The authors also investigate learning stability and compare layerwise representations obtained from adapters and fully fine-tuned models. They find that adapter models are more similar to the base model in earlier layers which are considered to contain foundational knowledge. In our work, we provide another kind of analysis: we compare uncertainty scores of different kinds of models and conclude that adapters can be applied in many areas of active learning, since the they do not affect the uncertainty scores. The research (Jukić and Šnajder, 2023) also provides comparison of performance scores of full and adapter models. However, there is no mention of time taken to train the models. We close this gap by measuring the speed of full and adapter models. Finally, in (Nguyen et al., 2022), adapters were used to improve time efficiency of the successor model, but their impact on the acquisition model was not analysed. We explore how adapters affect the time of the whole AL loop.

In our research, we bring empirical contribution by testing and analyzing the adapter application in active learning. We provide analysis of the uncertainty scores of adapter models to demonstrate the potential applicability of adapters in any domain of AL. Time efficiency of adapters in the AL loop is explored as well. It is also shown that adapters can help solve the ASM problem in active learning.

2.1 Adapters

Adapter modules were first introduced in (Houlsby et al., 2019). These modules are a small set of new layers introduced to the pre-trained model to be further updated without affecting the weights of the original model. Adapters offer a faster, more lightweight alternative to full fine-tuning, while maintaining the performance level of the latter.

As adapter training has proved to be a good PEFT method, a convenient open-source framework for adapters has been introduced in (Pfeiffer et al., 2020). The Adapters library (Poth et al., 2023)[1] offers a seamless way of adding, training and sharing a wide range of adapter modules for transformer models. This framework is used in this research to train and evaluate models with adapters.

3 Experiments

3.1 Experimental setup

The methodology we employ to set up our active learning experiments is consistent with the schema widely utilized in numerous prior studies (Settles and Craven, 2008b; Shen et al., 2017; Siddhant and Lipton, 2018; Shelmanov et al., 2021). This approach involves a simulated cycle of active learning, which consists of several distinct phases:

1. A small random sample (1% in our case) is taken from the dataset to initialize the training and annotation cycle.

2. An initital version of the acquisition model is constructed using the random data sample.

3. Each iteration of the cycle is continued by sampling a fraction of the data from the unlabeled pool (also 1%) by a query strategy and adding it to the training dataset that is used on the subsequent iterations.

4. On each iteration, the successor model is trained on the acquired data and evaluated on the whole test set.

5. Several iterations (12 in our case) are run in this way and a performance chart is built. Accuracy is used as a performance metric for the classification task investigated in this research.

6. Each reported experiment is run on five fixed random seeds to report standard deviation of the scores.

We use four query strategies to evaluate unlabeled samples in the active learning loop. For the classification task we use least confidence (LC) and breaking ties (BT). For NER, the strategy is

[1] https://github.com/adapter-hub/adapters

Maximum Normalized Log-Probability (MNLP). We also use random sampling to verify all other strategies against it. The strategies are described in detail in the section A.1.

Our approach is evaluated on three popular classification datasets that belong to different domains: English AG News topic classification dataset (Zhang et al., 2015), Banking77, a single-domain intent classification dataset (Casanueva et al., 2020) and the English language part of the Amazon MASSIVE dataset (FitzGerald et al., 2022), which contains utterances that belong to 18 different domains. We also evaluate on the CoNLL-2003 dataset, which is used for the NER task (Tjong Kim Sang and De Meulder, 2003). The datasets statistics can be found in the the section A.3.

3.2 Uncertainty scores evaluation

In order to verify that the adapters do not tamper with the output probability distributions when attached to a model and trained, we perform an analysis of the distributions of full models and adapter models. Since the query strategies used in this research rely on uncertainty scores, we evaluate the scores obtained from the models with adapters and compare them to the those from the full models. To perform the analysis, we utilize the uncertainty estimation framework presented in (Vazhentsev et al., 2022). We have applied the following statistical methods for scores evaluation: **Wasserstein distance (WD)** (Rubner et al., 1998) and **Kullback–Leibler (KL) divergence** (Kullback and Leibler, 1951).

The uncertainty score we evaluate is Bayesian Active Learning by Disagreement (BALD) (Houlsby et al., 2011). This metric of uncertainty assigns scores to data points according to the extent to which their labels would enhance our understanding of the actual distribution of model parameters.

All values of WD and KL divergence between the scores of full models and the scores of adapter models are presented in the Table 4.

3.3 Models

We conduct the experiments with pre-trained Transformers. In particular, ELECTRA-base (110M parameters) (Clark et al., 2020) and DistilBERT (66M parameters) (Sanh et al., 2019) models are fine-tuned on the three classification datasets. We have picked ELECTRA for the closest inspection. This decision is motivated by an assumption that we make: we theorize that in active learning, adapters perform more efficiently than than a small under-parameterized (distilled) models. For example, a combination of DistilBERT as an acquisition model and full ELECTRA as a successor perform worse than adapter ELECTRA model as acquisition and full ELECTRA as successor. Thus, we propose to use adapters to solve the ASM problem by matching the model architectures in acquisition and succession stages while reducing memory and time consumption. See the Results section for details.

3.4 Adapters for acquisition model

The acquisition model is equipped with a bottleneck adapter which consists of feed-forward layers after the multi-head attention block of each layer. The parameters are kept default as they are defined in the `BnConfig` base class of the Adapters library. The performance of this acquisition model with an adapter is then compared to the same kind of model but with no adapter attached.

4 Analysis

Figures 1, 2, 3 and 4 represent accuracy curves of four combinations of models and strategies. Each curve represents metrics averaged out over five seeds.

For both classification (Figures 3, 2, 1) and NER (Figure 4) tasks, it is obvious that ELECTRA with an adapter performs better than DistilBERT, which means that the acquisition-successor mismatch problem can be solved with adapters. In addition to it, adapters save training time while preserving the performance scores at the same level as full models.

In order to measure the speedup that adapter modules can provide in the active learning loop, we train full ELECTRA and adapter ELECTRA on the four datasets. We measure the time it takes to train on 2, 6 and 12% of the data and report it in the Table 1. As it is seen from the Table 1, adapter modules benefit from shorter training times in all cases. The average speedup adapters provide is 20.84%.

5 Results

As it is observed from Table 4, both distance metrics that have been measured between adapter and full ELECTRA models are close to zero, which means that the distributions of uncertainty scores

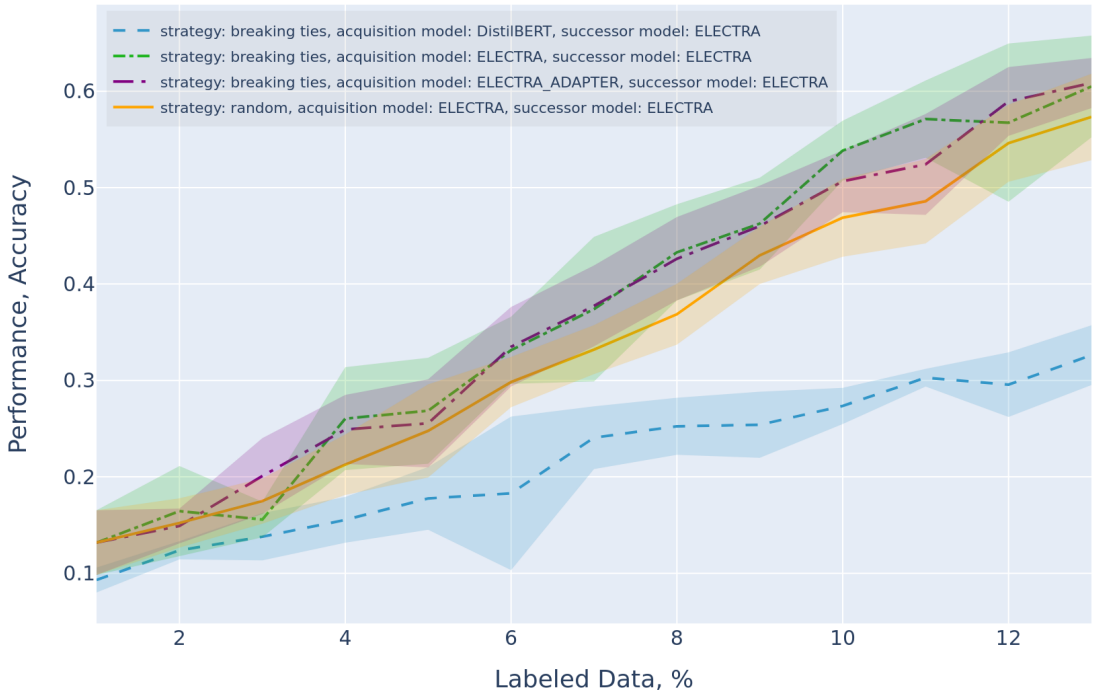

Figure 1: Text classification on Amazon Massive.

Dataset	**2%**	**6%**	**12%**
AG NEWS - full	367	1098	2201
AG NEWS - adapter	285	860	1730
BANKING 77 - full	30	89	178
BANKING 77 - adapter	23	71	140
MASSIVE (EN) - full	34	103	207
MASSIVE (EN) - adapter	27	82	164
CONLL - full	11	32	67
CONLL - adapter	9	25	55

Table 1: Time in seconds taken to train a full ELECTRA model and an ELECTRA model with a bottleneck adapter on four different datasets with 2, 6 and 12% of the data.

of those models are quite close to each other. Since active learning strategies rely on uncertainty scores, it means that in the active learning settings, training a model with an adapter speeds up the training time and consumes less memory without influencing the model's predictions compared to the full model fine-tuning.

Our experiments on four datasets show that models with the bottleneck adapter demonstrate a comparable performance on each active learning iteration with full models. We have also included experiments with DistilBERT as an acquisition model and this setup performs worse in comparison with all other setups due to the ASM problem discussed earlier. In addition, we have concluded that the adapter helps speed up the active learning process when added to the acquisition model. All this makes the adapter models more efficient for classification in active learning.

6 Conclusion

The finding of this study include the following:

1. Statistical tests of uncertainty scores (BALD, in particular) obtained from full models and adapter models have concluded that the predictions of the two types of models are similar enough to use the adapter models in active learning with no significant perturbation of predictions.

2. Adapter models require shorter training time, which may be utilized to accelerate the cycles of active learning.

3. In active learning settings, adapter models can be used to overcome the ASM problem caused by different architectures of acquisition and successor models.

7 Limitations

Although we have demonstrated that adapters can be useful in the active learning settings, our experiments only include the task of text classification and named entity recognition on four particular open source datasets. For further research, adapters may be tested on different tasks and datasets.

In addition, this research is only focused on one particular model and investigates the behavior of ELECTRA in the active learning settings. It would be interesting to apply the same approach to models of different architectures as well.

References

Iñigo Casanueva, Tadas Temcinas, Daniela Gerz, Matthew Henderson, and Ivan Vulic. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on NLP for ConvAI - ACL 2020*. Data available at https://github.com/PolyAI-LDN/task-specific-datasets.

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.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

David A Cohn, Zoubin Ghahramani, and Michael I Jordan. 1996. Active learning with statistical models. *Journal of artificial intelligence research*, 4:129–145.

Liat Ein Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2020. Active learning for bert: an empirical study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7949–7962.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, and Prem Natarajan. 2022. Massive: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp.

Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. 2011. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*.

Josip Jukić and Jan Šnajder. 2023. Parameter-efficient language model tuning with active learning in low-resource settings. *arXiv preprint arXiv:2305.14576*.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

D Gale Lewis and William A Gale. 1994. W. a sequential algorithm for training text classifiers. In *Proceedings of SIGIR-94*, pages 3–12.

Tong Luo, Kurt Kramer, Dmitry B Goldgof, Lawrence O Hall, Scott Samson, Andrew Remsen, Thomas Hopkins, and David Cohn. 2005. Active learning to recognize multiple types of plankton. *Journal of Machine Learning Research*, 6(4).

Minh Van Nguyen, Nghia Trung Ngo, Bonan Min, and Thien Huu Nguyen. 2022. Famie: A fast active learning framework for multilingual information extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Demonstrations*.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. Adapterhub: A framework for adapting transformers. *arXiv preprint arXiv:2007.07779*.

Clifton Poth, Hannah Sterz, Indraneil Paul, Sukannya Purkayastha, Leon Engländer, Timo Imhof, Ivan Vulić, Sebastian Ruder, Iryna Gurevych, and Jonas Pfeiffer. 2023. Adapters: A unified library for parameter-efficient and modular transfer learning. *arXiv preprint arXiv:2311.11077*.

Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. 1998. A metric for distributions with applications to image databases. In *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*, pages 59–66. IEEE.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Burr Settles and Mark Craven. 2008a. An analysis of active learning strategies for sequence labeling tasks. In *proceedings of the 2008 conference on empirical methods in natural language processing*, pages 1070–1079.

Burr Settles and Mark Craven. 2008b. An analysis of active learning strategies for sequence labeling tasks. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 1070–1079, Honolulu, Hawaii. Association for Computational Linguistics.

Artem Shelmanov, Dmitri Puzyrev, Lyubov Kupriyanova, Denis Belyakov, Daniil Larionov, Nikita Khromov, Olga Kozlova, Ekaterina Artemova, Dmitry V. Dylov, and Alexander Panchenko. 2021. Active learning for sequence tagging with deep pre-trained models and Bayesian uncertainty estimates. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1698–1712, Online. 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.

Aditya Siddhant and Zachary C. Lipton. 2018. Deep Bayesian active learning for natural language processing: Results of a large-scale empirical study. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2904–2909, Brussels, Belgium. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Akim Tsvigun, Artem Shelmanov, Gleb Kuzmin, Leonid Sanochkin, Daniil Larionov, Gleb Gusev, Manvel Avetisian, and Leonid Zhukov. 2022. Towards computationally feasible deep active learning. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1198–1218, Seattle, United States. Association for Computational Linguistics.

Artem Vazhentsev, Gleb Kuzmin, Artem Shelmanov, Akim Tsvigun, Evgenii Tsymbalov, Kirill Fedyanin, Maxim Panov, Alexander Panchenko, Gleb Gusev, Mikhail Burtsev, et al. 2022. Uncertainty estimation of transformer predictions for misclassification detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8237–8252.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *CoRR*, abs/1509.01626.

A Appendix

A.1 Query strategies

In the experiments, the following query strategies are used to evaluate the queries from the pool of the unlabeled data and add them to the labeled pool:

Random sampling is used as a baseline for all experiments. It simply picks data from a dataset randomly from a uniform distribution.

Least Confidence (LC) strategy is applied in most of the experiments. LC is a popular measure of uncertainty which is defined as follows:

$$\text{LC} = 1 - \max_{y} \left(\mathbb{P}(y|x)\right)$$

where x is an instance of the unlabeled data and y is a class that was predicted for this data instance. (Lewis and Gale, 1994)

Breaking Ties (BT) strategy inspects two maximal probabilities and picks instances with the minimum margin between them. (Luo et al., 2005)

$$\text{BT} = \min_{y}(\mathbb{P}(y_1|x) - \mathbb{P}(y_2))$$

where y_1 and y_2 are the first and second most likely labels respectively.

Maximum Normalized Log-Probability (MNLP) is a strategy proposed specifically for the NER task (Tjong Kim Sang and De Meulder, 2003). It is based on sampling the instances with the lowest log probability, which has been normalized by sequence length.

A.2 Statistical methods for comparing UE scores

- **Wasserstein distance (WD)**, also known as the earth mover distance (Rubner et al., 1998), shows how much "work" needs to be applied to transform one probability distribution into another. It can be assumed that a low numerical value of WD means that two distrubutions are similar.
- **Kullback–Leibler (KL) divergence** (Kullback and Leibler, 1951) is a general measure of how different one probability distribution is in reference to another. A low value of KL divergence means the two distributions are identical in the context of the information they convey.

A.3 Datasets

We evaluate our approach on the classificaton task. We utilize three popular datasets: English AG News topic classification dataset (Zhang et al., 2015), Banking77, a single-domain intent classification dataset (Casanueva et al., 2020) and the English language part of the Amazon MASSIVE dataset (FitzGerald et al., 2022). We also evaluate our approach on NER task. For this task, we utilize the CoNLL-2003 dataset (Tjong Kim Sang and De Meulder, 2003).

The statistics on the datasets are presented in the Table 2.

Dataset	Train	Test	C
AG NEWS	120K	7.6K	4
BANKING 77	10K	3K	77
MASSIVE (EN)	11.5K	2.9K	60
CONLL-2003	14K	3.4	9

Table 2: Datasets statistics on the number of samples in the train, validation and test sets. C stands for the number of classes or labels.

Amazon Massive dataset and Banking 77 dataset are distributed under Creative Commons Attribution 4.0 International Public License.

All models used in this research are distributed under Apache License 2.0.

A.4 Datasets and active learning strategies

As it can be observed from the Table 2, the AG News dataset contains much more samples and much less classes than any other dataset explored in this research. So the accuracy gains in the experiments on AG News can be explained by the fact that this information is quite easy to learn.

While experimenting with active learning on these datasets, we pointed out several observations. In some cases, randomly picking data samples demonstrates very similar performance metrics to those setups that use a query strategy but never outperforms them, as it is seen in Figure 2 for the Banking77 dataset. However, in the cases of a more balanced and structured data with less classes random sampling performs much worse (for example, AG News in Figure 3).

Two query strategies (LC and BT) have been analyzed for different classification datasets and it has been found that BT, which is based on selecting the samples with almost identical predictions for most probable classes, demonstrates a better performance on the AG News dataset than LC. At the same time, LC strategy, which simply queries the samples that the classifier is the least certain about, is more effective on Banking77 and Amazon Massive. We conclude that exploring a variety of strategies is important particularly when faced with a singular task accompanied by multiple datasets of diverse structures.

A.5 Computing infrastructure

Experiments were conducted using one NVIDIA GeForce RTX 3090 GPU with 24 GB of memory, hosted on a server with 2 Intel Xeon Silver 4216 CPUs at 2.10GHz with 60GB of RAM running Ubuntu 22.04.2 LTS. Our models were implemented using PyTorch 2.1.2. We ensured reproducibility by setting five random seeds for all experiments. Hyperparameter tuning was not performed, a fixed set of hyperparameters was used instead, which is listed in the Table 3. The average training time for each seed of our models was approximately 1.5 hours.

Hyperparameter	**Value**
Learning Rate	2e-5
Batch Size	16
Epochs	15
Dropout Rate	0

Table 3: Hyperparameter setup for all models used in the experiments. For adapter models the value of the learning rate is 1e-4.

A.6 Tables

Dataset	**WD**	**KL divergence**
AG NEWS	0.0004	0.0469
BANKING 77	0.0007	0.0071
MASSIVE (EN)	0.0006	0.01
CONLL	0.0007	0.0034

Table 4: Distance metrics computed over BALD scores obtained from full ELECTRA model and adapter ELECTRA model. The two configurations of the models have been fine-tuned on four different datasets.

A.7 Figures

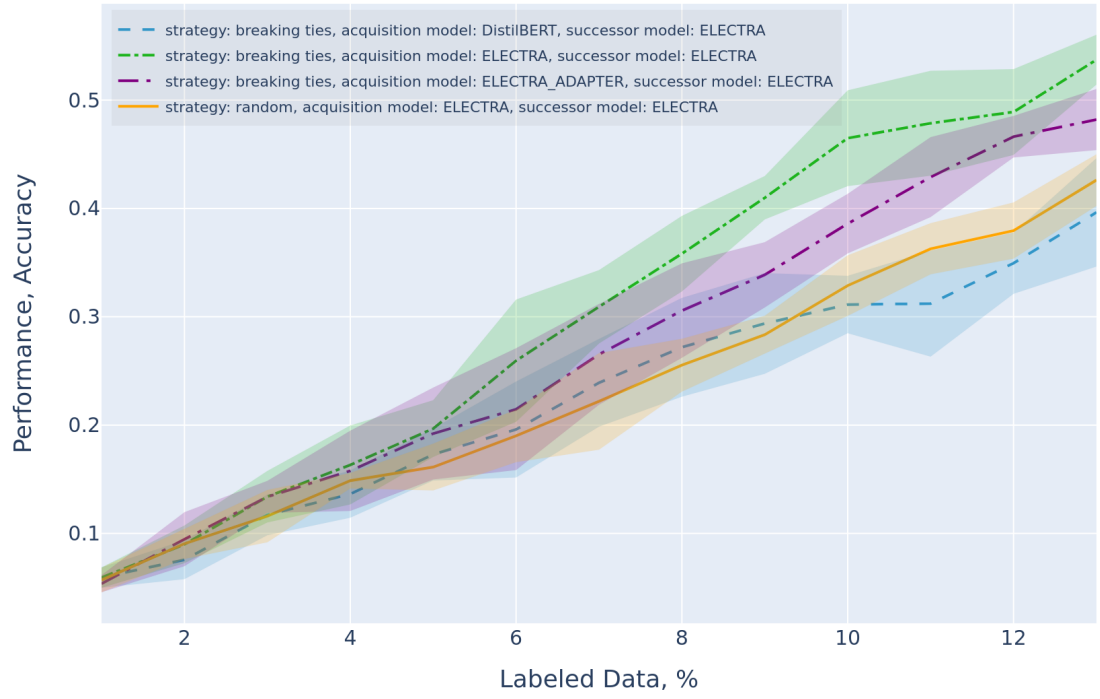

Figure 2: Text classification on Banking77.

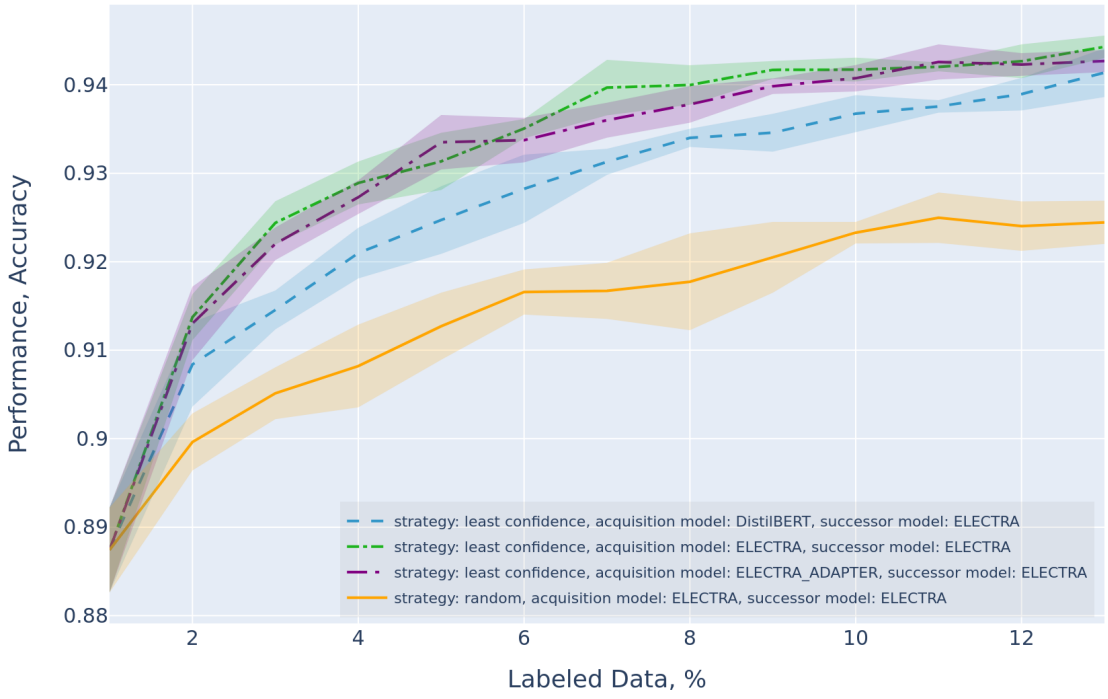

Figure 3: Text classification on AG News.

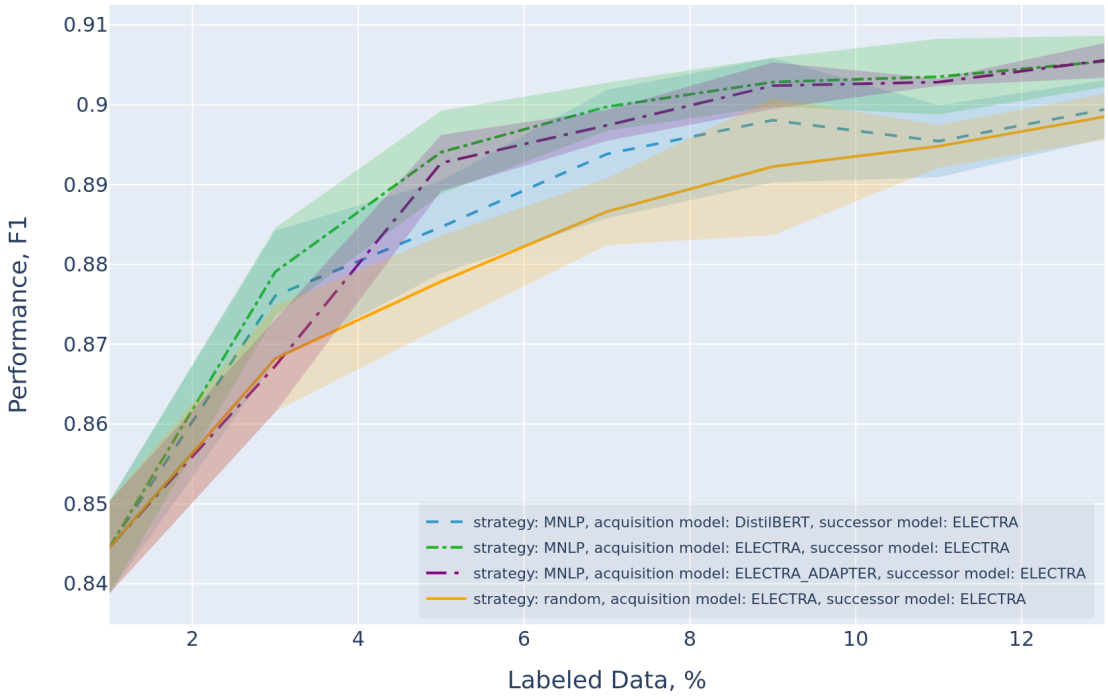

Figure 4: Named entity recognition on CoNLL 2003.