Revisiting Uncertainty-based Query Strategies for Active Learning with Transformers

Christopher Schröder, **Andreas Niekler**, and **Martin Potthast**

Leipzig University

Abstract

Active learning is the iterative construction of a classification model through targeted labeling, enabling significant labeling cost savings. As most research on active learning has been carried out before transformer-based language models ("transformers") became popular, despite its practical importance, comparably few papers have investigated how transformers can be combined with active learning to date. This can be attributed to the fact that using state-of-the-art query strategies for transformers induces a prohibitive runtime overhead, which effectively nullifies, or even outweighs the desired cost savings. For this reason, we revisit uncertainty-based query strategies, which had been largely outperformed before, but are particularly suited in the context of fine-tuning transformers. In an extensive evaluation, we connect transformers to experiments from previous research, assessing their performance on five widely used text classification benchmarks. For active learning with transformers, several other uncertainty-based approaches outperform the well-known prediction entropy query strategy, thereby challenging its status as most popular uncertainty baseline in active learning for text classification.

1 Introduction

Collecting labeled data for machine learning can be costly and time-consuming. A key technique to minimize labeling costs has been active learning, where an oracle (e.g., a human expert) is queried to label problem instances selected that are deemed to be most informative to the learning algorithm's next iteration according to a query strategy.

Active learning is characterized by the real-world machine learning scenario in which large amounts of training data are unavailable, which may explain why comparably little research has investigated deep learning in this context. The recent widely successful transformer-based language models can circumvent the limitations imposed by small training datasets (Vaswani et al., 2017; Devlin et al., 2019). Pre-trained on large amounts of unlabeled text, they can be fine-tuned to a given task using far less training data than when trained from scratch. However, their high number of model parameters renders them computationally highly expensive, for query strategies that are targeted at neural networks or text classification (Settles et al., 2007; Zhang et al., 2017), resulting in prohibitive turnaround times between labeling steps.

In this paper, we systematically investigate uncertainty-based query strategies as a computationally inexpensive alternative. Despite their relative disadvantages in traditional active learning, when paired with transformers, they are highly effective as well as efficient. Our extensive experiments assess a multitude of combinations including state-of-the-art transformer models BERT (Devlin et al., 2019) and DistilRoBERTa (Sanh et al., 2019), five well-known sentence classification benchmarks, and five query strategies.[1]

2 Related Work

Uncertainty-based query strategies used to be the most common choice in active learning, using uncertainty scores obtained from the learning algorithm (Lewis and Gale, 1994), estimates obtained via ensembles (Krogh and Vedelsby, 1994; RayChaudhuri and Hamey, 1995), or prediction entropy (Perona et al., 2008). More recently—predating transformers—neural network-based active learning predominantly employed query strategies that select problem instances according to (1) the magnitude of their backpropagation-induced gradients (Settles et al., 2007; Zhang et al., 2017), where instances causing a high-magnitude gradient inform the model better, and (2) representativity-based criteria (e.g., coresets (Sener and Savarese, 2018)), which select instances from a vector space to geometrically represent the full dataset.

[1]Code: https://github.com/webis-de/ACL-22

Findings of the Association for Computational Linguistics: ACL 2022, pages 2194 - 2203

For today's deep neural networks, ensembles are too computationally expensive, and prediction entropy has been observed to be overconfident (Guo et al., 2017; Lakshminarayanan et al., 2017). The exception are flat architectures, where, among others, Prabhu et al. (2019) showed fastText (Joulin et al., 2017) to be effective, well-calibrated, and computationally efficient. Prior to transformers, query strategies relying on expected gradient length (Settles et al., 2007) achieved the best results on many active learning benchmarks for text classification (Zhang et al., 2017). Gradients depend on the current model, which means, when used for a query strategy, they scale with the vast number of a transformer's parameters, and moreover, they need to be computed per-instance instead of batch-wise, thereby becoming computationally expensive.

The cost of ensembles, the adverse scaling of network parameters in gradient-based strategies, and a history of deeming neural networks to be overconfident effectively rule out the most predominantly used query strategies. This might explain why transformers, despite the success of fine-tuning them for text classification (Howard and Ruder, 2018; Yang et al., 2019; Sun et al., 2019), have only very recently been considered at all in combination with active learning (Lu and MacNamee, 2020; Yuan et al., 2020; Ein-Dor et al., 2020; Margatina et al., 2021). All of the related works mitigate the computationally complex query strategies by subsampling the unlabeled data before querying (Lu and MacNamee, 2020; Ein-Dor et al., 2020; Margatina et al., 2021), by performing fewer queries with larger sample sizes (Yuan et al., 2020; Margatina et al., 2021), or by tailoring to less expensive settings, namely binary classification (Ein-Dor et al., 2020). Subsampling, however, introduces additional randomness which can aggravate comparability across experiments, and large sample sizes increase the amount of labeled data, which is contrary to minimizing the labeling effort.

Due to this computationally challenging setting, the uncertainty-based prediction entropy query strategy (Roy and McCallum, 2001; Schohn and Cohn, 2000) is therefore a frequently used baseline and a lowest common denominator in recent work on active learning for text classification (Zhang et al., 2017; Lowell et al., 2019; Prabhu et al., 2019; Ein-Dor et al., 2020; Lu and MacNamee, 2020; Yuan et al., 2020; Margatina et al., 2021; Zhang and Plank, 2021). Apart from being employed as baselines, uncertainty-based query strategies have not been systematically analyzed in conjunction with transformers, and moreover, comparisons to the previous benchmarks by Zhang et al. (2017) have been omitted by the aforementioned related work. Our work not only closes this gap, but also reevaluates the relative strength of uncertainty-based approaches, including two recently largely neglected strategies, thereby challenging the status of prediction entropy as the most popular baseline.

3 Transformer-based Active Learning

The goal of active learning is to minimize the labeling costs of training data acquisition while maximizing a model's performance (increase) with each newly labeled problem instance. In contrast to regular supervised text classification ("passive learning"), it operates iteratively, where in each iteration (1) a so-called query strategy selects new instances for labeling according to an estimation of their informativeness, (2) an oracle (e.g., a human expert) provides the respective label, and (3) a learning algorithm either uses the newly labeled instance for its next learning step, or a model is retrained from scratch using all previously labeled instances. This work considers pool-based active learning (Lewis and Gale, 1994), where the query strategies have access to all unlabeled data. Notation-wise, we denote instances by $x_1, x_2, \ldots, x_n$, the number of classes by c, the respective label for instance x_i by y_i (where $\forall i : y_i \in \{1, \ldots, c\}$), and $P(y_i|x_i)$ is a probability-like predicted class distribution.

Query Strategies We consider three well-known uncertainty-based query strategies, one recent state-of-the-art strategy that coincidentally also includes uncertainty, and a random baseline:

(1) Prediction Entropy (PE; Roy and McCallum, 2001; Schohn and Cohn, 2000) selects instances with the highest entropy in the predicted label distribution with the aim to reduce overall entropy:

$$\underset{x_i}{\operatorname{argmax}} \left[-\sum_{j=1}^{c} P(y_i = j|x_i) \log P(y_i = j|x_i) \right]$$

(2) Breaking Ties (BT; Scheffer et al., 2001; Luo et al., 2005) takes instances with the minimum margin between the top two most likely probabilities:

$$\underset{x_i}{\operatorname{argmin}} \left[P(y_i = k_1^*|x_i) - P(y_i = k_2^*|x_i) \right]$$

where k_1^* is the most likely label in the posterior class distribution $P(y_i|x_i)$, and k_2^* the second most

Dataset Name (ID)	Type	Classes	Training	Test
AG's News (AGN)	N	4	120,000	(*) 7,600
Customer Reviews (CR)	S	2	3,397	378
Movie Reviews (MR)	S	2	9,596	1,066
Subjectivity (SUBJ)	S	2	9,000	1,000
TREC-6 (TREC-6)	Q	6	5,500	(*) 500

Table 1: Key information about the examined datasets. The dataset type was abbreviated as follows: N: News, S: Sentiment, Q: Questions. (*): Predefined test sets were available and adopted.

likely label respectively. *In the binary case*, this margin is small iff the label entropy is high, which is why BT and PE then select the same instances.
(3) Least Confidence (LC; Culotta and McCallum, 2005) selects instances whose most likely label has the least confidence according to the current model:

$$\underset{x_i}{\operatorname{argmax}} \left[1 - P(y_i = k_1^*|x_i)\right]$$

(4) Contrastive Active Learning (CA; Margatina et al., 2021) selects instances with the maximum mean Kullback-Leibler (KL) divergence between the predicted class distributions ("probabilities") of an instance and each of its m nearest neighbors:

$$\underset{x_i}{\operatorname{argmax}} \left[\frac{1}{m}\sum_{j=1}^{m} \mathrm{KL}(P(y_j|x_j^{knn}) \parallel P(y_i|x_i))\right]$$

where the instances x_j^{knn} are the m nearest neighbors of instance x_i.
(5) Random Sampling (RS), a commonly used baseline, draws uniformly from the unlabeled pool.

Oracle The oracle is usually operationalized using the training datasets of existing benchmarks: To ensure comparability with the literature, we pick important standard text classification tasks.

Classification We fine-tune BERT (Devlin et al., 2019) and DistilRoBERTa (Sanh et al., 2019) on several natural language understanding datasets. BERT is well-researched as transformer and has recently also shown strong results in active learning (Yuan et al., 2020; Ein-Dor et al., 2020; Margatina et al., 2021). The model consists of 24 layers, hidden units of size 1024 and 336M parameters in total. DistilRoBERTa, by contrast, is a more parameter-efficient alternative which has merely six layers, hidden units of size 768, and 82M parameters. We also trained a passive model on the full data.

The classification model consists of the respective transformer, on top of which we add a fully connected projection layer, and a final softmax output layer. We use the "[CLS]" token that is computed by the transformer as sentence representation. Regarding fine-tuning, we adopt the combination of discriminative fine-tuning and slanted triangular learning rates (Howard and Ruder, 2018). The main active learning routine is then as follows: (1) The query strategy, either using the model from the previous iteration, or sampling randomly, selects 25 instances from the unlabeled pool. (2) The oracle provides labels for these instances. (3) The next model is trained using all data labeled so far.

Model	Strategy	Mean Rank		Mean Result	
		Acc.	AUC	Acc.	AUC
SVM	PE	1.80	2.60	0.764	0.663
	BT	**1.60**	**1.60**	**0.767**	**0.697**
	LC	3.00	2.60	0.751	0.672
	CA	5.00	5.00	0.667	0.593
	RS	3.00	2.60	0.757	0.686
KimCNN	PE	1.60	2.40	**0.818**	0.742
	BT	**1.60**	**2.00**	**0.818**	**0.750**
	LC	3.80	2.80	0.810	0.732
	CA	3.80	4.80	0.793	0.711
	RS	3.60	2.40	0.804	0.749
D.RoBERTa	PE	2.60	3.00	0.901	0.856
	BT	2.20	**1.80**	0.902	**0.864**
	LC	**1.40**	2.00	**0.904**	0.860
	CA	3.00	3.40	0.901	0.852
	RS	5.00	4.20	0.884	0.853
BERT	PE	2.40	2.40	0.909	0.859
	BT	**2.00**	**1.60**	0.914	**0.873**
	LC	2.20	3.80	**0.917**	0.866
	CA	2.80	2.60	0.916	0.872
	RS	5.00	4.00	0.899	0.861

Table 2: The "Mean Rank" columns show the mean rank when ordered by mean accuracy (Acc.) after the final iteration and by overall AUC. The "Mean Result" columns show the mean accuracy and AUC.

Baselines For comparison, we consider a linear SVM, and KimCNN (Kim, 2014), which have been used extensively in text classification, disregarding active learning. We adopted the KimCNN parameters from Kim (2014) and Zhang et al. (2017).

4 Evaluation

We evaluate five query strategies in combination with BERT, DistilRoBERTa and two baselines.

Datasets and Experimental Setup In Table 1, we show the five datasets employed, which have previously been used to evaluate active learning: AG's News (AGN; Zhang et al., 2015), Customer Reviews (CR; Hu and Liu, 2004), Movie Reviews

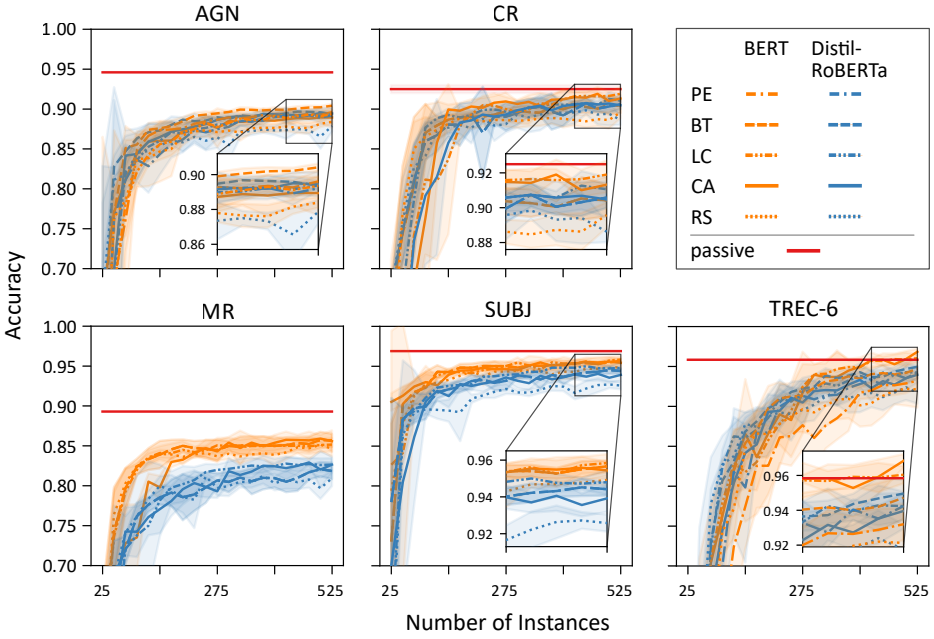

Figure 1: Active learning curves of BERT and DistilRoBERTa when combined with five query strategies: Prediction Entropy (PE), Breaking Ties (BT), Least Confidence (LC), Contrastive Active Learning (CA), and Random Sampling (RS). The tubes around the lines represent standard deviation over five runs. For comparison, the horizontal line depicts a passive text classification for which BERT has been trained using the entire training set.

(MR; Pang and Lee, 2005), Subjectivity (SUBJ; Pang and Lee, 2004), and TREC-6 (Li and Roth, 2002). These datasets encompass binary and multi-class classification in different domains, and they are class-balanced, except for TREC-6. Where available, we employed the pre-existing test sets, or otherwise a random sample of 10%.

We follow the experiment setup of Zhang et al. (2017): 25 training instances are used to train the first model, followed by 20 active learning iterations, during each of which 25 instances are queried and labeled. Using 10% of the so far labeled data as validation set, we stop early (Duong et al., 2018) when accuracy surpasses 98%, or the validation loss does not increase for five epochs.

Results For each combination of dataset, model, and query strategy, Figure 1 shows the respective learning curves. The horizontal line shows the best model's score when trained on the full dataset, which four out of five datasets approach very closely, or even exceed. As expected, BERT generally achieves steeper learning curves than DistilRoBERTa, but surprisingly, during later iterations DistilRoBERTa reaches scores only slightly worse than BERT for all datasets except MR. Regarding query strategies, RS is a strong contender during early iterations, e.g., as can be seen for the first few iterations of CR. This is partly because all but one of the datasets are balanced, but nevertheless, RS is eventually outperformed by the other strategies in most cases. For imbalanced datasets, Ein-Dor et al. (2020) have shown RS to be less effective, which we can confirm for TREC-6. While in terms of area under the learning curve (AUC)

Dataset	Model	Strategy	Acc.	Data Use
AGN	BERT	BT	0.904	0.4%
	BERT	passive (ours)	0.946	100.00%
	XLNet[1]	passive	0.955	100.00%
CR	BERT	LC	0.919	15.45%
	BERT	passive (ours)	0.925	100.00%
	HAC[2]	passive	0.889	100.00%
MR	BERT	PE, BT	0.857	0.547%
	BERT	passive (ours)	0.893	100.00%
	SimCSE[3]	passive	0.884	100.00%
SUBJ	BERT	LC	0.958	5.83%
	BERT	passive (ours)	0.969	100.00%
	AdaSent[4]	passive	0.955	100.00%
TREC-6	BERT	CA	0.968	9.55%
	BERT	passive (ours)	0.958	100.00%
	RCNN[5]	passive	0.962	100.00%

Table 3: Best final accuracy compared to (our) passive classification and state-of-the-art text classification: [1]Yang et al. (2019), [2]Zheng et al. (2019), [3]Gao et al. (2021), [4]Zhao et al. (2015), [5]Tay et al. (2018). "Data Use" indicates proportion of training data used.

there seems to be no overall best strategy, PE/BT and CA often show very steep learning curves.

In Table 2, we rank the query strategies by their average accuracy and AUC results, ranging from 1 (best) to 5 (worst). We also report their average accuracy and AUC per model and query strategy. Surprisingly, we can see that PE, a commonly used and proven to be strong baseline, which has been a lowest common denominator in recent work on active learning for text classification (Zhang et al., 2017; Lowell et al., 2019; Prabhu et al., 2019; Ein-Dor et al., 2020; Lu and MacNamee, 2020; Yuan et al., 2020; Margatina et al., 2021; Zhang and Plank, 2021), is on average outranked by BT when using transformers. BT achieves the best AUC ranks and scores, and in many cases also the best accuracy ranks and scores. It seems to be similarly effective on the baselines as well. Moreover, LC also outperforms PE for DistilRoBERTa where it even competes with BT. Detailed accuracy and AUC scores including standard deviations are reported in Appendix Tables 5 & 7.

Table 3 compares the best model trained via active learning per dataset against passive text classification, namely (1) our own model trained on the full training set, and (2) state-of-the-art results. The largest discrepancy between active learning and passive text classification is observed on AGN, which is also the largest dataset from which the active learning models use less than 1% for training. Otherwise, all models are close to or even surpass the state of the art, using only between 0.4% and 14% of the data. Noteworthy, LC achieves the best accuracy result for two datasets, while the strong baseline PE and the state-of-the-art approach CA perform best on only one dataset each.

In Table 4, we report the best AUC scores per dataset, and compare them to previous work. BT ranks highest in two out of three cases with CA achieving the best result on the remaining two datasets. BERT achieves the best AUC scores on all datasets with a considerable increase in AUC compared to Zhang et al. (2017).

In summary, we use recent transformer models in combination with several query strategies to evaluate a previously established but lately neglected benchmark. We find that the PE baseline is outperformed by BT, which, as a reminder, selects the same instances as PE for binary classification, but shows superior results on multi-class datasets. We conclude that BT, which even outperforms the state-of-the-art strategy CA in many cases, is therefore a strong contender to become the new default uncertainty-based baseline. Finally, DistilRoBERTa, using less than 25% of BERT's parameters, achieves results that are remarkably close to BERT at only a fraction of the overhead. Considering the computational burdens that motivated this work, this increase in efficiency is often preferable from a practitioner's perspective.

Dataset	Model	AUC
AGN	BERT (BT, ours)	0.875
	–	
CR	BERT (PE, BT; ours)	0.877
	CNN^6	0.743
MR	BERT (PE, BT; ours)	0.833
	CNN^6	0.707
SUBJ	BERT (CA, ours)	0.943
	CNN^6	0.856
TREC-6	BERT (CA, ours)	0.868
	–	

Table 4: Best area under curve (AUC) scores (averaged over five runs) compared to Zhang et al. (2017).

5 Conclusions

An investigation of the effectiveness of uncertainty-based query strategies in combination with BERT and DistilRoBERTa for active learning on several sentence classification datasets shows that uncertainty-based strategies still perform well. We evaluate five query strategies on an established benchmark, for which we achieve results close to state-of-the-art text classification on four out of five datasets, using only a small fraction of the training data. Contrary to current literature, prediction entropy, the supposedly strongest uncertainty-based baseline, is outperformed by several uncertainty-based strategies on this benchmark—in particularly by the breaking ties strategy. This invalidates the common practice of solely relying on prediction entropy as baseline, and shows that uncertainty-based strategies demand renewed attention especially in the context of transformer-based active learning.

Acknowledgments

We thank the anonymous reviewers for their valuable and constructive feedback. This research was partially funded by the Development Bank of Saxony (SAB) under project number 100335729.

Ethical Considerations

Research on active learning improves the labeling of data, by efficiently supporting the learning algorithm with targeted information, so that overall less data has to be labeled. This could contribute to creating machine learning models, which would otherwise be infeasible, either due to limited budget, or time. Active learning can be used for good or bad, and our contributions would—in both cases–show how to make this process more efficient.

Moreover, we use pre-trained models, which can contain one or more types of bias. Bias, however, affects all approaches based on fine-tuning pre-trained language models, but therefore this has to be kept in mind and mitigated all the more.

References

Aron Culotta and Andrew McCallum. 2005. Reducing labeling effort for structured prediction tasks. In *Proceedings of the 20th National Conference on Artificial Intelligence*, pages 746–751.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 4171–4186.

Long Duong, Hadi Afshar, Dominique Estival, Glen Pink, Philip Cohen, and Mark Johnson. 2018. Active learning for deep semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 43–48.

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.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6894–6910.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 1321–1330.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 328–339.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 427–431.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

Anders Krogh and Jesper Vedelsby. 1994. Neural network ensembles, cross validation and active learning. In *Proceedings of the 7th International Conference on Neural Information Processing Systems (NIPS)*, pages 231–238.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS)*, pages 6405–6416.

David D. Lewis and William A. Gale. 1994. A sequential algorithm for training text classifiers. In *Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval*, pages 3–12.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics (COLING)*, pages 1–7.

David Lowell, Zachary C. Lipton, and Byron C. Wallace. 2019. Practical obstacles to deploying active learning. 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 21–30.

Jinghui Lu and Brian MacNamee. 2020. Investigating the effectiveness of representations based on pre-trained transformer-based language models in active learning for labelling text datasets. *arXiv preprint arXiv:2004.13138*.

Tong Luo, Kurt Kramer, Dmitry B. Goldgof, Lawrence O. Hall, Scott Samson, Andrew Remsen, and Thomas Hopkins. 2005. Active learning to recognize multiple types of plankton. *Journal of Machine Learning Research (JMLR)*, 6:589–613.

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. 2021. Active learning by acquiring contrastive examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 650–663.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 271–278.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 115–124.

P. Perona, A. Holub, and M. C. Burl. 2008. Entropy-based active learning for object recognition. In *Proceedings of the 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops)*, pages 1–8.

Ameya Prabhu, Charles Dognin, and Maneesh Singh. 2019. Sampling bias in deep active classification: An empirical study. 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 4058–4068.

Tirthankar RayChaudhuri and Leonard G. C. Hamey. 1995. Minimisation of data collection by active learning. In *Proceedings of International Conference on Neural Networks (ICNN)*, pages 1338–1341.

Nicholas Roy and Andrew McCallum. 2001. Toward optimal active learning through sampling estimation of error reduction. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, pages 441–448.

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

Tobias Scheffer, Christian Decomain, and Stefan Wrobel. 2001. Active hidden markov models for information extraction. In *Proceedings of the 4th International Conference on Advances in Intelligent Data Analysis (IDA)*, pages 309–318.

Greg Schohn and David Cohn. 2000. Less is more: Active learning with support vector machines. In *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*, pages 839–846.

Ozan Sener and Silvio Savarese. 2018. Active learning for convolutional neural networks: A core-set approach. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*.

Burr Settles, Mark Craven, and Soumya Ray. 2007. Multiple-instance active learning. In *Proceedings of the 20th International Conference on Neural Information Processing Systems (NIPS)*, pages 1289–1296.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune BERT for text classification? In *Chinese Computational Linguistics - 18th China National Conference (CCL)*, pages 194–206.

Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Recurrently controlled recurrent networks. In *Proceedings of the Advances in Neural Information Processing Systems 31 (NeurIPS)*, pages 4731–4743.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Proceedings of the Advances in Neural Information Processing Systems 30 (NeurIPS)*, pages 5998–6008.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, and Quoc V. Salakhutdinov, Russ R a.nd Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Proceedings of the Advances in Neural Information Processing Systems 32 (NeurIPS)*, pages 5753–5763.

Michelle Yuan, Hsuan-Tien Lin, and Jordan Boyd-Graber. 2020. Cold-start active learning through self-supervised language modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7935–7948.

Mike Zhang and Barbara Plank. 2021. Cartography active learning. In *Findings of the Association for Computational Linguistics (EMNLP Findings)*, pages 395–406.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Proceedings of the Advances in Neural Information Processing Systems 28 (NIPS)*, pages 649–657.

Ye Zhang, Matthew Lease, and Byron C. Wallace. 2017. Active discriminative text representation learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 3386–3392.

Han Zhao, Zhengdong Lu, and Pascal Poupart. 2015. Self-adaptive hierarchical sentence model. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 4069–4076.

Wanshan Zheng, Zibin Zheng, Hai Wan, and Chuan Chen. 2019. Dynamically route hierarchical structure representation to attentive capsule for text classification. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 5464–5470.

Supplementary Material

The experiments can be reproduced using the code that is referenced on the first page[2]. In the following, we summarize important details for reproduction, including details on the results.

A Technical Environment

All experiments were conducted within a Python 3.8 environment. The system had CUDA 11.1 installed and was equipped with an NVIDIA GeForce RTX 2080 Ti (11GB VRAM). Computations for fine-tuning transformers and training KimCNN were performed on the GPU .

B Implementation Details

Our experiments were built using well-known machine learning libraries: PyTorch[3], huggingface transformers[4], scikit-learn[5], scipy[6], and numpy[7]. For active learning and text classification, we used small-text[8] (Schröder et al., 2022).

C Experiments

Each experiment configuration represents a combination of model, dataset and query strategy, and has been run for five times. We used a class-balanced initial set to support the warm start of the first model for the imbalanced TREC-6 dataset, whose rarest class would otherwise only rarely be encountered if sampled randomly.

C.1 Pre-Trained Models

We fine-tuned DistilRoBERTa (distilroberta-base) and BERT-large (bert-large-uncased). Both of them are available via the huggingface model repository.

Dataset	Max. Seq. Length
AGN	60
CR	50
MR	60
SUBJ	50
TREC	40

Table 6: Hyperparameter settings for the maximum sequence length (as number of tokens) per dataset.

[2] https://github.com/webis-de/ACL-22
[3] https://pytorch.org/, 1.8.0
[4] https://github.com/huggingface/transformers, 4.11.0
[5] https://scikit-learn.org/, 0.24.0
[6] https://www.scipy.org/, 1.6.0
[7] https://numpy.org/, 1.19.5
[8] https://github.com/webis-de/small-text, 1.0.0a8

Dataset	Model	Query Strategy				
		PE	BT	LC	CA	RS
AGN	SVM	0.804 ± 0.000	0.804 ± 0.000	0.802 ± 0.009	0.539 ± 0.088	0.801 ± 0.006
	KimCNN	0.871 ± 0.004	0.874 ± 0.005	0.856 ± 0.012	0.814 ± 0.015	0.866 ± 0.007
	DistilRoBERTa	0.892 ± 0.002	0.894 ± 0.003	0.894 ± 0.002	0.894 ± 0.008	0.879 ± 0.008
	BERT	0.896 ± 0.003	**0.904 ± 0.002**	0.894 ± 0.006	0.889 ± 0.014	0.884 ± 0.003
CR	SVM	0.757 ± 0.000		0.755 ± 0.014	0.742 ± 0.022	0.763 ± 0.025
	KimCNN	0.765 ± 0.012		0.762 ± 0.012	0.748 ± 0.015	0.745 ± 0.014
	DistilRoBERTa	0.906 ± 0.007		0.911 ± 0.008	0.905 ± 0.011	0.886 ± 0.007
	BERT	0.904 ± 0.010		**0.919 ± 0.009**	0.913 ± 0.005	0.896 ± 0.008
MR	SVM	0.674 ± 0.000		0.650 ± 0.012	0.633 ± 0.014	0.641 ± 0.010
	KimCNN	0.719 ± 0.011		0.719 ± 0.017	0.726 ± 0.008	0.720 ± 0.013
	DistilRoBERTa	0.819 ± 0.012		0.826 ± 0.009	0.826 ± 0.011	0.809 ± 0.011
	BERT	**0.857 ± 0.009**		0.852 ± 0.009	0.856 ± 0.015	0.846 ± 0.011
SUBJ	SVM	0.843 ± 0.000		0.857 ± 0.006	0.827 ± 0.012	0.839 ± 0.012
	KimCNN	0.897 ± 0.004		0.880 ± 0.008	0.877 ± 0.010	0.896 ± 0.009
	DistilRoBERTa	0.944 ± 0.004		0.948 ± 0.008	0.939 ± 0.008	0.926 ± 0.005
	BERT	0.957 ± 0.004		**0.958 ± 0.005**	0.954 ± 0.005	0.949 ± 0.003
TREC-6	SVM	0.740 ± 0.000	0.758 ± 0.000	0.692 ± 0.101	0.596 ± 0.145	0.742 ± 0.031
	KimCNN	0.840 ± 0.016	0.836 ± 0.012	0.834 ± 0.015	0.802 ± 0.017	0.792 ± 0.020
	DistilRoBERTa	0.942 ± 0.008	0.950 ± 0.009	0.942 ± 0.009	0.940 ± 0.011	0.918 ± 0.016
	BERT	0.932 ± 0.010	0.947 ± 0.014	0.960 ± 0.006	**0.968 ± 0.004**	0.921 ± 0.025

Table 5: Final accuracy per dataset, model, and query strategy. We report the mean and standard deviation over five runs. The best result per dataset is printed in bold.

C.2 Datasets

Our experiments used datasets that are well-known benchmarks in text classification and active learning. All datasets have been made accessible to the Python ecosystem by several Python libraries that provide fast access to the raw text of those datasets. We obtain CR and SUBJ using gluonnlp, and AGN, MR, and TREC using huggingface datasets.

C.3 Hyperparameters

Maximum Sequence Lenght We set the maximum sequence length to the minimum multiple of ten for which 95% of the respective dataset's sentences contain less than or an equal number of tokens for both KimCNN and transformers (shown in Table 6).

Transformers AGN is trained for 50 epochs and all other datasets for 15 epochs (Howard and Ruder, 2018). For training, we use AdamW (Loshchilov and Hutter, 2019) with a learning rate of $\eta = 2\text{e}{-5}$, beta coefficients of $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and an epsilon of $\epsilon = 1\text{e}{-8}$. Training is done in batches, with a batch size of 12.

KimCNN We adopt the parameters by Zhang et al. (2017), i.e., 50 filters and filter heights of $(3, 4, 5)$. Training is done in batches with a batch size of 25, a learning rate of $\eta = 1\text{e}{-3}$, and word embeddings from word2vec (Mikolov et al., 2013).

D Standard Deviations and Runtimes

In Table 5 and Table 7 we report final accuracy and AUC scores including standard deviations, measured after the last iteration of active learning. Moreover, we report the runtimes of the query step per strategy in Table 8.

D.1 Evaluation Metrics

Active learning was evaluated using standard active learning metrics, namely accuracy und area under the learning curve. For both metrics, the respective scikit-learn implementation was used.

References

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 328–339.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings 1st International Conference on Learning Representations (ICLR)*.

Dataset	Model	Query Strategy				
		PE	BT	LC	CA	RS
AGN	SVM	0.693 ± 0.000	0.705 ± 0.000	0.690 ± 0.011	0.458 ± 0.057	0.699 ± 0.012
	KimCNN	0.753 ± 0.005	0.791 ± 0.013	0.739 ± 0.019	0.699 ± 0.022	0.810 ± 0.013
	DistilRoBERTa	0.855 ± 0.018	**0.875 ± 0.007**	0.852 ± 0.018	0.863 ± 0.020	0.855 ± 0.006
	BERT	0.858 ± 0.015	0.872 ± 0.005	0.848 ± 0.018	0.864 ± 0.012	0.849 ± 0.007
CR	SVM	0.717 ± 0.000		0.713 ± 0.009	0.695 ± 0.009	0.718 ± 0.007
	KimCNN	0.713 ± 0.015		0.717 ± 0.009	0.707 ± 0.004	0.705 ± 0.014
	DistilRoBERTa	0.874 ± 0.012		0.875 ± 0.008	0.853 ± 0.019	0.870 ± 0.010
	BERT	**0.877 ± 0.011**		0.857 ± 0.016	0.866 ± 0.017	0.868 ± 0.008
MR	SVM	0.612 ± 0.000		0.615 ± 0.012	0.584 ± 0.018	0.597 ± 0.004
	KimCNN	0.674 ± 0.009		0.683 ± 0.015	0.671 ± 0.009	0.677 ± 0.011
	DistilRoBERTa	0.784 ± 0.013		0.786 ± 0.026	0.785 ± 0.010	0.783 ± 0.007
	BERT	**0.833 ± 0.013**		0.831 ± 0.012	0.817 ± 0.009	0.827 ± 0.006
SUBJ	SVM	0.801 ± 0.000		0.802 ± 0.003	0.768 ± 0.008	0.797 ± 0.010
	KimCNN	0.859 ± 0.013		0.841 ± 0.007	0.838 ± 0.011	0.864 ± 0.008
	DistilRoBERTa	0.924 ± 0.006		0.925 ± 0.003	0.915 ± 0.015	0.902 ± 0.008
	BERT	0.939 ± 0.007		0.938 ± 0.016	**0.943 ± 0.005**	0.933 ± 0.005
TREC-6	SVM	0.491 ± 0.000	0.648 ± 0.000	0.538 ± 0.085	0.462 ± 0.112	0.619 ± 0.026
	KimCNN	0.711 ± 0.010	0.714 ± 0.009	0.683 ± 0.029	0.639 ± 0.025	0.688 ± 0.013
	DistilRoBERTa	0.840 ± 0.023	0.864 ± 0.014	0.860 ± 0.013	0.842 ± 0.005	0.856 ± 0.020
	BERT	0.789 ± 0.032	0.844 ± 0.013	0.858 ± 0.030	**0.868 ± 0.027**	0.828 ± 0.018

Table 7: Final AUC per dataset, model, and query strategy. We report the mean and standard deviation over five runs. The best result per dataset is printed in bold.

Dataset	Model	Query Strategy									
		PE		BT		LC		CA		RS	
AGN	SVM	1.852 ±	0.415	0.907 ±	0.203	**0.432** ±	**0.097**	516.554 ±	115.583	0.001 ±	0.000
	KimCNN	7.264 ±	1.626	**6.199** ±	**1.389**	10.256 ±	2.359	481.758 ±	142.013	0.002 ±	0.000
	DistilRoBERTa	97.479 ±	21.800	96.372 ±	21.551	**87.398** ±	**19.560**	852.457 ±	230.157	0.002 ±	0.000
	BERT	528.884 ±	118.347	503.454 ±	112.583	**480.401** ±	**107.422**	1475.960 ±	391.579	0.002 ±	0.000
CR	SVM	0.005 ±	0.001	0.005 ±	0.001	**0.003** ±	**0.001**	0.307 ±	0.070	0.000 ±	0.000
	KimCNN	0.184 ±	0.042	**0.155** ±	**0.035**	0.163 ±	0.036	0.705 ±	0.189	0.000 ±	0.000
	DistilRoBERTa	1.942 ±	0.434	1.916 ±	0.428	**1.912** ±	**0.428**	2.627 ±	0.648	0.000 ±	0.000
	BERT	**12.112** ±	**2.709**	12.374 ±	2.767	12.427 ±	2.780	12.750 ±	2.852	0.000 ±	0.000
MR	SVM	0.014 ±	0.003	0.014 ±	0.003	**0.009** ±	**0.002**	1.889 ±	0.425	0.000 ±	0.000
	KimCNN	0.521 ±	0.117	**0.436** ±	**0.098**	0.468 ±	0.105	3.672 ±	1.098	0.000 ±	0.000
	DistilRoBERTa	7.558 ±	1.691	7.481 ±	1.673	**7.183** ±	**1.627**	12.303 ±	3.293	0.000 ±	0.000
	BERT	**41.428** ±	**9.265**	42.247 ±	9.447	41.960 ±	9.391	43.480 ±	9.747	0.000 ±	0.000
SUBJ	SVM	0.014 ±	0.003	0.013 ±	0.003	**0.009** ±	**0.002**	1.969 ±	0.444	0.000 ±	0.000
	KimCNN	0.472 ±	0.106	**0.409** ±	**0.091**	1.708 ±	1.144	3.161 ±	0.954	0.000 ±	0.000
	DistilRoBERTa	5.219 ±	1.167	5.153 ±	1.153	**5.099** ±	**1.140**	10.508 ±	2.885	0.000 ±	0.000
	BERT	**31.332** ±	**7.006**	32.908 ±	7.358	33.043 ±	7.393	37.832 ±	8.478	0.000 ±	0.000
TREC-6	SVM	0.085 ±	0.019	0.042 ±	0.009	**0.018** ±	**0.004**	0.609 ±	0.138	0.000 ±	0.000
	KimCNN	0.289 ±	0.065	**0.248** ±	**0.055**	1.111 ±	0.745	1.504 ±	0.447	0.000 ±	0.000
	DistilRoBERTa	2.934 ±	0.656	**2.887** ±	**0.646**	3.239 ±	1.473	4.691 ±	1.271	0.000 ±	0.000
	BERT	14.577 ±	3.260	**14.539** ±	**3.251**	14.963 ±	3.350	17.901 ±	17.213	0.000 ±	0.000

Table 8: Query time in seconds. We report the mean and standard deviation over five runs. The best result (with the lowest query time) per dataset and model is printed in bold.

Christopher Schröder, Lydia Müller, Andreas Niekler, and Martin Potthast. 2022. Small-Text: Active learning for text classification in python. *arXiv preprint arXiv:2107.10314*.

Ye Zhang, Matthew Lease, and Byron C. Wallace. 2017. Active discriminative text representation learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 3386–3392.