Low-resource Interactive Active Labeling for Fine-tuning Language Models

Seiji Maekawa[1]*, **Dan Zhang**[2], **Hannah Kim**[2], **Sajjadur Rahman**[2], **Estevam Hruschka**[2]
[1] Osaka University, [2] Megagon Labs
maekawa.seiji@ist.osaka-u.ac.jp,
{dan_z,hannah,sajjadur,estevam}@megagon.ai

Abstract

Recently, active learning (AL) methods have been used to effectively fine-tune pre-trained language models for various NLP tasks such as sentiment analysis and document classification. However, given the task of fine-tuning language models, understanding the impact of different aspects on AL methods such as labeling cost, sample acquisition latency, and the diversity of the datasets necessitates a deeper investigation. This paper examines the performance of existing AL methods within a low-resource, interactive labeling setting. We observe that existing methods often underperform in such a setting while exhibiting higher latency and a lack of generalizability. To overcome these challenges, we propose a novel active learning method TYROGUE that employs a hybrid sampling strategy to minimize labeling cost and acquisition latency while providing a framework for adapting to dataset diversity via user guidance. Through our experiments, we observe that compared to SOTA methods, TYROGUE reduces the labeling cost by up to 43% and the acquisition latency by as much as $11X$, while achieving comparable accuracy. Finally, we discuss the strengths and weaknesses of TYROGUE by exploring the impact of dataset characteristics.

1 Introduction

While fine-tuning pre-trained language models has become a standard practice for NLP tasks, data labeling remains a major bottleneck for NLP. To alleviate this problem, active learning (AL) has been recently employed to fine-tune language models for downstream tasks (Ash et al., 2020; Yuan et al., 2020; Margatina et al., 2021, 2022). Active learning aims to reduce the human labeling effort by focusing on the most informative samples that can enhance model performance efficiently.

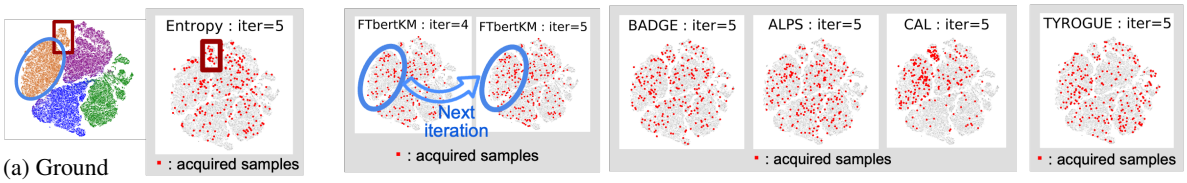
Unfortunately, even with state-of-the-art AL approaches, the number of labels needed to fine-tune language models is still significant. However, the availability of suitable annotators is often scarce, and obtaining human annotation can be expensive. For example, labeling tens of thousands of data samples may be impractical for domains such as medical or legal, considering the cost and time for labeling as well as the overhead of finding and training domain experts. This paper focuses on the low-resource setting where less than 1,000 samples are labeled in total, following Grießhaber et al. (2020); Yuan et al. (2020); Schröder et al. (2022).

Given such a low-resource setup, another challenge is the interactivity of AL methods. AL interactivity is often not considered in existing literature, but annotators' waiting time between labeling iterations can be a significant bottleneck. In addition, low latency is essential for the early stages of model building, where NLP researchers and practitioners aim to explore the model performance over faster AL iterations. However, when selecting samples, existing methods operate over the entire unlabeled dataset leading to higher latency in acquiring labeling candidates. As a result, the acquisition time for SOTA methods often violates the time-constraint of interactive systems (Liu and Heer, 2014). Therefore, we argue that an interactive low-resource acquisition strategy must balance the trade-off between labeling budget and the model performance, as they acquire samples while ensuring faster turnaround time.

Another relevant aspect we observe through experiments is that state-of-the-art AL approaches acquire redundant samples. Existing AL acquisition strategies focus on uncertainty (Houlsby

*This work was done when Seiji Maekawa was a research intern at Megagon Labs.

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 3230 - 3242
December 7-11, 2022



(a) Ground truth. (b) Uncertainty-based (c) Diversity-based (d) Hybrid. (e) Proposed.

Figure 1: Illustration of the sample redundancy challenge on AgNews dataset (Zhang et al., 2015). (a) shows 2D projection of BERT embeddings, where colors indicate ground truth class labels. (b) Uncertainty-based methods tend to acquire similar data points from a specific area within an iteration (see the red box). (c) Diversity-based methods tend to acquire data points similar to the samples acquired in previous iterations (see the blue circles). (d) Hybrid methods may suffer from either sample redundancies depending on which objective they prioritize, *i.e.*, diversity (BADGE (Ash et al., 2020) and ALPS (Yuan et al., 2020)) vs. uncertainty (CAL (Margatina et al., 2021)). (e) Our proposed method TYROGUE balances diversity and uncertainty by acquiring samples that are diverse and also closer to the model decision boundary. We use t-SNE (Van Der Maaten, 2014) to project the embeddings.

et al., 2011; Gal et al., 2017) or diversity (Sener and Savarese, 2018; Bādoiu et al., 2002; Gissin and Shalev-Shwartz, 2019), or both (Yuan et al., 2020; Margatina et al., 2021). Uncertainty-based methods select samples near decision boundaries, whereas diversity-based methods try to obtain diverse samples. As shown in Figure 1b, uncertainty-based methods acquire similar samples within an AL iteration — *intra-iteration redundancy*. Diversity-based AL approaches acquire similar samples across iterations — *inter-iteration redundancy* (Figure 1c). Existing hybrid methods aiming to balance uncertainty and diversity objectives surprisingly prioritize one objective over another and do not provide any mechanism to control the balance between these objectives. Therefore, even existing hybrid approaches also suffer from the aforementioned redundancies. Such redundancy leads to a wasted labeling budget, indicating room for reducing labeling costs across AL iterations. Moreover, as existing methods optimize for specific objectives, *i.e.*, uncertainty, and diversity, it is unclear whether these approaches generalize to datasets of varying degrees of difficulty with respect to scale and domain diversity.

Taking these observations into consideration, we propose TYROGUE [1], a low-resource interactive active learning method that minimizes labeling budget by reducing sample redundancy while achieving comparable accuracy. TYROGUE adopts a hybrid strategy where it incorporates independent steps implementing diversity and uncertainty sampling while instrumenting a control parameter to adapt the degree of influence of each objective during acquisition. Moreover, TYROGUE performs an initial filter on the unlabeled data to reduce the candidate sampling pool, thereby reducing latency without negatively impacting model performance. Our contributions are the following:

- We identify design criteria for developing low-resource interactive AL methods that address three challenges: high labeling cost, high acquisition latency, and sample redundancy.
- We propose a method, TYROGUE, that implements the design criteria, making it suitable for adoption in an interactive low-resource setup. TYROGUE outperforms SOTA methods (Ash et al., 2020; Yuan et al., 2020; Margatina et al., 2021) in terms of both effectiveness and efficiency. Compared to SOTA methods, TYROGUE reduces:
  - labeling cost by up to 43% while achieving comparable accuracy.
  - acquisition latency by as much as $11X$.
- Finally, we explore how aspects such as domain diversity and class distribution may impact the model's performance. We also discuss how user-guided adaptation strategies can ensure consistent performance in terms of accuracy across datasets.

2 Related Work

We now discuss related work relevant to our setup.

Learning with pre-trained LMs. Two popular

[1] Tyrogue is a Pokèmon that evolves in its appearance depending on the situation https://bulbapedia.bulbagarden.net/wiki/Tyrogue_(Pok%C3%A9mon)

methods for learning with pre-trained language models (LMs) are few-shot in-context learning (ICL) and fine-tuning. ICL enables pre-trained LMs to perform a new task without any gradient-based training by providing a small number of training examples as part of the input. However, ICL lacks interactivity as it processes all of the training examples for each prediction made, incurring significant storage and computational costs (Liu et al., 2022). Fine-tuning, on the other hand, trains parameters to enable a model to perform the new task. We specifically focus on active learning strategies to acquire samples for fine-tuning.

Several settings for active learning have been proposed over the years (Settles, 2009). These methods can be either instance-based, *i.e.*, acquiring a single data point, or batch-based, *i.e.*, acquiring a collection of data points. Batch active learning is suitable for meaningfully fine-tuning language models. In this paper, we adopt the batch active learning as in recent existing studies (Ash et al., 2020; Yuan et al., 2020; Citovsky et al., 2021; Margatina et al., 2021, 2022).

Acquisition strategies for batch AL. As mentioned in Section 1, uncertainty-based methods employ various techniques such as leveraging the model's predictive entropy, predictive confidence, and mutual information, to select data points for annotation (Cohn et al., 1996; Houlsby et al., 2011; Gal et al., 2017). Diversity-based methods employ strategies such as core-set construction-based sampling (Bādoiu et al., 2002) and discriminative learning (Gissin and Shalev-Shwartz, 2019) to select data points that are representative of the unlabeled data pool. All of these approaches optimize for the accuracy of the model and do not capture other aspects such as labeling budget, acquisition latency, and generalizability to diverse datasets. TYROGUE aims to address all three challenges for fine-tuning language models for NLP tasks.

Hybrid acquisition strategies combine uncertainty and diversity sampling. To combine both objectives, both BADGE (Ash et al., 2020) and ALPS (Yuan et al., 2020) leverage model uncertainty to compute embeddings of data points and then perform clustering to acquire diverse samples. Cluster-Margin (Citovsky et al., 2021) operates in large batches, *i.e.*, selecting 100k data points, and combines hierarchical clustering with predictive uncertainty to acquire data points. These methods ignore the interactive setting and employ clustering over the entire unlabeled pool which can be time-consuming. CAL (Margatina et al., 2021) balances the trade-off between uncertainty and diversity by acquiring contrastive data points that are similar in the model feature space but differ in model's predictive likelihoods. To operate efficiently, CAL requires a larger batch size than expected in a low-resource setting. Hence, these hybrid methods are not suitable for the low-resource and interactive setting. Moreover, they use a single acquisition objective combining diversity and uncertainty without any control afforded to the user. As a result, they tend to prioritize one objective over another (e.g., CAL is highly uncertainty-focused), depending on the algorithm designs even though datasets may have different characteristics. Unlike these approaches, TYROGUE introduces a hybrid acquisition strategy with tunable control parameters, which is suitable for the low-resource and interactive setting.

Low-resource active learning for LMs. Schröder et al. (2022) explore the performance of state-of-the-art uncertainty-based active learning methods in a low-resource setting while fine-tuning language models such as BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019). The authors highlight that there is significant drop in performance of the models in a low-resource setting. In our work, besides uncertainty-based methods, we investigate the performance of diversity-based and hybrid methods in a low-resource setting while exploring additional dimensions such as interactivity and dataset diversity. While ALPS (Yuan et al., 2020) also operates in low-resource setting, it's not interactive due to the higher latency in clustering the entire unlabeled data pool. Hence, ALPS is not suitable for the interactive setting. Finally, Grießhaber et al. (2020) also explore low-resource active learning with regards to aspects such as model update strategies which is beyond the scope of this paper.

3 Low-resource Interactive AL

We now present TYROGUE, a novel acquisition function that performs effectively and efficiently in the low-resource and interactive setting. We

first explain the limitations of existing AL methods and then describe our problem setting. We then describe key design criteria and the corresponding pipeline for TYROGUE to overcome the limitations.

3.1 Limitations of Existing Methods

High labeling cost. Several existing studies (Citovsky et al., 2021; Margatina et al., 2021, 2022) assume that annotators label a large batch of data points per iteration, *e.g.*, larger than 1% of data points in an unlabeled pool. Labeling such a large batch is impractical when we have large-scale datasets. Hence, we focus on small labeling budgets and more iterative acquisition, *i.e.*, the low-resource interactive setting.

High acquisition latency. Existing AL methods (Ash et al., 2020; Yuan et al., 2020; Margatina et al., 2021, 2022) consider the entire unlabeled pool during sampling, *i.e.*, constructing representations for all data points in the unlabeled data pool and/or executing clustering on the pool. As the acquisition time depends on the size of the unlabeled pool, existing methods do not scale to large unlabeled datasets — computing the representation for or executing clustering over all data points can be time-consuming. Hence, in an interactive setting, a new solution should avoid the computation over the entire unlabeled data pool.

Sample redundancy. Given the low-resource and interactive setting, it is crucial to acquire samples that are critical for fine-tuning the model and salient such that acquisition time is not wasted on redundant samples. As shown in Figure 1, we observed two sample redundancies in existing methods: intra- and inter-iteration redundancy.

Uncertainty-based methods tend to sample similar data points in the same iteration as they prioritize the model's predictive performance — all the data points selected based on uncertainty may be very similar (Figure 1b). Diversity-based methods, *e.g.*, FTbertKM (Yuan et al., 2020), tend to sample similar data across AL iterations as they emphasize acquiring diverse samples in the unlabeled pool while ignoring data points already labeled — the unlabeled pool may contain data points similar to annotated data (Figure 1c).

Existing hybrid approaches usually employ a single metric to both enhance diversity and reduce the uncertainty during sample acquisition. However, these methods tend to prioritize one objective over the other due to the design of their acquisition function. CAL's (Margatina et al., 2021) choice for selecting the most contrastive data points — assumed to enforce diversity — utilizes the model's predictive confidence, a metric employed by uncertainty-based methods. As CAL does not employ diversity measures, the selected contrastive points may be similar, causing intra-iteration redundancy. We empirically show that CAL acquires similar data points within an iteration (see Figure 1d). Methods such as BADGE (Ash et al., 2020) and ALPS (Yuan et al., 2020), on the other hand, prioritize diversity. As explained in Section 2, both approaches employ clustering on the uncertainty-based embeddings of the unlabeled data points and acquire candidates from diverse clusters, representing varying degrees of uncertainty. Such selection is prone to inter-iteration redundancy as data points similar to the labeled data and predicted with higher confidence by the model may be selected. Figure 1d shows that BADGE and ALPS exhibit similar sampling distribution to the diversity-based method FTbertKM. These redundancies indicate more room for further reducing the labeling budget than SOTA methods and performing active learning in a low-resource setting. Therefore, an AL method should aim to minimize such redundancies.

3.2 Design Criteria of TYROGUE

Motivated by the aforementioned limitations, we identify two key designs that can improve the effectiveness and efficiency of sample acquisition: **D1** reduce the unlabeled pool being considered for acquisition and **D2** decouple the diversity and uncertainty objectives in hybrid acquisition.

D1. Random sampling to reduce acquisition latency. The first design involves applying random sampling to an unlabeled data pool to obtain a smaller candidate set on which the acquisition function can then be applied. Such filtering reduces the latency of acquisition, a bottleneck in applying existing methods in an interactive setting. While existing methods only focus on the model performance (i.e., selecting informative data samples to fine-tune language models), we focus on both the model performance and acquisition latency, leading to an accuracy-latency trade-off. Random sam-

pling enables us to execute inference and clustering on a small subset of the unlabeled pool. Despite the significant computational cost reduction, we show empirically that such sampling does not hurt performance much in a low-resource setting.

D2. Employing diversity and uncertainty sampling independently to reduce redundancy. The second design proposes effectively combining diversity and uncertainty sampling to avoid intra- and inter-iteration redundancies. As mentioned in Section 3.1, existing hybrid methods may suffer from these redundancies due to unifying the uncertainty and diversity objectives into a single acquisition function — such strategies often exhibit affinity toward one objective over another. The basic idea is a two-step selection; executing 1) diversity sampling, *e.g.*, selecting cluster centers, to reduce intra-iteration redundancy and 2) uncertainty sampling, *e.g.*, selecting data points with high entropy, to avoid inter-iteration redundancy.

In the first step, by using diversity sampling, we select a subset of an unlabeled data pool consisting of diverse data points in the BERT embedding space. As for the second step, by using uncertainty sampling, we acquire data points from the subset, which are predicted with low confidence by the current model. We assume that data points in an unlabeled pool are predicted with high confidence if they are similar to those in the labeled pool. Uncertainty sampling is expected to mitigate the inter-iteration sample redundancy since it selects data points with low model confidence. In fact, we empirically show that our method of explicitly combining diversity and uncertainty sampling outperforms a diversity-based method FTbertKM (Yuan et al., 2020) suffering from the sample redundancy across iterations (see Figure 1c and 1e).

We employ diversity sampling first as opposed to uncertainty sampling to avoid intra-iteration redundancy — uncertainty-based acquisition strategies are prone to obtaining redundant samples as they prioritize the model's predictive performance. Moreover, in a low-resource setting, in earlier active learning iterations, the data points can be potentially less informative since the model is trained on insufficient data. In our experiments, we empirically show that an uncertainty-based method such as Entropy (Wang and Shang, 2014) does not perform well in the low-resource setting due to the lack of sample diversity as shown in Figure 1b. Concretely, Entropy acquires many similar data points from a specific area within an iteration even though the area is not close to a decision boundary for the model trained on the entire AgNews dataset (Zhang et al., 2015) (see the red box in Figure 1a). Hence, we execute diversity sampling first to ensure that acquired data points are diverse.

Algorithm 1 AL iterations

Require: labeled data $\mathcal{D}_l$, unlabeled data $\mathcal{D}_u$, acquisition size b, model $\mathcal{M}$, acquisition function $\mathcal{A}$

1: $\mathcal{D}_l = \{\}$
2: **for** iterations $t = 1, \ldots, T$ **do**
3: $\quad Q_t \leftarrow$ Acquire b data points by acquisition function $\mathcal{A}$ on model $\mathcal{M}$, data $\mathcal{D}_u$
4: $\quad \mathcal{D}_t \leftarrow$ Label acquired samples Q_t
5: $\quad \mathcal{D}_l = \mathcal{D}_l \cup \mathcal{D}_t$
6: $\quad \mathcal{D}_u = \mathcal{D}_u \backslash \mathcal{D}_t$
7: $\quad \mathcal{M} \leftarrow$ Fine-tune $\mathcal{M}$ on $\mathcal{D}_l$
return $\mathcal{M}$

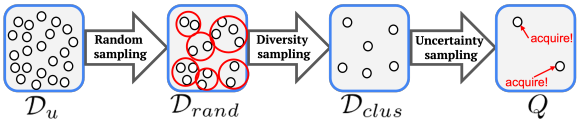


Figure 2: Overall pipeline of TYROGUE.

3.3 Problem Setting

We define $\mathcal{D}_l$ and $\mathcal{D}_u$ as a labeled and unlabeled data pool, respectively. Given these data pools, we perform AL for T iterations. For each iteration, we train a model on $\mathcal{D}_l$ and then acquire a batch Q consisting of b data points from $\mathcal{D}_u$ by using an acquisition function (see Algorithm 1). Then, the acquired data points are labeled (line 4), added to the labeled data pool $\mathcal{D}_l$ (line 5), and removed from the unlabeled data pool $\mathcal{D}_u$ (line 6). Following previous work (Margatina et al., 2021; Ash et al., 2020; Yuan et al., 2020), we simulate the AL setting, *i.e.*, we assume the data pool $\mathcal{D}_u$ to be unlabeled even though their labels are available in the benchmark and use the labels for evaluation. In our experiments, we fine-tune BERT model $\mathcal{M}$ (Devlin et al., 2018) at each AL iteration using the current labeled data pool $\mathcal{D}_l$ (line 7).

3.4 Pipeline of TYROGUE

Figure 2 outlines the detailed pipeline of TYROGUE in an AL iteration. First, TYROGUE obtains a subset $\mathcal{D}_{rand}$ of an unlabeled data pool $\mathcal{D}_u$ by drawing $|\mathcal{D}_{rand}|$ samples uniformly at random (see the "Random sampling" in the figure) from the unlabeled pool. This step ensures the data candidate pool for the next step is small enough to satisfy design criteria **D1**. Note that $|\mathcal{D}_{rand}|$ is a user-specified parameter which impacts the latency of the eventual acquisition. For example, a higher value of $|\mathcal{D}_{rand}|$ causes a higher latency during clustering due to the larger pool of data points. In our experiments we set $|\mathcal{D}_{rand}| = 10,000$, to ensure that the execution time of k-means is faster.

TYROGUE then performs diversity and uncertainty sampling in separate steps to enforce design criteria **D2**. We utilize a user-specified parameter r to control the trade-off between diversity and uncertainty while acquiring b samples. TYROGUE applies the k-means clustering algorithm to $l2$-normalized BERT embeddings of the randomly selected data points. TYROGUE first sets $k = r \times b$ as the clustering parameter and then selects the cluster centers as samples[2], thus enforcing diversity. From the $k = r \times b$ samples, TYROGUE acquires top-b data points based on the entropy of the current model's prediction, thus enforcing uncertainty. By appropriately setting r, TYROGUE can flexibly incorporate both diversity and uncertainty into its acquisition. Therefore, TYROGUE enables the users to control the degree of emphasis on one objective over another, unlike other hybrid strategies. For an extreme example, TYROGUE skips uncertainty sampling when $r = 1$ and diversity sampling when $r \geq |\mathcal{D}_{rand}|/b$.

4 Experiments

We now present the experiment set-up and results.

Setup. To demonstrate the reduction in labeling cost and acquisition latency, we compare TYROGUE with SOTA uncertainty-based (Entropy), diversity-based (FTbertKM), and hybrid (BADGE, ALPS, and CAL) methods. Note that Entropy is a baseline used in Margatina et al. (2021). We also include Random, which draws samples uniformly at random from the unlabeled pool. Random acquisition is non-active (or passive) as its sample selection does not depend on any model output.

We evaluate the methods by fine-tuning the pre-trained BERT in active iterations on seven datasets used by Margatina et al. (2021) and the PAWS-QQP dataset (Zhang et al., 2019). Used datasets are described in Table 1. Following the low-resource setting in Grießhaber et al. (2020), we evaluate our method with a total labeling budget of 1,000 samples. We set batch size per iteration as 50 samples to ensure interactivity. Note that CAL uses a labeled validation set to help the selection of samples. To ensure a fair comparison across methods, we exclude the validation step.

Implementations. We use the HuggingFace (Wolf et al., 2020) implementation of BERT-BASE (Devlin et al., 2018) with an additional classification layer. We use the open-source implementations of baseline methods used by CAL[3]. We repeat all experiments with five random seeds to get different initial model output layer weights and initial $\mathcal{D}_l$. Entropy, CAL, and BADGE start from an initial random sample set (warm start), while FTbertKM, ALPS, and TYROGUE utilize the active acquisition functions from the first iteration (cold start). For each active iteration, we train the model for three epochs. As for hyper-parameters we set $|\mathcal{D}_{rand}| = 10,000$ and $r = 3$ in following experiments based on empirical observations. In fact, we observe that $|\mathcal{D}_{rand}|$'s values >1K (5K, 10K, 20K) do not lead to significant accuracy differences (see Appendix B). We execute all experiments on a GPU node with 8 NVIDIA A100-SXM cores. More details can be found in Appendix A, and Appendix B.

4.1 Performance Evaluation

Labeling cost. To start with, we look at labeling cost reduction with respect to the number of labeled data points needed to achieve comparable prediction performance with the models fine-tuned on the entire training set (i.e., fully supervised). We set the target F1 score to be 85% and 95% of the fully supervised model. In Figure 3, we report the average number of data samples needed over five random trials for each acquisition function.

[2]Following existing approaches like FTbertKM and ALPS, we select samples closest to the cluster center.

[3]https://github.com/mourga/contrastive-active-learning

Table 1: Summary of datasets. Tasks: Sentiment Analysis (SA), Topic Classification (TC), Natural Language Inference (NLI), and Paraphrase Detection (PD).

Dataset	Task	Domain	Train/Val/Test	Classes
IMDB (Maas et al., 2011)	SA	Movie Reviews	22.5K/2.5K/25K	2
SST-2 (Socher et al., 2013)	SA	Movie Reviews	60.6K/6.7K/871	2
AgNews (Zhang et al., 2015)	TC	News	114K/6K/7.6K	4
DBPEDIA (Zhang et al., 2015)	TC	News	20K/2K/70K	14
PubMed (Dernoncourt and Lee, 2017)	TC	Medical	180K/30.2K/30.1K	5
QNLI (Wang et al., 2019)	NLI	Wikipedia	99.5K/5.2K/5.5K	2
PAWS-QQP (Zhang et al., 2019)	PD	Social QA Questions	10.8K/1.2K/677	2
QQP (Wang et al., 2019)	PD	Social QA Questions	327K/36.4K/80.8K	2

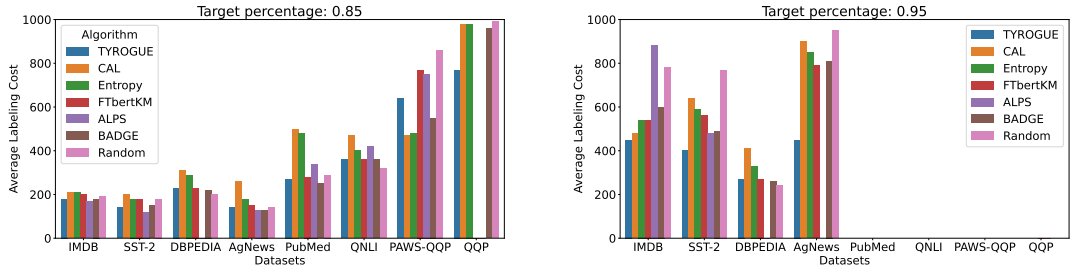


Figure 3: Average labeling cost (number of data samples) per iteration to achieve 85% and 95% of the F1 score by a model trained with the entire training set. With TYROGUE, models can achieve the same prediction F1 using up to 43% fewer labeled training examples compared to second best acquisition algorithm.

We only report the cost if a model achieves the target F1-score, leaving bars empty for failure cases. Moreover, we do not report the performance of FTbertKM and ALPS for QQP, the largest dataset, due to computing resource constraints in performing k-means clustering on the entire unlabeled data.

For all datasets tested, with a good acquisition algorithm, models can reach the target of 85% of the fully-supervised performance with 1,000 actively selected training samples. The batch size is less than 1% for large datasets like QQP and PubMed. In our low-budget setting, models achieve the highest target of 95% for half of the used datasets. TYROGUE reduces the labeling cost by up to 43% to achieve the same accuracy compared with the second-best method FTbertKM on AgNews. At 95% target accuracy, TYROGUE is the best-performing algorithm except for DBPEDIA. At 85% target accuracy, TYROGUE is among the 2 best performing algorithms except for DBPEDIA and PAWS-QQP. PAWS-QQP is an artificial dataset derived from the QQP corpus and the DBPEDIA dataset. We look closely at their characteristics in Section 4.2.

Interactivity. To ensure an interactive experience for iterative model development and debugging, the latency of acquisition algorithms matters. Figure 4 reports the time needed to select the next batch of samples to annotate for each acquisition method, averaged over all active iterations and five random trials. TYROGUE reduces the run time up to 11 times (compared with CAL on QQP) and is the fastest algorithm for six of the eight datasets. Diversity-based methods are known to be slow due to the expensive embedding calculation and clustering for the entire corpus. CAL also tends to be slow since it needs to compute the nearest labeled

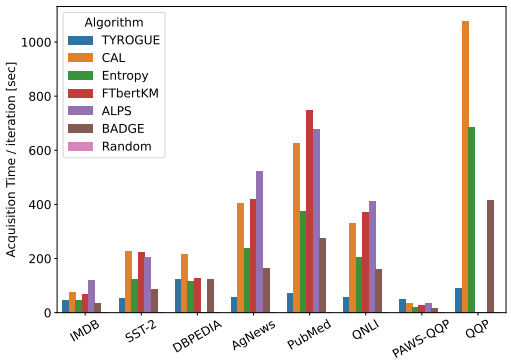
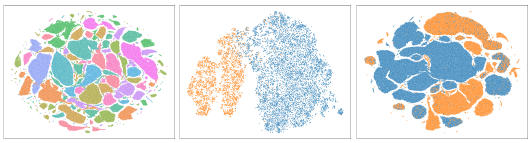


Figure 4: Average per-iteration acquisition time over 5 random runs. Unlike other approaches, TYROGUE's runtime does not increase with the size of the datasets , thereby, significantly reducing acquisition latency.

neighbors of data points in the entire unlabeled pool. Without compromising accuracy, TYROGUE can acquire samples in time which is dramatically less than the state-of-the-art algorithms. Our experiment's average training time per iteration is 2330s, comparable to the acquisition time for slower algorithms. Further, training can run in parallel with the acquisition in the next iteration. So reducing the acquisition time is crucial to the interactive experience of the active learning loops.

4.2 Impact of datasets and parameter r

Aside from an appropriate acquisition strategy, model performance depends on the characteristic of datasets and tasks. We empirically observe that larger datasets (e.g., QQP and PubMed) and data from a specialized domain (e.g., PubMed) are "harder" cases for actively fine-tuning under a limited budget. Complex tasks involving sentence or paragraph pairs (e.g., QNLI and the QQP datasets) are harder compared to sentiment and topic classification tasks.



(a) DBPEDIA (b) PAWS-QQP (c) QQP

Figure 5: Two-dimensional visualization of the embeddings by the BERT fine-tuned on the full training data set for (a) DBPEDIA, (b) PAWS-QQP, and its source (c) QQP. DBPEDIA dataset has diverse classes distributed over the space, requiring acquisition algorithms with good coverage and sample diversity. PAWS-QQP consists of artificially constructed pairs from the original QQP data, and unlike QQP, has a near-linear class separation. The simple class distribution reduces the need for overall sample coverage compared to QQP and favors uncertainty algorithms to acquire critical samples near the class boundaries. We use t-SNE (Van Der Maaten, 2014) to project the embeddings.

In Section 4.1, we demonstrated the effectiveness of TYROGUE in saving labeling cost and reducing acquisition latency. The benefit of incorporating uncertainty and diversity generalizes well across datasets except for two unique datasets. For DBPEDIA, diversity-based algorithms like FT-bertKM and hybrid methods that prioritize diversity like BADGE exhibit lower costs. All acquisition functions are outperformed by random selection, which can be considered a naive diversity-based acquisition strategy. While for PAWS-QQP, uncertainty-based like Entropy and hybrid methods that prioritize uncertainty like CAL perform better in terms of cost reduction.

Figure 5 shows two-dimensional visualizations of the embeddings for the two datasets and the original QQP dataset. The embeddings are generated by models trained on the entire training dataset, and colors represent the class labels. The scatterplots show that DBPEDIA data points from the 14 classes are distributed over disconnected clusters over the space. Models need diverse samples with good domain and class coverage, thus favoring diversity-heavy methods. On the other hand, PAWS-QQP consists of artificially constructed Adversarial pairs from the QQP data and is less diverse than its source. The near-linear class separation leads to a reduced need for sample diversity. In such a case, uncertainty-based samples near the decision boundary can help the model make better predictions in the ambiguous area.

The observations above highlight that the performance of an active learning method may vary depending on dataset characteristics such as scale and domain diversity. The design of TYROGUE enables users to control the balance between diversity and uncertainty by tuning the parameter r.

Figure 6 demonstrates the effect of varying parameter r for two datasets — PAWS-QQP[4] and DBPEDIA — representing two extreme cases. As shown in Figure 5b, PAWS-QQP is almost linearly separable and thus more suitable for the uncertainty-based acquisition, where acquiring samples only from the decision boundary may be sufficient for high-quality fine-tuning. The DBPEDIA dataset (Figure 5a), on the other hand, exhibits a more complex class boundary and requires a more diverse acquisition strategy to acquire representative samples. Generally, the larger r is, the TYROGUE tends to perform uncertainty-based acquisition. On the other hand, when $r = 1$, TYROGUE ignores the uncertainty-based sampling and performs diversity-based acquisition only. If $r = |\mathcal{D}_{rand}|/b$, TYROGUE skips the diversity sampling and per-

[4]Since PAWS-QQP is a harder dataset where no algorithm can reach the high 95% target accuracy, here we show results for a lower target 89%.

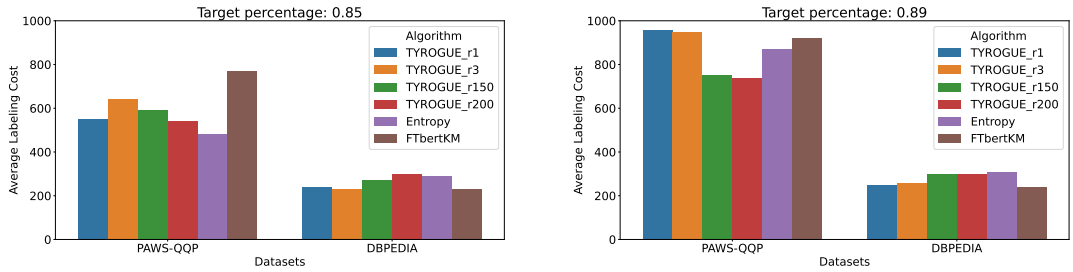


Figure 6: Imapct of parameter r varing in range $[1, |\mathcal{D}_{rand}|/b]$ on the special datasets PAWS-QQP and DBPEDIA, comapring with uncertainty-based (Entropy) and diversity-based (FTbertKM) algorithms.

forms uncertainty-based acquisition only. In keeping with these observations, the results in Figure 6 show that PAWS-QQP favors uncertainty-based methods, and DBPEDIA requires reasonable diversity in training samples. We believe such observations lay a solid foundation for future work on the design of automatically adaptive algorithms.

5 Conclusion and Future work

We present TYROGUE, a novel active learning method that overcomes the two major limitations of existing methods: 1) high acquisition latency and 2) sample redundancy. Through our experiments, we observe that TYROGUE runs faster than existing methods for larger datasets and reduces sample redundancy through effective combination of diversity and uncertainty, thereby reducing the labeling cost. We also observe how adaptability is crucial for obtaining consistent performance across datasets and identify the importance of instrumenting mechanisms to balance the uncertainty-diversity trade-off.

Towards adaptive acquisition. The trade-off between uncertainty and diversity is essential for active acquisition algorithms. We believe TYROGUE and the observations in this work lay the foundation for future work on adaptive acquisition functions that balance both objectives. Future extensions to our work can investigate strategies for attaining the optimal balance of uncertainty and diversity by taking into account aspects such as model performances and dataset characteristics.

Adoption in practical systems. The multi-step adaptive method proposed, TYROGUE can be incorporated into any annotation platforms. As outlined in Section 1, such frameworks can enable rapid iterations in the early stages of modeling building. Therefore, understanding how TYROGUE can be integrated into the existing annotation platforms is an interesting research problem.

Transparency and control for practitioners. Future studies may explore how users operate within the interactive AL framework. Our proposed design affords control to the users in balancing the acquisition dichotomy as mentioned above. However, it is imperative to understand how aspects such as transparency of the framework and interpretability of the model may impact users' experience as they reason over the control parameters.

6 Limitations

TYROGUE has been tested only on popular textual data classification tasks like sentiment analysis and pair classification like paraphrase detection.

We only used the base BERT model, which is pre-trained on the standard Wikipedia and book datasets, for all the diversity-based approaches. Schröder et al. (2022) utilized both BERT and DistilRoBERTa to test the performance of uncertainty-based methods. DistilRoBERTa showed similar results with a smaller model with fewer parameters. The conclusion opens up interesting future work on the choice of embedding models in low-resource scenarios. To work well on less represented domains such as scientific publications, customized pre-trained models may be necessary for our low-budget setting.

TYROGUE focuses on the acquisition steps in active learning iterations and assumes standard iterative model fine-tuning in batches. For example, with the training strategy that adapts to downstream application proposed by Margatina et al. (2022), our conclusions on the acquisition strategies may not generalize directly.

7 Acknowledgement

We thank Yoshi Suhara and Yuliang Li for their valuable suggestions on additional experiments. We also thank Makoto Onizuka and Yuki Arase for their insightful feedback.

References

Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. 2020. Deep batch active learning by diverse, uncertain gradient lower bounds. *ICLR*.

Mihai Bādoiu, Sariel Har-Peled, and Piotr Indyk. 2002. Approximate clustering via core-sets. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 250–257.

Gui Citovsky, Giulia DeSalvo, Claudio Gentile, Lazaros Karydas, Anand Rajagopalan, Afshin Rostamizadeh, and Sanjiv Kumar. 2021. Batch active learning at scale. *Advances in Neural Information Processing Systems*, 34.

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

Franck Dernoncourt and Ji Young Lee. 2017. PubMed 200k RCT: a dataset for sequential sentence classification in medical abstracts. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 308–313, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL*.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep Bayesian active learning with image data. In *ICML*, volume 70 of *Proceedings of Machine Learning Research*, pages 1183–1192. PMLR.

Daniel Gissin and Shai Shalev-Shwartz. 2019. Discriminative active learning. *arXiv preprint arXiv:1907.06347*.

Daniel Grießhaber, Johannes Maucher, and Ngoc Thang Vu. 2020. Fine-tuning BERT for low-resource natural language understanding via active learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1158–1171, Barcelona, Spain (Online). International Committee on Computational Linguistics.

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

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *ArXiv*, abs/2205.05638.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Zhicheng Liu and Jeffrey Heer. 2014. The effects of interactive latency on exploratory visual analysis. *IEEE transactions on visualization and computer graphics*, 20(12):2122–2131.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Katerina Margatina, Loïc Barrault, and Nikolaos Aletras. 2022. On the importance of effectively adapting pretrained language models for active learning. *ACL*.

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. 2021. Active learning by acquiring contrastive examples. *EMNLP*.

Christopher Schröder, Andreas Niekler, and Martin Potthast. 2022. Revisiting uncertainty-based query strategies for active learning with transformers. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2194–2203.

Ozan Sener and Silvio Savarese. 2018. Active learning for convolutional neural networks: A core-set approach. *ICLR*.

Burr Settles. 2009. Active learning literature survey.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Laurens Van Der Maaten. 2014. Accelerating t-sne using tree-based algorithms. *The Journal of Machine Learning Research*, 15(1):3221–3245.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*.

Dan Wang and Yi Shang. 2014. A new active labeling method for deep learning. In *2014 International joint conference on neural networks (IJCNN)*, pages 112–119. IEEE.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Michelle Yuan, Hsuan-Tien Lin, and Jordan Boyd-Graber. 2020. Cold-start active learning through self-supervised language modeling. *EMNLP*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. PAWS: Paraphrase Adversaries from Word Scrambling. In *Proc. of NAACL*.

A Experimental Details

A.1 Hyperparameters for Pre-trained Model

As for the detailed set up of BERT-BASE, we follow Margatina et al. (2021). Concretely, we train all models with batch size 16, learning rate $2e-5$, no weight decay, and AdamW optimizer with epsilon $1e-8$. For all datasets we use maximum sequence length of 128, except for IMDB that contains longer input texts, for which we use 256. The base model has 12 layers, 768 hidden, 12 heads and 110M parameters.

A.2 Dataset Split

If available, we use the default test and train splits provided for all datasets. Otherwise, we randomly sample a validation set from the training set and use the split ratio in Margatina et al. (2021) as follows. For IMDB, SST-2, QNLI, and PAWS-QQP, we randomly sample 10% from the training set to serve as the validation set. As for AgNews and QQP we sample 5%. For DBPEDIA, we undersample both training and validation sets from the standard splits. The reason that we prepare a validation set is to allow future researchers to easily utilize a validation set for their settings. For all datasets, we use the default test set.

A.3 Detailed Procedure for Model Training with Active Learning

There are two major approaches to fine-tune pre-trained LMs in active learning. First, some methods such as Citovsky et al. (2021) use the same model across iterations and iteratively train it using the accumulated labeled data points that are acquired over iterations. This approach may prioritize data points acquired in early AL iterations since they are used for training more times than those acquired in later AL iterations. Other methods such as Ash et al. (2020); Yuan et al. (2020); Margatina et al. (2021) initialize a model for each AL iteration and train the initialized pre-trained model by using the current labeled data points that are accumulated. This approach requires more training epochs per iteration since it needs to fine-tune an initialized model. To accommodate our low-resource interactive setting, we use the first approach – same model across iterations in our experiments.

Table 2: Labeling cost (number of data samples per iteration) with 95% confidence intervals to achieve target accuracy 85%, 90%, and 95% of the fully-supervised model.

	IMDB	SST-2	DBPEDIA	AgNews	PubMed	QNLI	PAWS-QQP	QQP
a) Target accuracy: 85%								
TYROGUE	180.0+/-30.4	140.0+/-46.5	230.0+/-84.2	140.0+/-46.5	270.0+/-63.3	360.0+/-72.4	640.0+/-154.1	770.0+/-206.3
CAL	210.0+/-72.4	200.0+/-78.5	310.0+/-46.5	260.0+/-149.0	500.0+/-171.2	470.0+/-231.0	470.0+/-186.7	980.0+/-49.7
Entropy	210.0+/-46.5	180.0+/-63.3	290.0+/-46.5	180.0+/-74.5	480.0+/-160.0	400.0+/-68.0	480.0+/-108.2	980.0+/-49.7
FTbertKM	200.0+/-55.5	180.0+/-63.3	230.0+/-63.3	150.0+/-39.3	280.0+/-74.5	360.0+/-106.8	770.0+/-282.1	-
ALPS	166.7+/-58.6	120.0+/-30.4	-	130.0+/-49.7	340.0+/-72.4	420.0+/-100.9	750.0+/-633.3	-
BADGE	180.0+/-49.7	150.0+/-55.5	220.0+/-63.3	130.0+/-49.7	250.0+/-0.0	360.0+/-132.6	550.0+/-117.8	960.0+/-72.4
Random	190.0+/-46.5	180.0+/-74.5	200.0+/-39.3	140.0+/-24.8	290.0+/-60.8	320.0+/-84.2	860.0+/-347.7	990.0+/-24.8
b) Target accuracy: 90%								
TYROGUE	220.0+/-30.4	190.0+/-46.5	260.0+/-91.2	180.0+/-49.7	420.0+/-108.2	900.0+/-117.8	950.0+/-124.2	-
CAL	250.0+/-68.0	260.0+/-91.2	370.0+/-63.3	320.0+/-144.8	800.0+/-238.8	790.0+/-226.9	820.0+/-186.7	-
Entropy	250.0+/-55.5	270.0+/-84.2	310.0+/-72.4	240.0+/-132.6	950.0+/-68.0	750.0+/-161.9	880.0+/-217.2	-
FTbertKM	260.0+/-72.4	240.0+/-99.3	250.0+/-103.9	180.0+/-74.5	600.0+/-152.1	-	920.0+/-198.7	-
ALPS	200.0+/-101.4	190.0+/-46.5	-	180.0+/-84.2	740.0+/-220.0	-	-	-
BADGE	250.0+/-78.5	230.0+/-74.5	230.0+/-63.3	190.0+/-72.4	500.0+/-141.6	860.0+/-226.9	990.0+/-24.8	-
Random	270.0+/-63.3	260.0+/-132.6	210.0+/-46.5	150.0+/-0.0	820.0+/-198.7	-	-	-
c) Target accuracy: 95%								
TYROGUE	450.0+/-87.8	400.0+/-87.8	270.0+/-92.9	450.0+/-87.8	-	-	-	-
CAL	480.0+/-84.2	640.0+/-276.0	410.0+/-60.8	900.0+/-68.0	-	-	-	-
Entropy	540.0+/-72.4	590.0+/-212.9	330.0+/-84.2	850.0+/-166.6	-	-	-	-
FTbertKM	540.0+/-126.6	560.0+/-149.0	270.0+/-84.2	790.0+/-168.4	-	-	-	-
ALPS	883.3+/-211.1	480.0+/-231.0	-	-	-	-	-	-
BADGE	600.0+/-157.1	490.0+/-209.2	260.0+/-72.4	810.0+/-154.1	-	-	-	-
Random	780.0+/-139.4	770.0+/-213.6	240.0+/-46.5	950.0+/-124.2	-	-	-	-

Table 3: Average acquisition time (in seconds) per iteration, averaged over 5 random trials.

	IMDB	SST-2	DBPEDIA	AgNews	PubMed	QNLI	PAWS-QQP	QQP
TYROGUE	46.6+/-1.0	54.4+/-1.6	122.2+/-4.5	58.3+/-0.8	73.8+/-2.3	58.4+/-3.3	48.1+/-2.0	90.7+/-1.7
CAL	74.8+/-0.7	227.0+/-35.4	217.6+/-1.5	402.8+/-26.0	628.0+/-43.4	330.4+/-4.7	36.1+/-0.8	1077.4+/-4.1
Entropy	46.3+/-0.0	125.5+/-0.3	117.7+/-0.8	236.8+/-0.5	376.3+/-2.0	204.7+/-0.1	21.9+/-0.0	684.5+/-2.2
FTbertKM	67.4+/-1.5	223.0+/-5.6	127.2+/-1.2	420.8+/-11.5	748.0+/-38.7	369.6+/-26.8	29.3+/-1.5	-
ALPS	119.4+/-4.2	205.9+/-8.1	-	524.4+/-10.8	678.7+/-23.8	411.3+/-4.3	36.1+/-1.1	-
BADGE	36.7+/-0.3	86.0+/-14.0	123.0+/-1.3	166.1+/-10.5	275.8+/-18.3	159.6+/-4.2	17.3+/-0.5	414.1+/-1.5
Random	0.0+/-0.0	0.0+/-0.0	0.1+/-0.0	0.0+/-0.0	0.1+/-0.0	0.1+/-0.0	0.0+/-0.0	0.2+/-0.0

Table 4: Labeling cost of Tyrogue with different hyperparameter $|\mathcal{D}_{rand}|$. Results shown with 95% confidence intervals to achieve target accuracy 85%, 90%, and 95% of the fully-supervised model. Increasing the size of the random filter (oversampling) beyond the default setting does not lead to significant changes in labeling costs.

	IMDB	SST-2	DBPEDIA	AgNews	PubMed	QNLI	PAWS-QQP	QQP
a) Target accuracy: 85%								
TYROGUE_DR5k	180.0+/-49.7	160.0+/-24.8	200.0+/-39.3	140.0+/-24.8	280.0+/-84.2	450.0+/-136.0	510.0+/-294.9	-
TYROGUE_DR10k	180.0+/-30.4	140.0+/-46.5	230.0+/-84.2	140.0+/-46.5	270.0+/-63.3	360.0+/-72.4	640.0+/-154.1	770.0+/-206.3
TYROGUE_DR20k	180.0+/-30.4	170.0+/-49.7	190.0+/-24.8	160.0+/-60.8	270.0+/-49.7	390.0+/-159.0	490.0+/-159.0	-
b) Target accuracy: 90%								
TYROGUE_DR5k	220.0+/-74.5	190.0+/-46.5	230.0+/-30.4	190.0+/-24.8	600.0+/-171.2	890.0+/-168.4	840.0+/-243.3	-
TYROGUE_DR10k	220.0+/-30.4	190.0+/-46.5	260.0+/-91.2	180.0+/-49.7	420.0+/-108.2	900.0+/-117.8	950.0+/-124.2	-
TYROGUE_DR20k	230.0+/-30.4	240.0+/-60.8	210.0+/-24.8	220.0+/-63.3	510.0+/-99.3	860.0+/-216.5	950.0+/-78.5	-
c) Target accuracy: 95%								
TYROGUE_DR5k	430.0+/-63.3	420.0+/-121.7	250.0+/-0.0	490.0+/-106.8	-	-	-	-
TYROGUE_DR10k	450.0+/-87.8	400.0+/-87.8	270.0+/-92.9	450.0+/-87.8	-	-	-	-
TYROGUE_DR20k	420.0+/-63.3	430.0+/-84.2	240.0+/-24.8	490.0+/-138.3	-	-	-	-

B Additional Results

In this section, we report detailed results of our experiments.

Labeling cost Table 2 shows total labeling costs to achieve target accuracy of 85%, 90%, and 95% (compared to fully supervised models) and their 95% confidence intervals. Table 2 (a) 85% and (c) 95% correspond to the results in Figure 3.

Acquisition latency Table 3 shows acquisition times spent per iteration (in seconds) and their 95% confidence intervals. Table 3 corresponds to the results in Figure 4. All results are averaged over five trials.

$|\mathcal{D}_{rand}|$ **settings** For accuracy/labeling cost, TYROGUE is not very sensitive to $|\mathcal{D}_{rand}|$ as long as it provides representative samples of the unlabeled pool. Empirically as shown in Table 4, we observed that values >1K (5K, 10K, 20K) did not lead to significant accuracy differences. Regarding efficiency, acquisition latency grows as $|\mathcal{D}_{rand}|$ gets larger.

C Efficiency

In this section, we compare the computational efficiency of the acquisition functions used in our experiments. TYROGUE requires $\mathcal{O}(rb|\mathcal{D}_{rand}|d)$, where d indicates the dimension of embeddings. This is mainly driven by k-means clustering step on $\mathcal{D}_{rand}$. Other compared methods, Entropy, FTbertKM, BADGE, ALPS, and CAL, require at least $\mathcal{O}(|\mathcal{D}_{pool}|)$, which is significantly higher than $\mathcal{O}(rb|\mathcal{D}_{rand}|d)$ in real applications. By performing the initial random sampling, TYROGUE can greatly reduce the scale of later operations like clustering and ranking. Hence, TYROGUE runs faster than other AL methods on middle- or large-sized datasets (detailed numbers are included in Table 3).