Harnessing the Potential of Pretrained Language Models and Active Learning for Tweets Sentiment Analysis

Marc-Antoine Allard, Antoine Magron, Paul Teiletche
{marc-antoine.allard, antoine.magron, paul.teiletche}@epfl.ch
CS-433 - Machine Learning

***Abstract*—This work proposes a study of tweets' sentiment analysis performances in conditions with low computational power. We develop different sampling strategies: Random, Uniform Aware Sampling, and Entropy Aware Sampling to select the subsample of the tweets that would contribute the most to efficient training. During later training, we try to reduce the computational costs by developing an active learning training adaptation. A first benchmark, comparing in similar settings, the potential performances of multiple models indicates that the best-performing model for our setup would be BERTweet, leveraging its pretraining and focusing on fine- tuning for our sentiment analysis task. We then perform deep fine-tuning of the latter, applying our designed active learning techniques and aware sampling dataset. An ablation study that active learning could achieve 99.7% of the performances of our best working model while speeding up its training by 25%, reducing significantly the training costs. Aware sampling turned out to produce poor results compared to a random sampling but its effects were mitigated by the use of Active Learning. The source code is available at https://github.com/CS-433/ml-project-2-apma_ai.**

I. Introduction

Effectively using Machine Learning for Natural Language Processing (NLP) tasks in the dynamic world of social media is challenging due to the vast and particular amount of user-generated content. Among the multitude of social media platforms, one platform distinguishes itself in providing a substantial amount of text for NLP tasks: Twitter[1]. This paper delves into the exploration of utilizing real Tweets to address the task of sentiment analysis.

Tweet texts serve as authentic sources of information and have already demonstrated widespread utility in various language-related tasks. The challenge lies in the fact that many contemporary Large Language Models (LLMs), such as BERT [1], are pretrained on book or Wikipedia sources, which differ significantly from the specific characteristics of tweets [2]. Tweets vary not only in length but also in language structure, often containing emojis or abbreviations. Achieving competitive results in sentiment analysis tasks with such data necessitates a substantial amount of fine-tuning to adapt pretrained models to these unique formats.

Our contribution involves exploring the use of Active Learning (AL) and Aware Sampling fine-tuning Pretrained Language Models (PLMs) over tweet data. This combined use aims to achieve high-performance in sentiment analysis while maintaining low computational costs.

Ethical Risks

Sentiment analysis plays a pivotal role in social media regulations, particularly concerning the Twitter platform, offering a range of benefits to foster a healthier online environment [3]. However, the ethical implications support the need for responsible implementation, and a keen awareness of the potential concerning impacts on society and environment [4].

When sentiment analysis is conducted on non-anonymized tweets, it may exposes users personal information and opinons [5]. This might be targeted by malicious actors, and the potential for doxing[2] [6] becomes a serious concern [7], emphasizing the need for robust privacy protection measures [8]. Also, the implementation of sentiment analysis algorithms for content regulation - particularly when users are aware of non-anonymized data exploitation - may create a chilling effect on the freedom of expression [9]. Users may become hesitant to express their genuine opinions or engage in controversial discussions, fearing that their words will be analyzed, misinterpreted, or used against them. The challenge of non-anonymized tweets is tackled by the dataset[3] that utilizes consistent tags (`<USER>` for users and `<URL>` for URLs) to encode this information. Nevertheless, there remains the potential for conducting a thorough analysis of tweet content, which could potentially reveal information allowing for the identification of the original author.

Our research addresses another significant concern regarding the environmental impact of LLMs used in tweet sentiment analysis. Strubell et al. [10] have conducted a benchmark, comparing the environmental cost of various model development and training processes with the average human consumption. Remarkably, a single BERTbase model training emits approximately 1.4 tons of CO_2, nearly equivalent to a flight from New York to San Francisco. Similarily, the overall environmental cost of developing the complete Transformer model [11] is estimated at 284 tons of

[1]Now referred as X

[2]Publishing private information with malicious intent

[3]https://www.aicrowd.com/challenges/epfl-ml-text-classification/dataset_files

CO_2, representing 17 years of an average American's CO_2 emissions. Given the pressing environmental challenges our world faces, the field of NLP and its research efforts must consider these environmental costs. Our contribution to addressing this concern involves exploring Active Learning. Our AL algorithm, clarified in section III, strategically divides the training set in half at each epoch, limiting the total runtime to only 2 epochs on the entire dataset. Considering that the BERTbase model was originally trained using 40 epochs [1], our approach theoretically reduces training costs by 20 while maintaining an equivalent model performance (see Table III). Furthermore, as discussed in section III, we investigate the potential of Aware Sampling to reduce the size of the initial training set while striving to maintain significant results.

II. Related Work

Previous approaches tackling NLP tasks with tweets data align with the remarkable advancements in the field of unsupervised learning for textual data. For instance, previous work delved into the application of classifier ensembles and lexicons to categorize the sentiment of tweets [12], [13]. Another notable effort aimed to establish a cutting-edge Twitter sentiment classifier by employing Convolutional Neural Networks (CNNs) and Large Long Short-Term Memory (LSTMs) networks [14]. However, the recent emergence of LLMs has surpassed these previous methods. Notably, tweets fine-tuned BERT model [15] and XLM-R model [16] presented significantly enhanced performance.

Nevertheless, as cited in section I, these models requires important computational resources, and one of the solution to overcome this challenge is to use Active Learning. As far back as 1994, we used entropy of the model decision for AL [17]. Later on, researchs unlocked new improvement by developping algorithm for AL on black box model [18], [19]. This flexibility allows us to develop ressembling algorithms.

III. Methods

Data

Our objective is to categorize tweets based on their conveyed sentiment, and for this purpose, we begin by examining a substantial set of tweets. We initiate our analysis by exploring the entire dataset and introduce methodologies for the selection of relevant subdatasets. Basic statics are computed to show the expected variations among the selections.

All tweets: The provided dataset contains, as mentioned in table I, 2 500 000 tweets. In our analysis we considered only two possible sentiment, *positive* and *negative*. The two classes are perfectly balanced within the initial set of data. Tweets are famously known to be short and convey information very efficiently. This can be confirmed when looking at figure 1. We can see that the tweets have on average only 14 words. Twitter was also famously known for its former limit of 140 characters, for which we see a peak in figure 1.

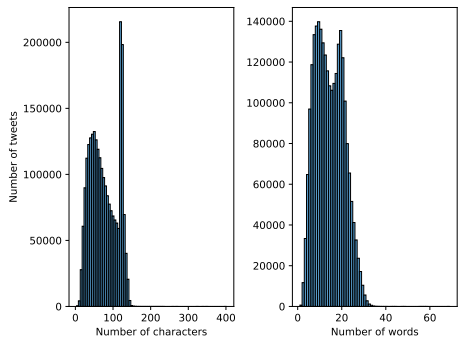

Figure 1. Number of character and words in the tweet datataset

Tweet selection: PLMs are notoriously very large models and can contains more than 100 million parameters. Training on the complete dataset would be ideal but is largely computationally unrealistic.

We use for the training subsamples of this dataset. A good subsamples contains a fair amount of tweets, but is signicantly smaller than the initial one and should be representative of the complete dataset, i.e. it should represent the original distributions. We present two subsampling methods.

Random sampling: We start by considering random subspampling of the complete dataset. This method straightforward by guarantees that the inner distribution are kept unchanged. Table I shows that when using random sampling, the positive and negative ratio is kept at balance and the average length of the tweets are also kept unchanged. The statistics are obtain by bootstrapping.

Aware sampling: As uniform sampling provide the most representative subsample of the entire dataset, we aim to train a machine learning model, so we try to come up with the most efficient subsampling. We try to understand the inner structure of the dataset by using the unsupervised learning algorithm KMeans.

We start by computing a vector representation of our tweets, an embeddings. Such a representation can be obtained with any PLM but we use BERTweet[15] for this purpose as it is specifically train to understand the semantic of such tweets. We do so by querying the last hidden state of the tweet's `CLS` token. The elbow methods applied on the distortion indicates an ideal amount number of 13 clusters. Figure 2 shows the population of each of these clusters and their inner class distribution. We use the assumption that selected from each of the computed cluster would represent the entire structure of the inner dataset.

We start by forming the **Uniform Aware Sampling** `unif_140` and `unif_200`. This dataset is composed by taking the same number of samples from each of the previouly computed cluster.

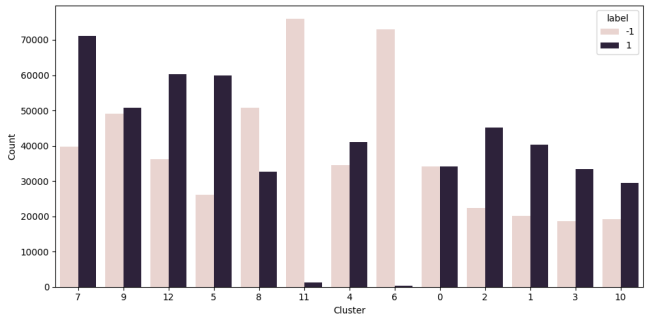

Figure 2. Distribution of the cluster's population and their class repartition

Let $\{\mathcal{C}_i\}_{i=1}^{13}$ be the set of clusters. Creating a **uniform aware sampling** dataset of size N corresponds to selecting randomly from each cluster D_i such that:

$$D_i = min\Big(\frac{N}{13}, min_i\big\{|\mathcal{C}_i|\big\}\Big) \tag{1}$$

We develop a second way to select from the cluster, the **Entropy away sampling**. The latter is defined by the will to over-represent in selection the cluster with a lot of uncertainty. The initial assertion is that a cluster with a great majority of sample in one class is way easier to classify for the model than a cluster with balance in its classes. To quantify this cluster uncertainty we use the entropy. Let C_i^p and C_i^n be the positive and negative samples in each cluster, respectively.

$$\mathbf{P}_p(\mathcal{C}_i) = \frac{|\mathcal{C}_i^p|}{|\mathcal{C}_i|} \; ; \; \mathbf{P}_n(\mathcal{C}_i) = \frac{|\mathcal{C}_i^n|}{|\mathcal{C}_i|}$$

$$H(\mathcal{C}_i) = -(\mathbf{P}_p(\mathcal{C}_i)log(\mathbf{P}_p(\mathcal{C}_i)) + \mathbf{P}_n(\mathcal{C}_i)log(\mathbf{P}_n(\mathcal{C}_i)) \tag{2}$$

Figure 3 shows the entropy of each sample. We apply a softmax function to transform these values into a probability distribution. A temperature of $t = 0.4$ is used to increase the probability difference.

$$\mathbf{P}(\mathcal{C}_i) = \frac{e^{H(\mathcal{C}_i)/t}}{\sum_{j=1}^{13} e^{H(\mathcal{C}_j)/t}} \tag{3}$$

Using the **Entropy Aware Sampling** procedure we create `entr_140` and `entr_200`.

Dataset	Avg. Tweet Length	Positive Ratio	# Tweets
complete data	15.74	0.5	2 500 000
random	15.739 ± 0.02	0.5 ± 0.001	140000
unif140	16.04	0.504	125997
unif200	16.02	0.503	179992
entr140	15.16	0.564	125994
entr200	15.16	0.57	179994

Table I
DATASET STATISTIC

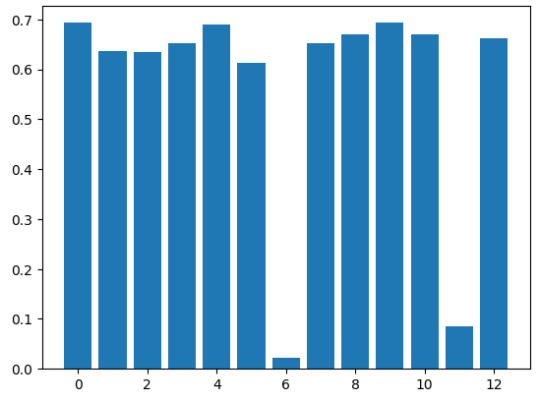

Figure 3. Entropy of the clusters

You can see in table I that the inner distribution of the datasets differ from the uniform. All the sets introduce a bias towards longer tweets and the positive tweets are oversampled.

Model

We start by describing the architecture and present the wide range of PLMs used in the experiments.

- **DistilBERT**:
  In our research, the initial model employed is the DistilBERT [20] pre-trained model. This compact and more efficient iteration of BERT contains 66 million parameters, compared to BERTbase's 110 million parameters. Remarkably, it preserves 97% of the language understanding capabilities of BERTbase while achieving a 60% increase in speed. The pre-training data encompasses the concatenation of the Toronto Book Corpus and English Wikipedia (same data as BERTbase).
- **RoBERTa**:
  We also utilized RoBERTabase a boosted version of the BERT transformer, that can match or exceed the performance of all of the post-BERT methods. The differences with BERTbase are simple, including extended duration, exclusion of the next sentence prediction objective, training on longer sequences, and dynamic masking pattern changes. They also presented a new dataset (CC-NEWS) for better control over training set size effects [21]. Roberta contains 125 millions parameters.
- **BERTweet**:
  The last model that we used is BERTweet [15]. BERTweet uses the same architecture as BERTbase [1]. BERTweet pre-training procedure is based on RoBERTa. Due to limitations in storage and size, our research focused exclusively on utilizing the base version, "*bertweet-base*." This variant was pretrained

on 850 million English Tweets and encompasses 135 million parameters.

Table II
STRUCTURAL COMPARISON OF DISTILBERT, ROBERTA, BERTWEET, AND GPT-2

Model	Parameters	Training Data	Max # Tokens
DistilBERT	67M	wikipedia, book-corpus	512
RoBERTa	125M	wikipedia, book-corpus	512
BERTweet	340M	Twitter-specific	130

Active Learning

Following Dagustpa [19], we propose a modified algorithm to reduce the amount of training predictions to do. The idea would be to assume that the PLM retains most of the task definition during the first epochs, when training on *simple* samples. The algorithm consists in dividing the number of train samples by 2 at each epochs, keeping only the samples for which the model is the most *uncertain* and keeping some randomness.

Let $\mathcal{X}$ be the set of all the tweets and $\mathcal{S}_k \subset \mathcal{X}$ be the set of selected tweets at iteration k. We chose to measure the *uncertainty* of a sample for a certain model $\mathcal{M}$ as :

$$H(x) = -\sum_{c \in \{0,1\}} \Big(\mathbf{P}(\mathcal{M}(x) = c) log(\mathbf{P}(\mathcal{M}(x) = c) \Big) \quad (4)$$

At iteration, say k, we compute the set of prediction $\mathcal{P}_k = \{\mathcal{M}(x), \forall x \in \mathcal{S}_k\}$. We then build S_{k+1} by merging the 25% of samples of S_k with the highest decision entropy ($|S_k|/4$ samples) and $|S_k|/4$ samples taken at random from the S_0.

We thus get :

$$|S_{k+1}| = \frac{|S_k|}{2}, \forall k \geq 0 \quad (5)$$

We stop this operation when $|S_k| \leq T$, for T a certain threshold. The runtime of this operation is strictly inferior to 2 eopchs. VII-A

IV. EXPERIMENTS

We assess and compare the results and performance of our pre-trained models, with and without our active learning methods, against robust baselines.

Model Selection

The initial step in our experiments involved selecting the best-performing pre-trained models from our chosen set. Subsequently, we fine-tuned these models to boost their effectiveness in sentiment analysis tasks. To achieve this, we employed the transformers library [22] to fine-tune BERTweet for our specific task, conducting 2 training epochs. The choice of a two-epoch training was motivated by the need for realistic training within a reasonable timeframe and resource constraints. Given the relatively small sample dataset (between 140,000 and 200,000 tweets), employing more epochs resulted in overfitting. For this model benchmarking step, we utilized AdamW with a fixed learning rate of 1.e-5 and a batch size of 32 (replicating the experimental setup detailed in [21] and [15]). We performed these training using a fix random seed and based our performance on the AI-Crowd score of the fine tuned model.

Hybrid Training Approach

After identifying the top-performing model within our resource constraints and obtaining its best scores through initial training on 200,000 Tweets, we conducted our primary experiment. Employing hybrid and sampling methods (inspired by Active Learning [23]), to enhance model performance without modifying the training sample size or model dimensions.

To achieve this, we trained our best model while still using a sample of 200,000 entries, leveraging our innovative sampling and clustering methods. To establish a strong baseline and ensure comparable results, we performed the training using a fixed random seed and the AI-Crowd score of the fine-tuned model.

In contrast to the initial simple general model selection step, we employed two types of optimizers to maximize the potential performance of our models: AdamW and Rectified Adam (RAdam) [24]. RAdam is a variant of Adam that introduces a term to rectify the variance of the adaptive learning rate. We incorporated this new optimizer in our benchmark, as the vanilla Adam algorithm could lead to suspicious/bad local optima and requires a learning rate warm-up stage to stabilize training. RAdam, meanwhile, is touted to be more robust and has the capability to automatically control the warm-up behavior.

Using these two optimizers along the two methods we designed we ended up with 8 configuration to test:

- Best model (x2: AdamW and RAdam)
- Best model with Entropy Sampling (x2: AdamW and RAdam)
- Best model with Clustered Dataset (x2: AdamW and RAdam)
- Best model with Entropy Sampling + Clustered Dataset (x2: AdamW and RAdam)

We conduct all these training sessions on a single Google Colab V100 GPU (16GB) with a fixed learning rate of 1.e-5 and an optimal weight decay of 0.001 for the optimizers.

V. RESULTS

The benchmarking of the models highlights the superiority to **BERTweet** in a similar and rigid test setup. As the results

Model	Test Accuracy	Sample Size	Optim.
BERTweet	0.902	200 000	AdamW
BERTweet Active Learning (#245873)	**0.905**	200 000	AdamW
BERTweet Aware Sampling + Active Learning	**0.905**	entr_200	AdamW
BERTweet (#247095)	**0.907**	200 000	RAdam
BERTweet Active Learning	0.902	200 000	RAdam
BERTweet Aware sampling	0.896	unif_200	RAdam
BERTweet Aware sampling + Active Learning	0.902	unif_200	RAdam
BERTweet Aware sampling	0.813	entr_200	RAdam
BERTweet Awaare Sampling + Active Learning	0.903	entr_200	RAdam

Table III
MODELS RESULTS

in table IV shows, distil-BERT yields, as expected, the poorest results. The difference in score between RoBERTa and BERTweet is also significant and highlights the difference between a general purpose PLM and a fine-tuned PLM.

Exploration of the capacity of **BERTweet** under a deeper set of experience yields the results available in table III. While the **active learning** methods don't yeild a significant improvement on the test accuracy, we reach similar results with a lighter training. With these training we only conduct 2 epochs and thus have **25% less training predictions**, and thus a similar speed up of the training overall.

Aware sampling performs on its own poorly. This may be due to a rigged clustering, leading to a dataset that doens't represent all the diversity of tweets. We see that combined with **Active Learning** the result are nevertheless close the best achieved result. **Active Learning** seem to be mitigating the initial biased distribution by subsampling the dataset properly.

Model	Test Accuracy	Sample Size	Optim.
Distil-BERT	0.862	140 000	AdamW
RoBERTa	0.870	140 000	AdamW
BERTweet	**0.901**	140 000	AdamW

Table IV
MODELS BENCHMARK

VI. CONCLUSION

We have presented the potential of Pretrained Language Models and Active Learning for tweets sentiment classification. We demonstrate the usefulness of Active Learning that achieves equal performance to standalone models' fine-tuning accuracy while concurrently mitigating its computational costs. We propose a novel tweet classification model: BERTweet with active learning. This model effectively presents an alternative innovation to reduce the training costs and time. However, considering the subpar results of aware sampling methods based on clustering similarity, further research in that direction could be a promising avenue.

We hope that other future work can help fostering the reduction of computational costs associated with the trainings while preserving data anonymity. One considerable direction is the exploration of the application of AL within a Federated Learning (FL) context, a ML setting showing promising performance in LLMs decentralization and data privacy [25], [26].

REFERENCES

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[2] J. Eisenstein, "What to do about bad language on the internet," in *Proceedings of the 2013 conference of the North American Chapter of the association for computational linguistics: Human language technologies*, pp. 359–369, 2013.

[3] L. Yue, W. Chen, X. Li, W. Zuo, and M. Yin, "A survey of sentiment analysis in social media," *Knowledge and Information Systems*, vol. 60, pp. 617–663, 2019.

[4] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, "On the dangers of stochastic parrots: Can language models be too big?," in *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pp. 610–623, 2021.

[5] L. Weidinger, J. Mellor, M. Rauh, C. Griffin, J. Uesato, P.-S. Huang, M. Cheng, M. Glaese, B. Balle, A. Kasirzadeh, *et al.*, "Ethical and social risks of harm from language models," *arXiv preprint arXiv:2112.04359*, 2021.

[6] D. M. Douglas, "Doxing: a conceptual analysis," *Ethics and information technology*, vol. 18, no. 3, pp. 199–210, 2016.

[7] Y. Karimi, A. Squicciarini, and S. Wilson, "Automated detection of doxing on twitter," *Proceedings of the ACM on Human-Computer Interaction*, vol. 6, no. CSCW2, pp. 1–24, 2022.

[8] J. Isaak and M. J. Hanna, "User data privacy: Facebook, cambridge analytica, and privacy protection," *Computer*, vol. 51, no. 8, pp. 56–59, 2018.

[9] T. M. Massaro and H. Norton, "Siri-ously? free speech rights and artificial intelligence," *Nw. UL Rev.*, vol. 110, p. 1169, 2015.

[10] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in nlp," *arXiv preprint arXiv:1906.02243*, 2019.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[12] N. F. Da Silva, E. R. Hruschka, and E. R. Hruschka Jr, "Tweet sentiment analysis with classifier ensembles," *Decision support systems*, vol. 66, pp. 170–179, 2014.

[13] C. Hutto and E. Gilbert, "Vader: A parsimonious rule-based model for sentiment analysis of social media text," in *Proceedings of the international AAAI conference on web and social media*, vol. 8, pp. 216–225, 2014.

[14] M. Cliche, "Bb_twtr at semeval-2017 task 4: Twitter sentiment analysis with cnns and lstms," *arXiv preprint arXiv:1704.06125*, 2017.

[15] D. Q. Nguyen, T. Vu, and A. T. Nguyen, "Bertweet: A pre-trained language model for english tweets," *arXiv preprint arXiv:2005.10200*, 2020.

[16] F. Barbieri, L. E. Anke, and J. Camacho-Collados, "Xlm-t: Multilingual language models in twitter for sentiment analysis and beyond," *arXiv preprint arXiv:2104.12250*, 2021.

[17] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," 1994.

[18] F. Cicalese, S. Filho, E. Laber, and M. Molinaro, "Teaching with limited information on the learner's behaviour," in *Proceedings of the 37th International Conference on Machine Learning* (H. D. III and A. Singh, eds.), vol. 119 of *Proceedings of Machine Learning Research*, pp. 2016–2026, PMLR, 13–18 Jul 2020.

[19] S. Dasgupta, D. Hsu, S. Poulis, and X. Zhu, "Teaching a black-box learner," in *Proceedings of the 36th International Conference on Machine Learning* (K. Chaudhuri and R. Salakhutdinov, eds.), vol. 97 of *Proceedings of Machine Learning Research*, pp. 1547–1555, PMLR, 09–15 Jun 2019.

[20] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," 2020.

[21] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019.

[22] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, *et al.*, "Huggingface's transformers: State-of-the-art natural language processing," *arXiv preprint arXiv:1910.03771*, 2019.

[23] K. Margatina, T. Schick, N. Aletras, and J. Dwivedi-Yu, "Active learning principles for in-context learning with large language models," 2023.

[24] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," *arXiv preprint arXiv:1908.03265*, 2019.

[25] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings, *et al.*, "Advances and open problems in federated learning," *Foundations and Trends® in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[26] A. Hilmkil, S. Callh, M. Barbieri, L. R. Sütfeld, E. L. Zec, and O. Mogren, "Scaling federated learning for fine-tuning of large language models," in *International Conference on Applications of Natural Language to Information Systems*, pp. 15–23, Springer, 2021.

VII. Appendix

A. *Proof for Active Learning*

The number of prediction to do is $|\mathcal{S}_0|$ at iteration 0, $|\mathcal{S}_1| = |\mathcal{S}_0|/2$ at iteration 1, and so on until having to do $|\mathcal{S}_k| = |\mathcal{S}_0|/2^k$ at iteration k.

We reached the bottom of recursion when :

$$|\mathcal{S}_k| = |\mathcal{S}_0|/2^k = T$$

$$|\mathcal{S}_\prime|/2^k = T \iff k = log_2(|\mathcal{S}_0|/T)$$

We'll do $N = \mathcal{O}(log(|\mathcal{S}_0|/T))$ iterations. with overall number of prediction $\mathcal{R}$ of

$$\mathcal{R} = |\mathcal{S}_0| \sum_{i=1}^{N} 2^{-i} \ < |\mathcal{S}_0| \sum_{i=1}^{+\infty} 2^{-i} = 2|\mathcal{S}_0|$$

The runtime is clearly upperbounded by 2 full epochs on $\mathcal{S}_0$.