Active Few-Shot Learning for Text Classification Tasks

Anonymous ACL submission

Abstract

The rise of Large Language Models (LLM) in the field of natural language processing has created opportunities to utilize the power of Few-Shot Learning (FSL) methods. These methods are able to achieve acceptable performance even when working with limited training data. The goal of FSL is to effectively utilize a small number of annotated samples in the learning process. However, the performance of FSL suffers when unsuitable support samples are chosen. This problem arises due to the heavy reliance on a limited number of support samples, which hampers consistent performance improvement even with the addition of more support samples. To address this challenge, we propose an active learning-based instance selection mechanism that identifies effective support instances from the unlabeled pool and is able to work with different LLMs like BART and FLAN-T5. We have conducted several experiments on three different classification tasks. The experimental results show that our proposed method consistently improves performance for different few-shot tasks.

1 Introduction

Deep learning systems have shown great performance when given enough labeled data, yet they struggle to learn from a small amount of labeled data (Sun et al., 2019). However, a large corpus of labeled data is costly and time-consuming to make for many real-world applications, and this often hinders the building of a supervised classifier for a new domain or application (Zhu et al., 2009). In Few-Shot Learning (FSL), the deep learning system has a small supply of data with supervised information for the target tasks (Wang et al., 2020). FSL seeks to grasp new concepts from limited labeled examples and build effective systems for a broader range of applications (Sun et al., 2019). On the other hand, with recent advances in large language models (LLM), the capabilities of FSL can be utilized more than before and these methods are able to reach acceptable performance even when using a small amount of training data. Several techniques have been proposed that are based on this concept (Gao et al., 2021; Chen et al., 2021; Karimi Mahabadi et al., 2022; Lin et al., 2022).

In the majority of FSL methods, the samples are typically selected randomly and variations in the quality of the samples can have a significant impact on the model's performance. In some scenarios, adding un- or less-informative samples can even decrease the accuracy of the fine-tuned model or may result in a large variance in the model's performance (Zhang et al., 2020; Schick and Schütze, 2021b).

Considering these challenges, we proposed a new Active Learning (AL)-based Few-Shot (FS) sample selection method that chooses the most informative unlabeled samples in order to enhance classification performance without increasing annotation costs. In accordance with successful AL algorithms (Settles, 2009), our algorithm selects instances based on different methods (i.e., entropy and clustering) to consider uncertainty, diversity, and representativeness in sample selection. It should be noted that in our proposed FSL approach, the chosen samples will be used for fine-tuning the LLMs and this method can be easily integrated with existing LLMs.

To assess the effectiveness of our methodology, we conducted an extensive series of experiments across three distinct classification tasks: specifically, we tackled type, polarity, and intensity classification problems using the Multi-Perspective Question Answering (MPQA) dataset. It is worth noting that even though these tasks were chosen from the same dataset, they belong to different categories. Our investigation also involved multiple language models, such as BART (Lewis et al., 2019) and FLAN-T5 (Chung et al., 2022). However, the proposed approaches in this paper can be

used on any LLM that provide access to the final hidden states of its encoder and the probability of each label's occurrence in the model's output.

Our contributions can be summarized as follows: 1) We introduce an AL-based sample selection scenario by combining uncertainty and representativeness measures for FS classification problems, which achieves state-of-the-art performance when paired with various recent FSL classification algorithms. 2) To the best of our knowledge, this is the first active FSL for text classification tasks.

2 Related Work

In previous studies, the FS scenario has been simulated by randomly sampling a subset from the complete training data (Chen et al., 2020; Schick and Schütze, 2021a; Gao et al., 2021; Chen et al., 2021; Lin et al., 2022). Among different FSL methods in Natural Language Processing (NLP), there are few methods that have paid attention to the sample selection strategies.

Some recent studies in the field of image processing have demonstrated the effectiveness of incorporating AL strategies in the context of FSL (Pezeshkpour et al., 2020; Boney et al., 2019; Li et al., 2021; Shin et al., 2022).

The study conducted by Chang et al. (2021) stands as the sole work that specifically addresses sample selection in NLP. Their research focuses on training instance selection in the context of FS neural text generation and using it in three different tasks with BART. Their approach is motivated by the idea that FS training instances should exhibit diversity and representativeness. To achieve this, they utilized K-Means clustering for choosing data points closer to the center of clusters as important.

3 Dataset

We use the MPQA Opinion Corpus 2.0 dataset that is annotated at the word or phrase level to extract the following features of attitudes expressed in the text: **type**, **polarity**, and **intensity**. To elaborate, a sentence may contain expressions that reflect different private states with various attitudes. These attitudes can belong to different types, and each type can express negative or positive opinions (polarity) toward targets with varying degrees of strength (intensity) (Wiebe et al., 2005; Wilson, 2008).

The original MPQA annotation scheme comprises 6 types of attitudes. We remove the *other* and *speculation* types in our experiments as these types of attitudes do not hold a polarity. That leaves us with a 4-class classification task for the type. Furthermore, an expression in a sentence may have zero to four labels as attitude types based on the expression itself and the sentence that contains the expression. This leads the type identifier task to be a multi-label classification task. Subsequently, we identify polarity and intensity using the attitude type, the expression that holds the attitude, and the expression's container sentence as the input. This input can only have one specific polarity and one intensity, which makes these tasks binary and 5-class multi-class classification tasks, respectively.

Task	Input		Output
	Attitude Type	Sentence	
T	-	The new US policy **deserves to be closely analyzed and monitored**.	arguing sentiment
P	intention	Canada is among the countries that have **pledged to** ratify the accord.	positive
I	sentiment	There is **a deep faith** here, however, in the power of democracy.	high

Table 1: The examples for Type (T), Polarity (P), and Intensity (I) tasks. The expressions within the sentences are in bold.

An example for each task is available in Table 1, and all labels and their distribution are as follows: **type:** *agreement* (×284), *arguing* (×2,466), *intention* (×420), and *sentiment* (×3,862) **polarity:** *negative* (×3,200) and *positive* (×3,832); and **intensity:** *low* (×658), *low-medium* (×1,262), *medium* (×2,615), *medium-high* (×1,258), and *high* (×1,239).

4 Embedding and Sampling Methods

Whether we are using a simple FS instance selection or an active one, we are going to have one or more iterations of selecting some samples and fine-tuning the model. We name the experiments with one iteration 'non-iterative' and name the rest 'iterative'. In each iteration, we retrieve embeddings from a certain source using an **embedding method**. Then, we perform some processing on the obtained embeddings, and choose some samples to be added to our support set using a **sampling method**. These methods are explained in this section.

4.1 Embedding Methods

We retrieve the embeddings in two different ways. The first way is to use the last hidden states of BART's or T5's encoder which we will call **En** for short. The other way is what we call scores or **Sc**. It uses the logits of the model and calculates the probability of the occurrence of each label so that each sample will end up with a 2 to 5-dimensional vector

(depending on the number of classes of the specific task) as its embedding. In both cases, we use a pre-trained model without any fine-tuning during the first iteration and use the fine-tuned model of the previous iteration during the subsequent iterations.

To calculate the scores, we need to compute the probability $P_m^{<t>}[n]$ (Equation 1) which represents how likely the token at position t in sample m's logits belongs to the n^{th} class out of all the N classes. $Logits_m^{<t>}[i]$ indicates the model's logit of the i^{th} word of the vocabulary at the position t for the sample m. During this procedure, we need our classes to be represented by a single token, and we will disregard all the other tokens in the vocabulary that are not included in the task's classes. For this matter, we use the dictionary $ClassId(i)$ to find the index of the i^{th} class in the vocabulary. Afterward, we get the score $Score_m[n]$ (Equation 2) by taking the maximum probability of the n^{th} class over all the output tokens (of size T) for the sample m. This is especially important for multi-label tasks like the type task that may have more than one token in the output to delineate more number of labels.

$$P_m^{<t>}[n] = \frac{e^{Logits_m^{<t>}[ClassId(n)]}}{\sum_{i=1}^{N} e^{Logits_m^{<t>}[ClassId(i)]}} \quad (1)$$

$$Score_m[n] = \max_{1 \le i \le T} (P_m^{<i>}[n]) \quad (2)$$

4.2 Sampling Methods

Within each iteration, M instances need to be sampled from the training set and added to the support set of size K. More precisely, these instances are sampled from the (simulated) unlabeled training set by considering the inputs and their corresponding embeddings. Only after choosing the samples, can we look at the labels of the M instances and use them in the fine-tuning process. M is a small number and is considered to designate the whole selection size, unlike typical FSL classification tasks that select M samples for each class (Ren et al., 2018; Chen et al., 2019; Wang et al., 2023), since we do not have access to those classes in our definition of the problem.

The sampling methods that we use in this paper are as follows: 1) **Random**: with this method, we simply sample M instances randomly without replacement. 2) **Representative (Rep)**: This method gets help from the embeddings we retrieved in our desired embedding method to cluster the unlabeled data into M groups using the *K-Means* algorithm. Then, inside each cluster, we sample the data point that is the closest (euclidean distance) to the cluster centroid. 3) **Uncertainty (Un)**: It can only benefit from the *Sc* embeddings to select the M samples about which the model has the most doubts. We will be using *entropy* (Shannon, 1948; Settles, 2009) as our uncertainty measure throughout this paper. 4) **Uncertainty Representative (UnRep)**: Using this technique, we first choose the $\alpha \times M$ most uncertain samples based on the *Sc* embeddings. Thereafter, we will do a representative sampling based on the *En* embeddings only on these selected data points in order to sample the final M unlabeled data. 5) **Cluster Uncertainty (ClUn)**: This strategy, at first, splits the data into M clusters considering the given embeddings using the *K-Means* algorithm. It will then pick the data point that the model has the least confidence about inside each cluster by looking at their *Sc* embeddings.

All of these methods can be used during the second iteration onwards, but only the ones that do not involve uncertainty (Random and Rep) can be used within the first iteration and/or non-iterative approaches since there's no previous step for the model to learn enough about the task and decide whether it has doubts about the data.

5 Experiments

To get better intuition about the tasks, we first calculate the majority baselines, which are the baselines we expect to beat. Additionally, we fine-tune the models using the whole training set as our support set (K = full training set size). These results represent a sort of top-line, which we do not expect to beat in our FS experiments. In addition, we fine-tune all pre-trained models with $K \in \{10, 20, 50, 100\}$ using random sampling, representative sampling, and our proposed iterative approaches. In iterative approaches, within each iteration, we sample $M = 10$ new data points to be added to our support set and show the results when we have fine-tuned the model using support sets of size $K \in \{10, 20, 50, 100\}$. We assign α, in Section 4.2, the value of 10.

Table 2 shows the outcomes of these experiments. The name of each model starts with the employed pre-trained model's name. It then continues with the sampling method we have used to choose new samples. If we have used an iterative approach, this part shows the sampling method during the first iteration, and in that case, we will have

Model Name	Type					Polarity					Intensity				
	10	20	50	100	Full (4,248)	10	20	50	100	Full (4,505)	10	20	50	100	Full (4,505)
Majority Baseline	-	-	-	-	56.6	-	-	-	-	54.8	-	-	-	-	37.2
Random Sampling															
BART-Random	**57.2**	56.9	59.3	63.5	80.3	72.8	77.7	81.9	87.2	92.5	36.0	36.2	36.6	35.8	47.0
FLAN-T5-Random	55.6	59.3	64.5	67.1	**80.7**	74.4	80.1	84.3	88.3	**94.2**	31.0	32.7	36.0	36.1	**50.0**
Representative Sampling															
BART-Rep(En)	56.2	57.0	59.2	63.9	-	71.4	77.5	82.5	86.3	-	**37.0**	35.2	37.0	36.9	-
FLAN-T5-Rep(En)	52.0	**63.3**	64.5	67.9	-	**78.3**	79.5	85.8	88.8	-	34.3	35.8	36.5	35.9	-
Iterative Approaches															
FLAN-T5-Rep(En)-Un	54.6	59.9	64.4	66.9	-	**78.3**	80.3	**88.2**	**91.0**	-	34.5	36.3	37.1	38.2	-
FLAN-T5-Rep(En)-Rep(Sc)	54.6	61.0	65.5	68.6	-	**78.3**	81.7	87.5	90.8	-	34.5	35.1	37.0	37.8	-
FLAN-T5-Rep(En)-Rep(En)	54.6	60.9	64.8	**68.8**	-	**78.3**	80.4	85.4	87.7	-	34.5	35.4	37.1	38.0	-
FLAN-T5-Rep(En)-UnRep	54.6	59.8	63.2	67.8	-	**78.3**	81.7	86.8	90.5	-	34.5	36.2	37.3	38.2	-
FLAN-T5-Rep(En)-ClUn(Sc)	54.6	60.4	**65.7**	68.6	-	**78.3**	**82.1**	87.7	90.6	-	34.5	**36.6**	36.4	37.7	-
FLAN-T5-Rep(En)-ClUn(En)	54.6	58.9	65.1	68.5	-	**78.3**	79.5	86.8	90.9	-	34.5	35.5	**37.5**	**38.8**	-

Table 2: The results for type, polarity, and intensity tasks. The sub-columns denote K (i.e., support set size).

another sampling method that denotes the method we have utilized during the second iteration onwards at the end of the name. Whenever the referred sampling method can make use of any of the embedding methods, we specify the used method inside parentheses. All results in this paper are reported in micro-F1 (%). Each FSL experiment was run with **10 different seeds** in the sampling phase, and the average of the F1-scores is reported. All the utilized pre-trained models are the base variants.

6 Discussion

Table 2 elucidates a huge difference in the results (especially when fine-tuning on the full dataset) that emerges from the differing definitions of the tasks, even though they are on the same dataset. These tasks cover binary (polarity), multi-class (intensity), and multi-label (type) classification problems. The type task has an imbalanced set of independent labels, yet the polarity task does not. The intensity task has more labels than the others that are mostly semantically close which makes the problem much harder. So, we believe that our chosen tasks are diverse and representative of a wide range of classification tasks.

This table further shows that the FLAN-T5-based models work better than the BART-based models in most cases. This is why we focus on fine-tuning our iterative experiments on FLAN-T5 pre-trained model. These results indicate that simple representative sampling is more effective than random sampling, even if we use it in a non-iterative setup. Nevertheless, the iterative approaches when $K \geq 20$ tend to work even better than most of the non-iterative approaches. They successfully manage to achieve the highest scores for all tasks when $K \in \{50, 100\}$ by a notable margin, especially for the polarity task. The intensity task also succeeds in outperforming the majority baseline in FSL experiments only when using the iterative approaches. Although the three tasks yield distinct results, the iterative approach 'FLAN-T5-Rep(En)-ClUn(En)' usually outperforms the random and non-iterative approaches, and it does so consistently for $K \in \{50, 100\}$. Figure 1 captures the contrast between the non-iterative FLAN-T5-based models and the best performer model at $K = 100$ in greater detail.

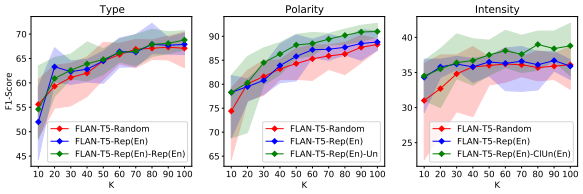

Figure 1: The F1 of all the tasks with steps of 10.

In addition, we have calculated the standard deviation, which varies between 0 and 7.4; while the standard deviation tends to decrease as the K increases, there are exceptions, and no strong pattern emerges. We provide the complete results with standard deviations alongside additional experimental results in Tables 3 and 4 of the appendix section.

7 Conclusion and Future Work

We propose a novel method for sampling data to be used in an FS setting with AL, while many others tend to sample data randomly. We show how using different embedding and sampling methods helps us achieve better results in classification tasks by choosing and labeling the most informative unlabeled samples that may represent the variety of data or that the model has the most doubts about. These methods unleash their full potential when used iteratively, using the fine-tuned model from the previous iterations. Future work will expand on new embedding and sampling methods in classification tasks as well as other types of NLP tasks.

Limitations

In the current study, we have centered our attention on English, using the MPQA Opinion Corpus 2.0 which is monolingual. In the future, we can focus on other natural languages and alternative datasets, but given the absence of corpora which are as detailed as MPQA for other languages, this may turn out to be difficult. Furthermore, our proposed methods are unable to be directly used in non-classification or non-NLP tasks and they need some modifications to be applied to these types of tasks. These experiments also require a lot of computational resources like the other AL approaches, since we have to iteratively run the same experiment 10 times with an incrementally augmented support set.

Ethics Statement

Our current study is a fundamental research work in the field of natural language processing and computational linguistics. There are many applications considered for these fields of research. For instance, understanding users' tweets on Twitter, e-commerce applications, and question answering. Although many research projects have been done in these fields, and a large number of them accomplished remarkable results, we do not explicitly recommend using these systems standalone. The reason is that there are open issues about the robustness and fairness of these systems. Hence, we see a need for human experts in interpreting the results. From our point of view, there are no ethical concerns about the platforms, technologies, tools, and algorithms used or proposed in this study. We should also note that the dataset, language models, tools, and libraries that we have utilized in this work are all publicly available.

References

Rinu Boney, Alexander Ilin, et al. 2019. Active one-shot learning with prototypical networks. In *ESANN*.

Ernie Chang, Xiaoyu Shen, Hui-Syuan Yeh, and Vera Demberg. 2021. On training instance selection for few-shot neural text generation. *arXiv preprint arXiv:2107.03176*.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. 2019. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*.

Yiming Chen, Yan Zhang, Chen Zhang, Grandee Lee, Ran Cheng, and Haizhou Li. 2021. Revisiting self-training for few-shot learning of language model. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9125–9135, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhiyu Chen, Harini Eavani, Wenhu Chen, Yinyin Liu, and William Yang Wang. 2020. Few-shot NLG with pre-trained language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 183–190, Online. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Rabeeh Karimi Mahabadi, Luke Zettlemoyer, James Henderson, Lambert Mathias, Marzieh Saeidi, Veselin Stoyanov, and Majid Yazdani. 2022. Prompt-free and efficient few-shot learning with language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3638–3652, Dublin, Ireland. Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Xiaorun Li, Zeyu Cao, Liaoying Zhao, and Jianfeng Jiang. 2021. Alpn: Active-learning-based prototypical network for few-shot hyperspectral imagery classification. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, et al. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052.

Pouya Pezeshkpour, Zhengli Zhao, and Sameer Singh. 2020. On the utility of active instance selection for few-shot learning. *NeurIPS HAMLETS*.

Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo

Larochelle, and Richard S Zemel. 2018. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*.

Timo Schick and Hinrich Schütze. 2021a. Few-shot text generation with natural language instructions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 390–402.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. pages 2339–2352.

Burr Settles. 2009. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Claude E Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.

Junsup Shin, Youngwook Kang, Seungjin Jung, and Jongwon Choi. 2022. Active instance selection for few-shot classification. *IEEE Access*.

Qianru Sun, Yaoyao Liu, Tat-Seng Chua, and Bernt Schiele. 2019. Meta-transfer learning for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xixi Wang, Xiao Wang, Bo Jiang, and Bin Luo. 2023. Few-shot learning meets transformer: Unified query-support transformers for few-shot classification. *IEEE Transactions on Circuits and Systems for Video Technology*.

Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni. 2020. Generalizing from a few examples: A survey on few-shot learning. *ACM Comput. Surv.*, 53(3).

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2):165–210.

Theresa Ann Wilson. 2008. *Fine-grained subjectivity and sentiment analysis: recognizing the intensity, polarity, and attitudes of private states*. University of Pittsburgh.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 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.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. 2020. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International Conference on Machine Learning*, pages 11328–11339. PMLR.

Jingbo Zhu, Huizhen Wang, Benjamin K Tsou, and Matthew Ma. 2009. Active learning with sampling by uncertainty and density for data annotations. *IEEE Transactions on audio, speech, and language processing*, 18(6):1323–1331.

A Additional Tables

Table 3 contains all the original FSL results in the paper with their standard deviations. This table further shows that using random sampling during the first iteration mostly fails to achieve as good results as representative sampling when K grows.

Table 4 encompasses the results of additional experiments for $K \in \{5, 10, 25, 50\}$. This time, we use $M = 5$ in the iterative approaches. Even though, in our problem context, K represents the overall size of the support set, distinct from the conventional few-shot learning classification tasks where K refers to the number of samples per class which makes equitable representation of all labels simultaneously within the support set much harder for a task like intensity, we can still see that the iterative approaches surpass the non-iterative approaches in most cases. Moreover, the iterative approach 'FLAN-T5-Rep(En)-ClUn(En)' still holds up to beat all the non-iterative approaches when $K = 50$.

B Implementation Details

Our models were implemented on PyTorch[1] neural network framework. Furthermore, we utilized the scikit-learn library[2], NumPy[3], and Matplotlib[4] packages. We used the BART and FLAN-T5 (all in *base* versions) models and their tokenizers from the *Hugging Face* Transformers library[5] (Wolf et al., 2020). Our models were executed on a single NVIDIA A100-SXM4-40GB GPU, DDR4 RAM, and dual AMD Rome 7742 CPUs (each with 2.25Ghz 64-Cores). The amount of GPU memory required for the experiments is at most 18GB. They also required a maximum of 16GB of RAM.

All results in this paper are reproducible by setting the random seeds to fixed numbers. In the

[1]https://pytorch.org/
[2]https://scikit-learn.org/stable/
[3]https://numpy.org/
[4]https://matplotlib.org/
[5]https://github.com/huggingface/transformers

Model Name	Type				Polarity				Intensity			
	10	20	50	100	10	20	50	100	10	20	50	100
Random Sampling												
BART-Random	57.2±2.4	56.9±1.3	59.3±1.3	63.5±2.4	72.8±4.2	77.7±3.4	81.9±3.4	87.2±1.6	36.0±1.9	36.2±2.2	36.6±1.2	35.8±1.7
FLAN-T5-Random	55.6±4.1	59.3±3.0	64.5±2.3	67.1±2.0	74.4±4.9	80.1±1.9	84.3±2.2	88.3±0.7	31.0±4.6	32.7±3.7	36.0±1.2	36.1±2.1
Representative Sampling												
BART-Rep(En)	56.2±0.7	57.0±1.2	59.2±2.3	63.9±2.8	71.4±0.0	77.5±2.8	82.5±2.7	86.3±2.0	**37.0**±0.0	35.2±2.3	37.0±0.6	36.9±0.7
FLAN-T5-Rep(En)	52.0±5.6	**63.3**±2.4	64.5±2.0	67.9±1.6	**78.3**±4.1	79.5±1.4	85.8±2.6	88.8±1.3	34.3±2.3	35.8±0.8	36.5±0.9	35.9±0.9
Iterative Approaches												
FLAN-T5-Random-Un	**57.3**±3.4	61.8±3.3	64.7±1.4	66.6±1.6	74.4±4.9	78.4±4.1	86.9±1.4	90.2±1.9	30.4±4.1	30.9±4.6	36.0±2.7	38.0±2.5
FLAN-T5-Random-Rep(En)	**57.3**±3.4	60.2±2.8	64.0±2.7	67.9±2.4	74.4±4.9	79.9±2.1	86.0±2.8	89.9±2.1	30.4±4.1	34.1±2.2	36.8±2.3	38.5±1.3
FLAN-T5-Rep(En)-Un	54.6±4.6	59.9±2.0	64.4±2.0	66.9±1.9	**78.3**±4.1	80.3±5.4	**88.2**±1.5	**91.0**±0.8	34.5±1.8	36.3±0.9	37.1±0.7	38.2±2.2
FLAN-T5-Rep(En)-Rep(Sc)	54.6±4.6	61.0±1.3	65.5±1.6	68.6±0.7	**78.3**±4.1	81.7±2.5	87.5±1.0	90.8±1.2	34.5±1.8	35.1±1.9	37.0±2.3	37.8±1.4
FLAN-T5-Rep(En)-Rep(En)	54.6±4.6	60.9±2.2	64.8±1.9	**68.8**±1.4	**78.3**±4.1	80.4±2.3	85.4±1.6	87.7±1.6	34.5±1.8	35.4±1.9	37.1±1.6	38.0±1.6
FLAN-T5-Rep(En)-UnRep	54.6±4.6	59.8±3.7	63.2±2.6	67.8±2.1	**78.3**±4.1	81.7±3.0	86.8±1.6	90.5±0.6	34.5±1.8	36.2±1.5	37.3±1.2	38.2±2.0
FLAN-T5-Rep(En)-ClUn(Sc)	54.6±4.6	60.4±2.4	**65.7**±2.4	68.6±1.6	**78.3**±4.1	**82.1**±4.3	87.7±1.5	90.6±1.1	34.5±1.8	**36.6**±0.6	36.4±1.4	37.7±1.6
FLAN-T5-Rep(En)-ClUn(En)	54.6±4.6	58.9±3.0	65.1±1.9	68.5±1.4	**78.3**±4.1	79.5±4.6	86.8±2.4	90.9±0.9	34.5±1.8	35.5±2.4	**37.5**±1.7	**38.8**±2.1

Table 3: All FSL results of type, polarity, and intensity tasks with their standard deviation when $M = 10$ (i.e., selection size) in iterative approaches. The sub-columns denote K (i.e., support set size).

Model Name	Type				Polarity				Intensity			
	5	10	25	50	5	10	25	50	5	10	25	50
Random Sampling												
BART-Random	55.0±3.8	57.2±2.4	58.0±2.1	59.3±1.3	68.0±9.0	72.8±4.2	76.8±3.3	81.9±3.4	32.7±6.2	36.0±1.9	36.1±1.5	36.6±1.2
FLAN-T5-Random	46.8±8.5	55.6±4.1	59.7±3.6	64.5±2.3	67.2±8.9	74.4±4.9	80.5±1.7	84.3±2.2	28.0±5.0	31.0±4.6	34.6±4.9	36.0±1.2
Representative Sampling												
BART-Rep(En)	52.7±0.0	56.2±0.7	56.9±1.6	59.2±2.3	62.9±15.3	71.4±0.0	78.9±3.4	82.5±2.7	**35.9**±1.7	**37.0**±0.0	36.4±1.2	37.0±0.6
FLAN-T5-Rep(En)	45.8±4.2	52.0±5.6	62.2±2.7	64.5±2.0	**72.1**±1.3	78.3±4.1	80.6±1.4	85.8±2.6	28.5±0.6	34.3±2.3	35.4±1.4	36.5±0.9
Iterative Approaches												
FLAN-T5-Rep(En)-Un	**59.3**±2.4	59.4±5.2	**63.5**±1.9	**65.7**±1.8	**72.1**±1.3	73.6±3.1	**84.7**±2.1	**88.6**±1.6	29.2±0.6	33.3±2.8	35.7±2.7	38.0±2.2
FLAN-T5-Rep(En)-Rep(Sc)	**59.3**±2.4	61.2±3.2	61.0±3.5	65.1±2.1	**72.1**±1.3	**81.2**±1.7	83.5±2.2	87.7±2.1	29.2±0.6	34.0±2.2	35.7±1.9	37.4±1.5
FLAN-T5-Rep(En)-Rep(En)	**59.3**±2.4	**62.2**±2.0	63.2±3.0	65.4±2.4	**72.1**±1.3	78.2±3.2	81.9±2.2	84.1±1.7	29.2±0.6	31.9±2.8	33.8±3.1	34.7±1.8
FLAN-T5-Rep(En)-UnRep	**59.3**±2.4	57.2±4.7	62.7±4.3	65.0±1.3	**72.1**±1.3	79.1±2.8	84.3±1.4	87.5±1.5	29.2±0.6	32.6±2.7	35.1±2.5	**38.9**±1.0
FLAN-T5-Rep(En)-ClUn(Sc)	**59.3**±2.4	61.8±3.3	**63.5**±2.8	65.0±2.4	**72.1**±1.3	80.3±2.3	84.0±1.8	88.5±1.8	29.2±0.6	33.7±1.5	**36.6**±1.0	37.6±1.5
FLAN-T5-Rep(En)-ClUn(En)	**59.3**±2.4	60.7±1.7	63.2±2.4	65.1±2.6	**72.1**±1.3	78.2±3.0	84.5±1.7	87.8±1.3	29.2±0.6	34.1±1.9	35.2±3.8	37.4±2.4

Table 4: All FSL results of type, polarity, and intensity tasks with their standard deviation when $M = 5$ (i.e., selection size) in iterative approaches. The sub-columns denote K (i.e., support set size).

present study, we utilized the MPQA opinion corpus. Hence, we did not use any human annotators.