Cluster & Tune: Boost Cold Start Performance in Text Classification

Eyal Shnarch,* Ariel Gera,* Alon Halfon,* Lena Dankin, Leshem Choshen, Ranit Aharonov, Noam Slonim

IBM Research
{eyals, alonhal, lenad, leshem.choshen, noams}@il.ibm.com,
ariel.gera1@ibm.com, ranitah1@gmail.com

Abstract

In real-world scenarios, a text classification task often begins with a *cold start*, when labeled data is scarce. In such cases, the common practice of fine-tuning pre-trained models, such as BERT, for a target classification task, is prone to produce poor performance. We suggest a method to boost the performance of such models by adding an intermediate unsupervised classification task, between the pre-training and fine-tuning phases. As such an intermediate task, we perform clustering and train the pre-trained model on predicting the cluster labels. We test this hypothesis on various data sets, and show that this additional classification phase can significantly improve performance, mainly for topical classification tasks, when the number of labeled instances available for fine-tuning is only a couple of dozen to a few hundred.

1 Introduction

The standard paradigm for text classification relies on supervised learning, where it is well known that the size and quality of the labeled data strongly impact the performance (Raffel et al., 2019). Hence, developing a text classifier in practice typically requires making the most of a relatively small set of annotated examples.

The emergence of transformer-based pre-trained language models such as BERT (Devlin et al., 2018) has reshaped the NLP landscape, leading to significant advances in the performance of most NLP tasks, text classification included (e.g., Nogueira and Cho, 2019; Ein-Dor et al., 2020). These models typically rely on pretraining with massive and heterogeneous corpora on a general Masked Language Modeling (*MLM*) task, i.e., predicting a word that is masked in the original text. Later on, the obtained model is fine-tuned to the actual task of interest, termed here the *target task*, using the labeled data available for this task. Thus, pretrained models serve as general sentence encoders which can be adapted to a variety of target tasks (Lacroix et al., 2019; Wang et al., 2020a).

Our work focuses on a challenging yet common scenario, where unlabeled data is available but labeled data is scarce. In many real-world scenarios, obtaining even a couple of hundred of labeled examples per class is challenging. Commonly, a target class has a relatively low prior in the examined data, making it a formidable goal to collect enough positive examples for it (Japkowicz and Stephen, 2002). Moreover, sometimes data cannot be labeled via crowd-annotation platforms due to its confidentiality (be it for data privacy reasons or for protecting intellectual property) or since the labeling task requires special expertise. On top of this, often the number of categories to be considered is relatively large, e.g., 50, thus making even a modest demand of 200 labeled examples per class a task of labeling 10K instances, which is inapplicable in many practical cases (for an extreme example, cf. Partalas et al., 2015).

In such limited real-world settings, fine-tuning a large pretrained model often yields far from optimal performance. To overcome this, one may take a gradual approach composed of various phases. One possibility is to further pretrain the model with the *self-supervised* MLM task over unlabeled data taken from the target task domain (Whang et al., 2019). Alternatively, one can train the pretrained model using a *supervised* intermediate task which is different in nature from the target-task, and for which labeled data is more readily available (Pruksachatkun et al., 2020; Wang et al., 2019a; Phang et al., 2018). Each of these steps is expected to provide a better starting point for the final fine-tuning phase, performed over the scarce labeled data available for the target task, aiming to end up with improved performance.

Following these lines, here we propose a strat-

*These authors contributed equally to this work.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 7639 - 7653

egy that exploits *unsupervised* text clustering as the intermediate task towards fine-tuning a pretrained model for text classification. Our work is inspired by the use of clustering to obtain labels in computer vision (Gidaris et al., 2018; Kolesnikov et al., 2019). Specifically, we use an efficient clustering technique, that relies on simple Bag Of Words (BOW) representations, to partition the unlabeled training data into relatively homogeneous clusters of text instances. Next, we treat these clusters as labeled data for an intermediate text classification task, and train the pre-trained model – with or without additional MLM pretraining – with respect to this multi-class problem, prior to the final fine-tuning over the actual target-task labels. Extensive experimental results demonstrate the practical value of this strategy on a variety of benchmark data. We further analyze the results to gain insights as to why and when this approach would be most valuable, and conclude that it is most prominently when the training data available for the target task is relatively small and the classification task is of a topical nature. Finally, we propose future directions.

We release code for reproducing our method.[1]

2 Intermediate Training using Unsupervised Clustering

A pre-trained model is typically developed in consecutive phases. Henceforth, we will refer to BERT as the canonical example of such models. First, the model is *pretrained* over massive general corpora with the MLM task.[2] We denote the obtained model simply as *BERT*. Second, BERT is *finetuned* in a supervised manner with the available labeled examples for the target task at hand. This standard flow is represented via Path-1 in Fig. 1.

An additional phase can be added between these two, referred to next as *intermediate training*, or inter-training in short. In this phase, the model is exposed to the corpus of the target task, or a corpus of the same domain, but still has no access to labeled examples for this task.

A common example of such an intermediate phase is to continue to intertrain BERT using the self-supervised MLM task over the corpus or the domain of interest, sometimes referred to as further or adaptive pre-training (e.g., Gururangan et al., 2020). This flow is represented via Path-2 in Fig. 1, and the resulting model is denoted $BERT_{IT:MLM}$, standing for Intermediate Task: MLM.

A key contribution of this paper is to propose a new type of intermediate task, which is designed to be aligned with a text classification target task, and is straightforward to use in practice. The underlying intuition is that inter-training the model over a related text classification task would be more beneficial compared to MLM inter-training, which focuses on different textual entities, namely predicting the identity of a single token.

Specifically, we suggest *unsupervised* clustering for generating pseudo-labels for inter-training. As the clustering partition presumably captures information about salient features in the corpus, feeding this information into the model could lead to representations that are better geared to perform the target task. These pseudo-labels can be viewed as weak labels, but importantly they are not tailored nor require a specific design per target task. Instead, we suggest generating pseudo-labels in a way independent of the target classification task. The respective flow is represented via Path-3 in Fig. 1. In this flow, we first cluster to partition the training data into n_c clusters. Next, we use the obtained partition as 'labeled' data in a text classification task, where the classes are defined via the n_c clusters, and intertrain BERT to predict the cluster label. In line with MLM, inter-training includes a classifier layer on top of BERT, which is discarded before the fine-tuning stage. The resulting inter-trained model is denoted $BERT_{IT:CLUST}$.

Finally, Path-4 in Fig. 1 represents a sequential composition of Paths 2 and 3. In this flow, we first intertrain BERT with the MLM task. Next, the obtained model is further intertrained to predict the n_c clusters, as in Path-3. The model resulting from this hybrid approach is denoted $BERT_{IT:MLM+CLUST}$.

Importantly, following Path-3 or Path-4 requires no additional labeled data, and involves an *a-priori* clustering of training instances that naturally gives rise to an alternative or an additional inter-training task. As we show in the following sections, despite its simplicity, this strategy provides a significant boost in performance, especially when labeled data for the final fine-tuning is in short supply.

[1] https://github.com/IBM/intermediate-training-using-clustering

[2] BERT was originally also pretrained over "next sentence prediction"; however, later works (Yang et al., 2019; Liu et al., 2019b) have questioned the contribution of this additional task and focused on MLM.

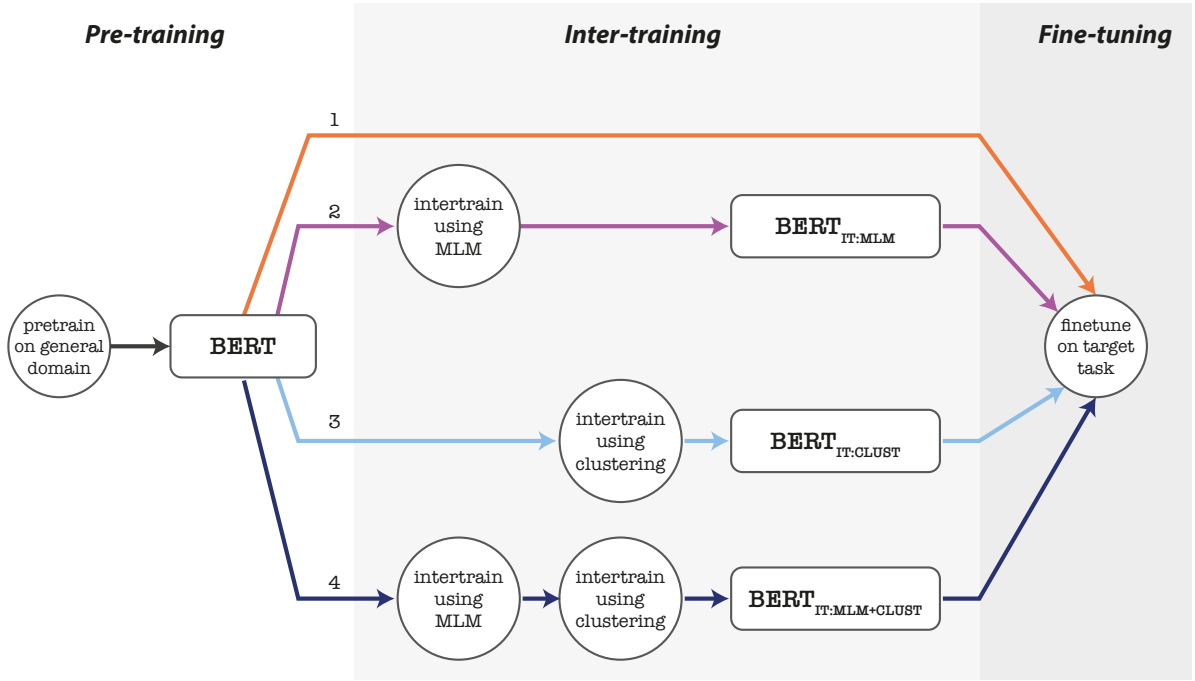

Figure 1: Phases of a pre-trained model (BERT in this figure) - circles are training steps which produce models, represented as rectangles. In the pre-training phase, only general corpora are available. The inter-training phase is exposed to target domain data, but not to its labeled instances. Those are only available at the fine-tuning phase.

3 Experiments

3.1 Tasks and Datasets

We evaluate over 6 topical datasets and 3 non-topical ones (see Table 1), which cover a variety of classification tasks and domains: Yahoo! Answers (Zhang et al., 2015), which separates answers and questions to types; DBpedia (Zhang et al., 2015, CC-BY-SA) which differentiates entity types by their Wikipedia articles; AG's News (Zhang et al., 2015) which categorize news articles; CFPB, which classifies consumer complaints; 20 newsgroups (Lang, 1995), which classifies 20 Usenet discussion groups; ISEAR (Shao et al., 2015, CC BY-NC-SA 3.0), which considers personal reports for emotion; SMS spam (Almeida et al., 2011), which identifies spam messages; Polarity (Pang and Lee, 2005), which includes sentiment analysis on movie reviews, and Subjectivity (Pang and Lee, 2004), which categorizes movie snippets as subjective or objective.

A topical dataset splits sentences by a high-level distinction related to what the sentence is about (e.g., sports vs. economics). Non-topical datasets look for finer stylistic distinctions that may depend on the way the sentence is written or on fine details rather than on the central meaning it discusses. It may also separate almost identical sentences; for example, "no" could distinguish between sentences with negative and positive sentiment.

When no split is provided we apply a $70\%/10\%/20\%$ train-dev-test split, respectively.[3] To reduce the computational cost over the larger datasets (DBpedia, AG's News, Yahoo! Answers and CFPB) we trim the train/test sets of these datasets to $15K/3K$ instances respectively, by randomly sampling from each set.[4] All runs and all methods use only the trimmed versions.

3.2 Experimental Setup

In our main set of experiments, we compare the performance of fine-tuning BERT-based models over a target task, for different settings of intermediate training. We consider four BERT-based settings, as described in Section 2 and in Figure 1. Two baselines – (i) BERT, without intermediate training, and (ii) $\text{BERT}_{\text{IT:MLM}}$ intertrained on MLM; and two settings that rely on clustering – (i) $\text{BERT}_{\text{IT:CLUST}}$, where predicting cluster labels is used for inter-training, and (ii) $\text{BERT}_{\text{IT:MLM+CLUST}}$, which combines the two intermediate tasks.

[3] The dev set is not being used by any method.

[4] We verified that relying on the full dataset provides no significant performance improvements to $\text{BERT}_{\text{IT:MLM}}$ and $\text{BERT}_{\text{IT:CLUST}}$. The results are omitted for brevity.

Training samples: For each setting, the final fine-tuning for the target task is performed, per dataset, for training budgets varying between 64 and 1024 labeled examples. For each data size x, the experiment is repeated 5 times; each repetition representing a different sampling of x labeled examples from the train set. The samplings of training examples are shared between all settings. That is, for a given dataset and train size the final training for all settings is done with respect to the same 5 samples of labeled examples.

Inter-training: Intermediate training, when done, was performed over the unlabeled train set for each dataset (ignoring instances' labels). We studied two implementations for the clustering task: K-means (Lloyd, 1982) and sequential Information Bottleneck (sIB) which is known to obtain better results in practice (Slonim et al., 2002) and in theory (Slonim et al., 2013). Based on initial experiments, and previous insights from works in the computer vision domain (Yan et al., 2020) we opted for a relatively large number of clusters, and rather than optimizing the number of clusters per dataset, set it to 50 for all cases.[5] K-means was run over GloVe (Pennington et al., 2014) representations following word stemming. We used a publicly available implementation of sIB[6] with its default configuration (i.e., 10 restarts and a maximum of 15 iterations for every single run). For sIB clustering, we used Bag of Words (BOW) representations on a stemmed text with the default vocabulary size (which is defined as the 10K most frequent words in the dataset). Our results indicate that inter-training with respect to sIB clusters consistently led to better results in the final performance on the target task, compared to inter-training with respect to the clusters obtained with K-means (see Section 5.1 for details). We also considered inter-training only on representative examples of clustering results – filtering a given amount of outlier examples – but obtained no significant gain (data not shown).

Note that the run time of the clustering algorithms is only a few seconds. The run time of the fine-tuning step of the inter-training task takes five and a half minutes for the largest train set (15K instances) on a Tesla V100-PCIE-16GB GPU.

[5]Setting the number of clusters to be equal to the number of classes resulted in inferior accuracy. In addition, one may not know how many classes truly exist in the data, so this parameter is not necessarily known in real-world applications.

[6]https://github.com/IBM/sib

	Train	Test	# classes
Yahoo! answers	15K	3K	10
DBpedia	15K	3K	14
CFPB	15K	3K	15
20 newsgroups	10.2K	7.5K	20
AG's news	15K	3K	4
ISEAR	5.4K	1.5K	7
SMS spam	3.9K	1.1K	2
Subjectivity	7K	2K	2
Polarity	7.5K	2.1K	2

Table 1: Dataset details. Topical datasets are at the top.

BERT hyper-parameters: The starting point of all settings is the $\text{BERT}_{\text{BASE}}$ model (110M parameters). BERT inter-training and fine-tuning runs were all performed using the Adam optimizer (Kingma and Ba, 2015) with a standard setting consisting of a learning rate of 3×10^{-5}, batch size 64, and maximal sequence length 128.

In a practical setting with a limited annotations budget one cannot assume that a labeled dev set is available, thus in all settings we did not use the dev set, and fine-tuning was arbitrarily set to be over 10 epochs, always selecting the last epoch. For inter-training over the clustering results we used a single epoch, for two reasons. First, loosely speaking, additional training over the clusters may drift the model too far towards learning the partition into clusters, which is an auxiliary task in our context, and not the real target task. Second, from the perspective of a practitioner, single epoch training is preferred since it is the least demanding in terms of run time. For $\text{BERT}_{\text{IT:MLM}}$ we used 30 epochs with a replication rate of 5, and followed the masking strategy from Devlin et al. (2018).[7]

Computational budget: Overall we report the results of 1440 BERT fine-tuning runs (4 experimental settings $\times$ 9 datasets $\times$ 8 labeling budgets $\times$ 5 repetitions). In addition, we performed 288 inter-training epochs over the full datasets (9 datasets $\times$ (30 $\text{BERT}_{\text{IT:MLM}}$ epochs + 1 $\text{BERT}_{\text{IT:CLUST}}$ epoch + 1 $\text{BERT}_{\text{IT:MLM+CLUST}}$ epoch)). In total, this would equate to about 60 hours on a single Tesla V100-PCIE-16GB GPU.

4 Results

Table 2 depicts the results over all datasets, focusing on the practical use case of a budget of 64

[7]In preliminary experiments we found this to be the best configuration for this baseline.

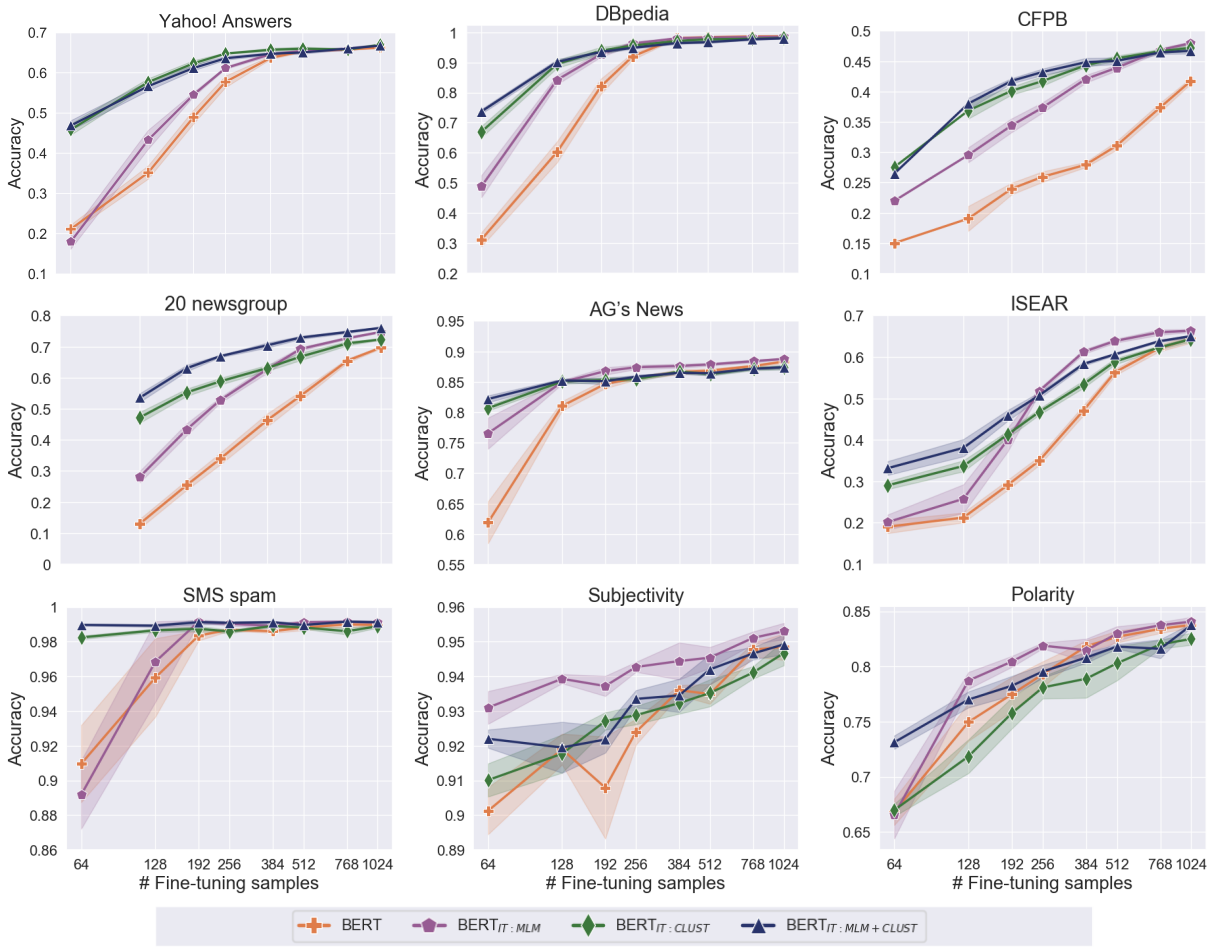

Figure 2: Classification accuracy (±SEM, standard error of the mean) on all datasets vs. the number of labeled samples used for fine-tuning (log scale). Each point is the average of 5 repetitions (for 20 newsgroups and a budget of 64, all 5 repetitions did not cover all classes and hence this data point is not presented).

samples for fine-tuning (128 for 20 newsgroup, see explanation in Fig. 2). As shown in the table, the performance gains of BERT$_{\text{IT:CLUST}}$ are mainly reflected in the 6 topical datasets. For these datasets, BERT$_{\text{IT:CLUST}}$ confers a significant benefit in accuracy (110% accuracy gain, 33% error reduction).

Figure 2 depicts the classification accuracy for the different settings for varying labeling budgets, using sIB for clustering-based inter-training. Over the topical datasets, BERT$_{\text{IT:CLUST}}$ and BERT$_{\text{IT:MLM+CLUST}}$ clearly outperform BERT and BERT$_{\text{IT:MLM}}$ in the small labeled data regime, where the gain is most prominent for the smallest labeled data examined – when only 64 labeled examples are available – and gradually diminishes as more labeled samples are added.

We performed paired t-tests to compare BERT$_{\text{IT:CLUST}}$ with BERT and BERT$_{\text{IT:MLM}}$, pooling together all datasets and repetitions for a given

Dataset	BERT accuracy	BERT$_{\text{IT:CLUST}}$ accuracy	Gain	Error reduction
Yahoo! Answers	21.2	45.9	117%	31%
DBpedia	31.2	67.0	115%	52%
CFPB	15.0	27.5	83%	15%
20 newsgroup	13.0	47.2	263%	39%
AG's News	61.9	80.7	30%	49%
ISEAR	19.0	29.0	53%	12%
avg. topical	26.9	49.6	**110%**	**33%**
SMS spam	91.0	98.2	8%	80%
Subjectivity	90.1	91.0	1%	9%
Polarity	66.8	67.0	0%	1%
avg. non-topical	82.6	85.4	**3%**	**30%**

Table 2: BERT$_{\text{IT:CLUST}}$ outperforms BERT in topical datasets. Comparing 64 samples, the smallest amount for fine-tuning. The accuracy gain and the error reduction (1-accuracy) are relative to BERT's accuracy/error.

Train size	64	128	192	256	384	512	>512
vs. BERT	1×10^{-6}	1×10^{-6}	6×10^{-7}	2×10^{-5}	2×10^{-3}	9×10^{-3}	–
vs. $\text{BERT}_{\text{IT:MLM}}$	8×10^{-5}	3×10^{-3}	4×10^{-2}	–	–	–	–

Table 3: Paired t-test p-values (after Bonferroni correction) of classification accuracy for $\text{BERT}_{\text{IT:CLUST}}$ compared to BERT and to $\text{BERT}_{\text{IT:MLM}}$ (insignificant results, $p \geq 0.05$, are denoted by –).

labeling budget. As can be seen in Tab. 3, the performance gain, over all datasets, of $\text{BERT}_{\text{IT:CLUST}}$ over BERT is statistically significant for a budget up to 512.

$\text{BERT}_{\text{IT:CLUST}}$ is not as successful in the 3 non-topical datasets (cf. Tab. 2 and Fig. 2). A possible reason for the lack of success of inter-training in these three datasets is that their classification task is different in nature than the tasks in the other six datasets. Identifying spam messages, determining whether a text is subjective or objective, or analyzing the sentiment (polarity) of texts, can be based on stylistic distinctions that may depend on the way the sentence is written rather than on the central topic it discusses. Inter-training over BOW clustering seems to be less beneficial when such considerations are needed. We further analyze this in Section 5.4. Nevertheless, it is safe to apply $\text{BERT}_{\text{IT:CLUST}}$ even in these datasets, as results are typically comparable to the baseline algorithms, neither better nor worse.

Both $\text{BERT}_{\text{IT:MLM}}$ and $\text{BERT}_{\text{IT:CLUST}}$ expose the model to the target corpus. The performance gains of $\text{BERT}_{\text{IT:CLUST}}$ over $\text{BERT}_{\text{IT:MLM}}$ suggest that inter-training on top of the clustering carries an additional benefit. In addition, these inter-training approaches are complementary - as seen in Fig. 2, $\text{BERT}_{\text{IT:MLM+CLUST}}$ outperforms both $\text{BERT}_{\text{IT:CLUST}}$ and $\text{BERT}_{\text{IT:MLM}}$ (at the cost of some added runtime).

Taken together, our results suggest that in topical datasets, where labeled data is scarce, the pseudo-labels generated via clustering can be leveraged to provide a better starting point for a pre-trained model towards its fine-tuning for the target task.

5 Analysis

5.1 Additional Clustering Techniques

In the literature (Slonim et al., 2002) and on our initial trials, sIB showed better clustering performance, and therefore was chosen over other clustering methods. Next, we analyze whether sIB is also the best fit for inter-training.

We compare (see App. C) sIB over BOW representation to two other clustering configurations; K-means over GloVe representations and Hartigan's K-means (Slonim et al., 2013) over GloVe. For most datasets, inter-training over the results of sIB over BOW representations achieved the best results.

5.2 Comparison to BOW-based methods

Our inter-training method relies on BOW-based clustering. Since knowledge of the input words is potentially quite powerful for some text classification tasks, we examine the performance of several BOW-based methods. We used the same training samples to train multinomial Naive Bayes (NB) and Support Vector Machine (SVM) classifiers, using either Bag of Words (BOW) or GloVe (Pennington et al., 2014) representations. For GloVe, a text is represented as the average GloVe embeddings of its tokens. This yielded four reference settings: NB_{BOW}, NB_{GloVe}, SVM_{BOW} and $\text{SVM}_{\text{GloVe}}$. Overall, all four methods were inferior to $\text{BERT}_{\text{IT:CLUST}}$, as shown in App. B. Thus, the success of our method cannot simply be attributed to the information in the BOW representations.

Next, we inspect the contribution of inter-training to BERT's sentence representations.

5.3 Effect on Sentence Embeddings

The embeddings after $\text{BERT}_{\text{IT:CLUST}}$ show potential as a better starting point for fine-tuning. Figure 3 depicts t-SNE (van der Maaten and Hinton, 2008) 2D visualizations of the output embeddings over the full train set of several datasets, comparing the [CLS] embeddings before and after inter-training.

Manifestly, for topical datasets, the $\text{BERT}_{\text{IT:CLUST}}$ embeddings, obtained after inter-training with respect to sIB clusters, induce a much clearer separation between the target classes, even though no labeled data was used to obtain this model. Moreover, and perhaps not surprisingly, the apparent visual separation resulting from inter-training is aligned with the performance gain obtained later on in the fine-tuning phase

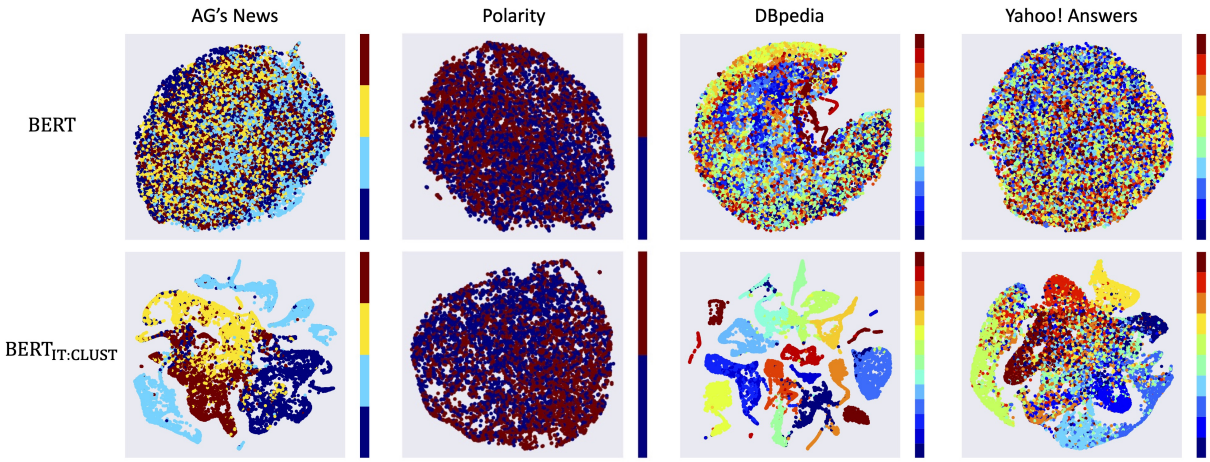

Figure 3: t-SNE visualizations of model embeddings over the train set, using BERT (top) vs. BERT$_{\text{IT:CLUST}}$ (bottom). The colors represent the gold labels for the target task (e.g., four classes in AG's News data set).

over the target task (as seen, for instance, in the visualizations of Polarity versus DBpedia data).

In addition to the qualitative results of the visualization, we pursue a more quantitative path. We assess whether examples of the same class are more closely represented after inter-training. Formally, given a set of instances' embeddings $e_1, \ldots, e_n$ and their corresponding class labels $l_1, \ldots, l_n \in \mathcal{L}$ we compute for each class $l \in \mathcal{L}$ a centroid c_l which is the average embedding of this class. We then compute the average Euclidean *Embeddings' Distance* (*ED*) from the corresponding centroids:[8]

$$ED(l, e) = \mathbb{E}_{i=0}^{n} \|e_i - c_i\|_2$$

As a sanity check, we apply a significance test to the ED statistic, confirming that representations of same-class examples are close to each other. Specifically, we apply a permutation test (Fisher, 1971), with 1000 repetitions, comparing the class labels to random labels. We find that EDs for both BERT and BERT$_{\text{IT:CLUST}}$ are significantly different from random ($p < 0.001$). This implies that both before and after inter-training, same-class representations are close. Next, we compare the representations before and after inter-training. We find that the randomly permuted EDs of BERT$_{\text{IT:CLUST}}$ are about 3 times larger than BERT's, despite similar norm values. This means that the post inter-training representations are more dispersed. Hence, to properly compare, we normalize ED by the average of the permuted EDs:

$$NED(l, e) = \frac{ED(l, e)}{\mathbb{E}_{\tau \in S_n} ED(\tau(l), e)}$$

Where $\tau \in S_n$ is a permutation out of S_n the set of all permutations.

Comparing the *Normalized Embeddings' Distance* (*NED*) before and after inter-training, we find that in all datasets the normalized distance is smaller after inter-training. In other words, BERT$_{\text{IT:CLUST}}$ brings same-class representations closer in comparison to BERT.

5.4 Are Clusters Indicative of Target Labels?

A natural explanation for the contribution of inter-training to BERT's performance is that the pseudo-labels, obtained via the clustering partition, are informative with regards to target task labels. To quantify this intuition, in Figure 4 we depict the Normalized Mutual Information (NMI) between sIB labels and the target task labels, calculated over the entire training set, versus the gain of using BERT$_{\text{IT:CLUST}}$, reflected as the reduction in classification error rate between BERT and BERT$_{\text{IT:CLUST}}$, at the extreme case of 64 fine-tuning samples. Evidently, in datasets where the NMI is around zero, BERT$_{\text{IT:CLUST}}$ does not confer a clear benefit; conversely, where the NMI is relatively high, the performance gains are pronounced as well. Notably, the three datasets with the lowest NMI are those for which inter-training was not beneficial, as discussed in Section 4.

Since the partition obtained via clustering is often informative for the target class labels, we examine whether it can be utilized directly, as opposed

[8] Macro average results were similar, we hence report only micro average results. Results with Cosine similarity were also similar, hence omitted.

to as pseudo-labels for BERT inter-training. To that end, we applied a simple heuristic. Given a labeling budget x, we divide it across clusters, relative to their size, while ensuring that at least one instance within each of the 50 clusters is labeled. We use the budget per cluster to reveal the labels of a random sample of examples in that cluster, and identify each cluster with its most dominant label. Next, given a new test example, we assign it with the label associated with its nearest cluster. Results (see App. B) showed that this rudimentary classifier is generally not on par with $\text{BERT}_{\text{IT:CLUST}}$, yet it can be surprisingly effective where the NMI is high and the labeling budget is small.

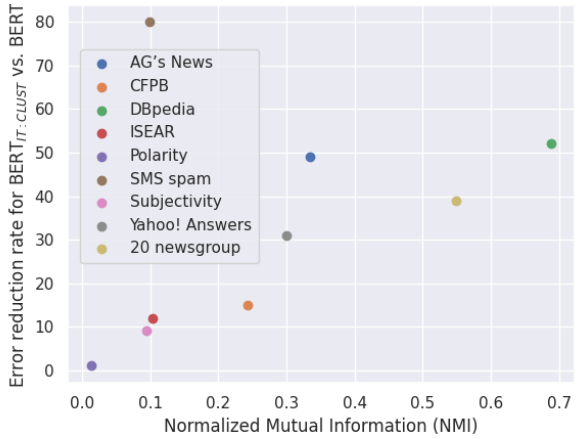

Figure 4: Improvement by $\text{BERT}_{\text{IT:CLUST}}$ vs Normalized Mutual Information (NMI) per dataset. x-axis: NMI between the cluster and class labels, over the train set. y-axis: The error reduction (percentage) by $\text{BERT}_{\text{IT:CLUST}}$, when fine-tuning over 64 samples.

6 Related Work

In our work, we transfer a pretrained model to a new domain with little data. Transfer learning studies how to transfer models across domains. It suggests methods such as pivoting (Ziser and Reichart, 2018), weak supervision (Shnarch et al., 2018), data augmentation (Anaby-Tavor et al., 2020) and adversarial transfer (Cao et al., 2018).

In Computer Vision, pretrained models are often learnt by image clustering (Caron et al., 2018). In NLP, however, clustering was mainly used for non-transfer scenarios. Ball (2019) relies on pretrained embeddings to cluster labeled and unlabeled data. Then, they fill the missing labels to augment the training data. Clustering itself was improved by combining small amounts of data (Torres and Vaca, 2019; Wang et al., 2016).

Pretrained models improved state-of-the-art in many downstream tasks (Nogueira and Cho, 2019; Ein-Dor et al., 2020) and they are especially needed and useful in low resource and limited labeled data settings (Lacroix et al., 2019; Wang et al., 2020a; Chau et al., 2020). There are many suggestions to improve such models, including larger models (Raffel et al., 2019), changes in the pretraining tasks and architecture (Yang et al., 2019), augmenting pre-training (Geva et al., 2020), or improving the transfer itself (Valipour et al., 2019; Wang et al., 2019b; Sun et al., 2019; Xu et al., 2020). Two findings on pretraining support our hypothesis on the intermediate task, namely that classification surpasses MLM. Some pretraining tasks are better than others (Lan et al., 2020; Raffel et al., 2019) and supervised classification as additional pre-training improves performance (Lv et al., 2020; Wang et al., 2019a; Pruksachatkun et al., 2020). All these works aim to improve the performance upon transfer, making it more suitable for any new domain. In contrast, we focus on improvement given the domain.

With a transferred model, one can further improve performance with domain-specific information. For example, utilizing metadata (Melamud et al., 2019), training on weakly-supervised data (Raisi and Huang, 2018; Meng et al., 2020) or multitasking on related tasks concurrently (Liu et al., 2019a). Given no domain-specific information, it was suggested to further pretrain on unlabeled data from the domain (Whang et al., 2019; Xu et al., 2019; Sung et al., 2019; Rietzler et al., 2020; Lee et al., 2020; Gururangan et al., 2020). This, however, is sometimes unhelpful or even hurts results (Pan, 2019).

Transferring a model and retraining with paucity of labels is often termed few-shot learning. Few shot learning is used for many language-related tasks such as named entity recognition (Wang et al., 2020b), relation classification (Hui et al., 2020), and parsing (Schuster et al., 2019). There have also been suggestions other than fine-tuning the model. Koch (2015) suggests ranking examples' similarity with Siamese networks. Vinyals et al. (2016) rely on memory and attention to find neighboring examples and Snell et al. (2017) search for prototypes to compare to. Ravi and Larochelle (2017) don't define in advance how to compare the examples. Instead, they meta-learn how to train the few shot learner. These works addressed the image classification domain, but they supply general methods

which are used, improved and adapted on language domains (Geng et al., 2019; Yu et al., 2018).

In conclusion, separate successful practices foreshadow our findings: Clustering drives pre-training on images; supervised classification aids pre-training; and training on unlabeled domain examples is helpful with MLM.

7 Conclusions

We presented a simple approach for improving pre-trained models for text classification. Specifically, we show that inter-training BERT over pseudo-labels generated via unsupervised clustering creates a better starting point for the final fine-tuning over the target task. Our analyses suggest that BERT can leverage these pseudo-labels, namely that there exists a beneficial interplay between the proposed inter-training and the later fine-tuning stage. Our results show that this approach yields a significant boost in accuracy, mainly over topical data and when labeled data is scarce. Note that the method does require the existence of an unlabeled corpus, in the order of several thousand examples.

We opted here for a practically oriented approach, which we do not claim to be optimal. Rather, the success of this approach suggests various directions for future work. In particular, several theoretical questions arise, such as what else determines the success of the approach in a given dataset; understanding the potential synergistic effect of BOW-based clustering for inter-training; could more suitable partitions be acquired by exploiting additional embedding space and/or more clustering techniques; co-training (Blum and Mitchell, 1998) methods, and more.

On the practical side, while in this work we fixed the inter-training to be over 50 clusters and for a single epoch, future work can improve performance by tuning such hyper-parameters. In addition, one may consider using the labeled data available for fine-tuning as anchors for the intermediate clustering step, which we have not explored here.

Another point to consider is the nature of the inter-training task. Here, we examined a multi-class setup where BERT is trained to predict one out of n_c cluster labels. Alternatively, one may consider a binary inter-training task, where BERT is trained to determine whether two samples are drawn from the same cluster or not.

Finally, the focus of the present work was on improving BERT performance for text classification. In principle, inter-training BERT over clustering results may be valuable for additional downstream target tasks, that are similar in spirit to standard text classification. Examples include Key-Point Analysis (Bar-Haim et al., 2020) and Textual Entailment (Dagan et al., 2013). The potential value of our approach in such cases is left for future work.

8 Ethical considerations

Any use of a language model for classification involves some risk of bias, which stems from the pre-training and training data used to construct the model. Here we aim to improve the language model representations by relying on clustering of data from the target domain. We have no reason to believe this process would introduce bias beyond the potential bias that can occur whenever fine-tuning a model, but this is a potential risk, as we did not verify this directly.

Acknowledgements

We thank Assaf Toledo for providing helpful advice on the clustering implementations.

References

Tiago A. Almeida, José María Gómez Hidalgo, and Akebo Yamakami. 2011. Contributions to the study of sms spam filtering: new collection and results. In *ACM Symposium on Document Engineering*, pages 259–262. ACM.

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, N. Tepper, and Naama Zwerdling. 2020. Do not have enough data? Deep learning to the rescue! In *AAAI*.

Michael Ball. 2019. RIPPED: Recursive intent propagation using pretrained embedding distances. Brown University.

Roy Bar-Haim, Lilach Eden, Roni Friedman, Yoav Kantor, Dan Lahav, and Noam Slonim. 2020. From arguments to key points: Towards automatic argument summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, COLT' 98, pages 92–100, New York, NY, USA. ACM.

Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, and Shengping Liu. 2018. Adversarial transfer learning for Chinese named entity recognition with self-attention mechanism. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 182–192, Brussels, Belgium. Association for Computational Linguistics.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. 2018. Deep clustering for unsupervised learning of visual features. In *ECCV*.

Ethan C. Chau, Lucy H. Lin, and Noah A. Smith. 2020. Parsing with multilingual BERT, a small corpus, and a small treebank. *arXiv:2009.14124*.

Ido Dagan, Dan Roth, Mark Sammons, and Fabio Massimo Zanzotto. 2013. Recognizing textual entailment: Models and applications. *Synthesis Lectures on Human Language Technologies*, 6(4):1–220.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*.

Liat Ein-Dor, Eyal Shnarch, Lena Dankin, Alon Halfon, Benjamin Sznajder, Ariel Gera, Carlos Alzate, Martin Gleize, Leshem Choshen, Yufang Hou, et al. 2020. Corpus wide argument mining-a working solution. In *AAAI*, pages 7683–7691.

Ronald. A. Fisher. 1971. Statistical methods for research workers. *Biometrics*, 27:1106.

Ruiying Geng, Binhua Li, Yongbin Li, Xiaodan Zhu, Ping Jian, and Jian Sun. 2019. Induction networks for few-shot text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3904–3913, Hong Kong, China. Association for Computational Linguistics.

Mor Geva, Ankit Gupta, and Jonathan Berant. 2020. Injecting numerical reasoning skills into language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 946–958, Online. Association for Computational Linguistics.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. 2018. Unsupervised representation learning by predicting image rotations. *arXiv:1803.07728*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv:2004.10964*.

Bei Hui, Liang Liu, Jia Chen, Xue Zhou, and Yuhui Nian. 2020. Few-shot relation classification by context attention-based prototypical networks with BERT. *EURASIP Journal on Wireless Communications and Networking*, 2020:1–17.

Nathalie Japkowicz and Shaju Stephen. 2002. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *arXiv:1412.6980*.

Gregory R. Koch. 2015. Siamese neural networks for one-shot image recognition.

Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. 2019. Revisiting self-supervised visual representation learning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1920–1929.

Ophélie Lacroix, Simon Flachs, and Anders Søgaard. 2019. Noisy channel for low resource grammatical error correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A lite BERT for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, D. Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*.

Xiaodong Liu, Pengcheng He, W. Chen, and Jianfeng Gao. 2019a. Multi-task deep neural networks for natural language understanding. *arXiv:1901.11504*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv:1907.11692*.

Stuart Lloyd. 1982. Least squares quantization in PCM. *IEEE transactions on information theory*, 28(2):129–137.

Shangwen Lv, Yuechen Wang, Daya Guo, Duyu Tang, N. Duan, F. Zhu, Ming Gong, Linjun Shou, Ryan Ma, Daxin Jiang, G. Cao, M. Zhou, and Songlin Hu. 2020. Pre-training text representations as meta learning. *arXiv:2004.05568*.

Oren Melamud, Mihaela Bornea, and Ken Barker. 2019. Combining unsupervised pre-training and annotator rationales to improve low-shot text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3884–3893, Hong Kong, China. Association for Computational Linguistics.

Yu Meng, Yunyi Zhang, Jiaxin Huang, Chenyan Xiong, Heng Ji, Chao Zhang, and Jiawei Han. 2020. Text classification using label names only: A language model self-training approach. *arXiv:2010.07245*.

Rodrigo Nogueira and Kyunghyun Cho. 2019. Passage re-ranking with BERT. *arXiv:1901.04085*.

Chenchen Pan. 2019. Analyzing BERT with pre-train on SQuAD 2.0. In *Stanford Archive*.

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 271–278, Barcelona, Spain.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan. Association for Computational Linguistics.

Ioannis Partalas, Aris Kosmopoulos, Nicolas Baskiotis, Thierry Artieres, George Paliouras, Eric Gaussier, Ion Androutsopoulos, Massih-Reza Amini, and Patrick Galinari. 2015. LSHTC: A benchmark for large-scale text classification. *arXiv:1503.08581*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Jason Phang, Thibault Févry, and Samuel R Bowman. 2018. Sentence encoders on stilts: Supplementary training on intermediate labeled-data tasks. *arXiv:1811.01088*.

Yada Pruksachatkun, Jason Phang, Haokun Liu, Phu Mon Htut, Xiaoyi Zhang, Richard Yuanzhe Pang, Clara Vania, Katharina Kann, and Samuel R Bowman. 2020. Intermediate-task transfer learning with pretrained models for natural language understanding: When and why does it work? *arXiv:2005.00628*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, W. Li, and Peter J. Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv:1910.10683*.

Elaheh Raisi and Bert Huang. 2018. Weakly supervised cyberbullying detection with participant-vocabulary consistency. *Social Network Analysis and Mining*, 8:1–17.

Sachin Ravi and Hugo Larochelle. 2017. Optimization as a model for few-shot learning. In *ICLR*.

Alexander Rietzler, Sebastian Stabinger, Paul Opitz, and Stefan Engl. 2020. Adapt or get left behind: Domain adaptation through BERT language model finetuning for aspect-target sentiment classification. *arXiv:1908.11860*.

Tal Schuster, Ori Ram, Regina Barzilay, and Amir Globerson. 2019. Cross-lingual alignment of contextual word embeddings, with applications to zero-shot dependency parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1599–1613, Minneapolis, Minnesota. Association for Computational Linguistics.

Bo Shao, Lorna Doucet, and David R. Caruso. 2015. Universality versus cultural specificity of three emotion domains: Some evidence based on the cascading model of emotional intelligence. *Journal of Cross-Cultural Psychology*, 46(2):229–251.

Eyal Shnarch, Carlos Alzate, Lena Dankin, Martin Gleize, Yufang Hou, Leshem Choshen, Ranit Aharonov, and Noam Slonim. 2018. Will it blend? blending weak and strong labeled data in a neural network for argumentation mining. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 599–605, Melbourne, Australia. Association for Computational Linguistics.

Noam Slonim, Ehud Aharoni, and Koby Crammer. 2013. Hartigan's K-means vs. Lloyd's K-means – is it time for a change? In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*.

Noam Slonim, Nir Friedman, and Naftali Tishby. 2002. Unsupervised document classification using sequential information maximization. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, page 129–136.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune BERT for text classification? *arXiv:1905.05583*.

Chul Sung, Tejas Dhamecha, Swarnadeep Saha, Tengfei Ma, Vinay Reddy, and Rishi Arora. 2019. Pre-training BERT on domain resources for short answer grading. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6071–6075, Hong Kong, China. Association for Computational Linguistics.

Johnny Torres and Carmen Vaca. 2019. Cl-aff deep semisupervised clustering. In *AffCon@AAAI*.

Mehrdad Valipour, En-Shiun Annie Lee, Jaime R. Jamacaro, and Carolina Bessega. 2019. Unsupervised transfer learning via BERT neuron selection. *arXiv:1912.05308*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Alex Wang, Jan Hula, Patrick Xia, Raghavendra Pappagari, R. Thomas McCoy, Roma Patel, Najoung Kim, Ian Tenney, Yinghui Huang, Katherin Yu, Shuning Jin, Berlin Chen, Benjamin Van Durme, Edouard Grave, Ellie Pavlick, and Samuel R. Bowman. 2019a. Can you tell me how to get past sesame street? sentence-level pretraining beyond language modeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4465–4476, Florence, Italy. Association for Computational Linguistics.

Ran Wang, Haibo Su, Chunye Wang, Kailin Ji, and Jupeng Ding. 2019b. To tune or not to tune? how about the best of both worlds? *arXiv:1907.05338*.

Sinong Wang, Madian Khabsa, and Hao Ma. 2020a. To pretrain or not to pretrain: Examining the benefits of pretrainng on resource rich tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2209–2213, Online. Association for Computational Linguistics.

Yaqing Wang, Subhabrata Mukherjee, H. Chu, Yuancheng Tu, Miaonan Wu, Jing Gao, and Ahmed Hassan Awadallah. 2020b. Adaptive self-training for few-shot neural sequence labeling. *arXiv:2010.03680*.

Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. 2016. Semi-supervised clustering for short text via deep representation learning. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 31–39.

Taesun Whang, Dongyub Lee, Chanhee Lee, Kisu Yang, Dongsuk Oh, and Heuiseok Lim. 2019. Domain adaptive training BERT for response selection. *arXiv:1908.04812*.

Hu Xu, Bing Liu, Lei Shu, and Philip Yu. 2019. BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2324–2335, Minneapolis, Minnesota. Association for Computational Linguistics.

Yige Xu, Xipeng Qiu, L. Zhou, and X. Huang. 2020. Improving BERT fine-tuning via self-ensemble and self-distillation. *arXiv:2002.10345*.

Xueting Yan, Ishan Misra, Abhinav Gupta, Deepti Ghadiyaram, and Dhruv Mahajan. 2020. Clusterfit: Improving generalization of visual representations. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6508–6517.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32.

Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. 2018. Diverse few-shot text classification with multiple metrics. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1206–1215, New Orleans, Louisiana. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, volume 28, pages 649–657.

Yftah Ziser and Roi Reichart. 2018. Pivot based language modeling for improved neural domain adaptation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1241–1251, New Orleans, Louisiana. Association for Computational Linguistics.

A Datasets

Links for downloading the datasets:

Polarity: http://www.cs.cornell.edu/people/pabo/movie-review-data/.

Subjectivity: http://www.cs.cornell.edu/people/pabo/movie-review-data/.

CFPB: https://www.consumerfinance.gov/data-research/consumer-complaints/.

20 newsgroups: http://qwone.com/~jason/20Newsgroups/
We used the version provided by scikit: https://scikit-learn.org/0.15/datasets/twenty_newsgroups.html.

AG's News, DBpedia and Yahoo! answers:
We used the version from: https://pathmind.com/wiki/open-datasets (look for the link *Text Classification Datasets*).

SMS spam: http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/

ISEAR: https://www.unige.ch/cisa/research/materials-and-online-research/research-material/.

B Additional reference methods

The results of NB_{BoW}, NB_{GloVe}, SVM_{BoW} and SVM_{GloVe} are shown in Figure 5.

sIB-based classifier As mentioned in §5.4, we experimented with building a rudimentary classifier that utilizes only the sIB clustering results and the labeling budget. Results for this setting are depicted in Fig. 5 in orange. Comparing these results to the BERT-based approaches reveals that clustering alone is not sufficient.

C Additional clustering techniques

Fig. 6 depicts the comparison of the sIB over BOW representation, denoted $BERT_{IT:CLUST}$, to two other configurations for the clustering intermediate task: K-means over GloVe representations and Hartigan's K-means (Slonim et al., 2013) over GloVe. The GloVe representation for each text is an average of GloVe representations for the individual tokens. The comparison reveals that in most cases sIB over BOW outperforms the other clustering configurations.

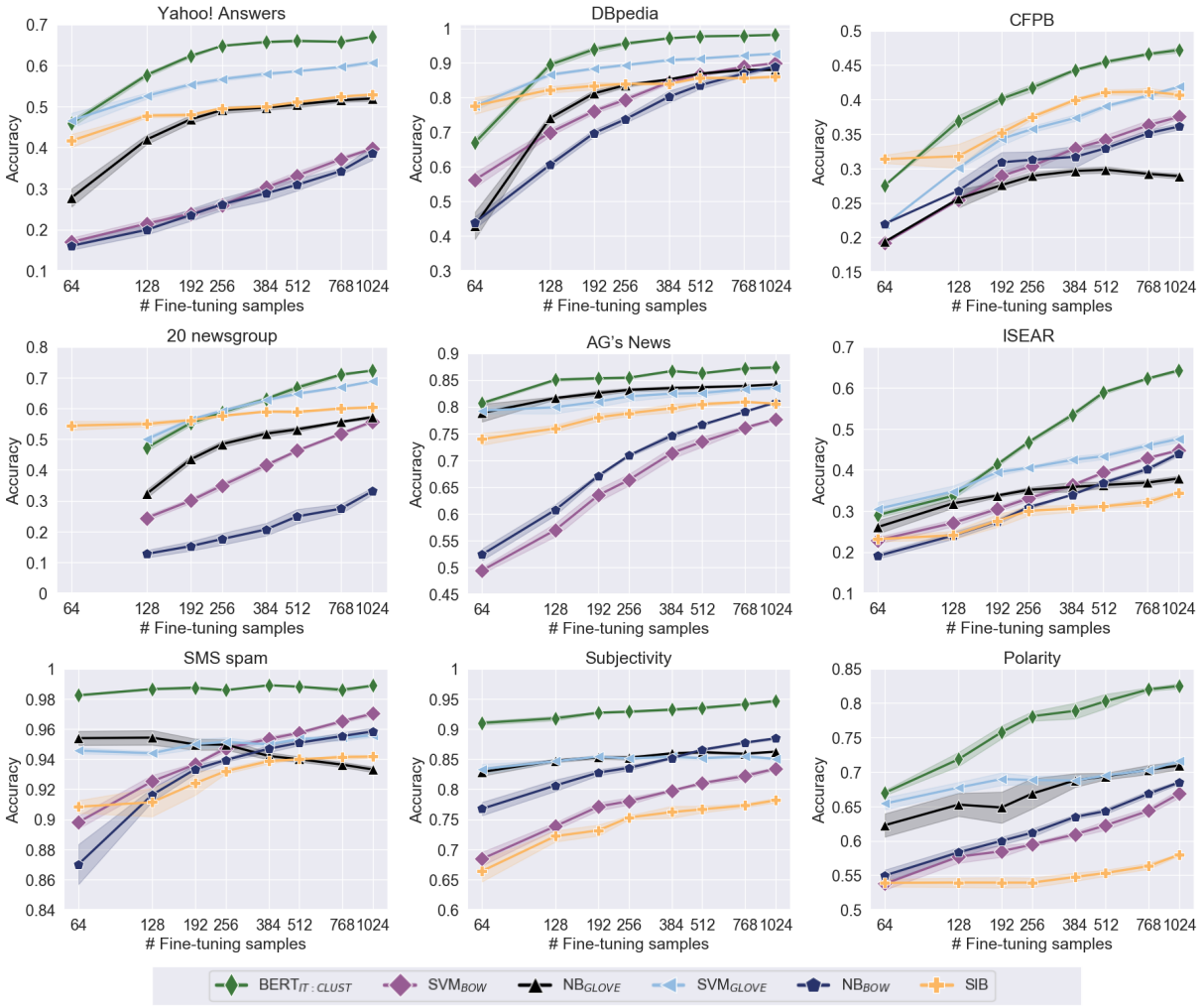

Figure 5: Comparing BOW methods and the BERT$_{\text{IT:CLUST}}$ setting. Each point is the average of five repetitions ($\pm$ SEM). X axis denotes the budget for training in log scale, and Y accuracy of each model.

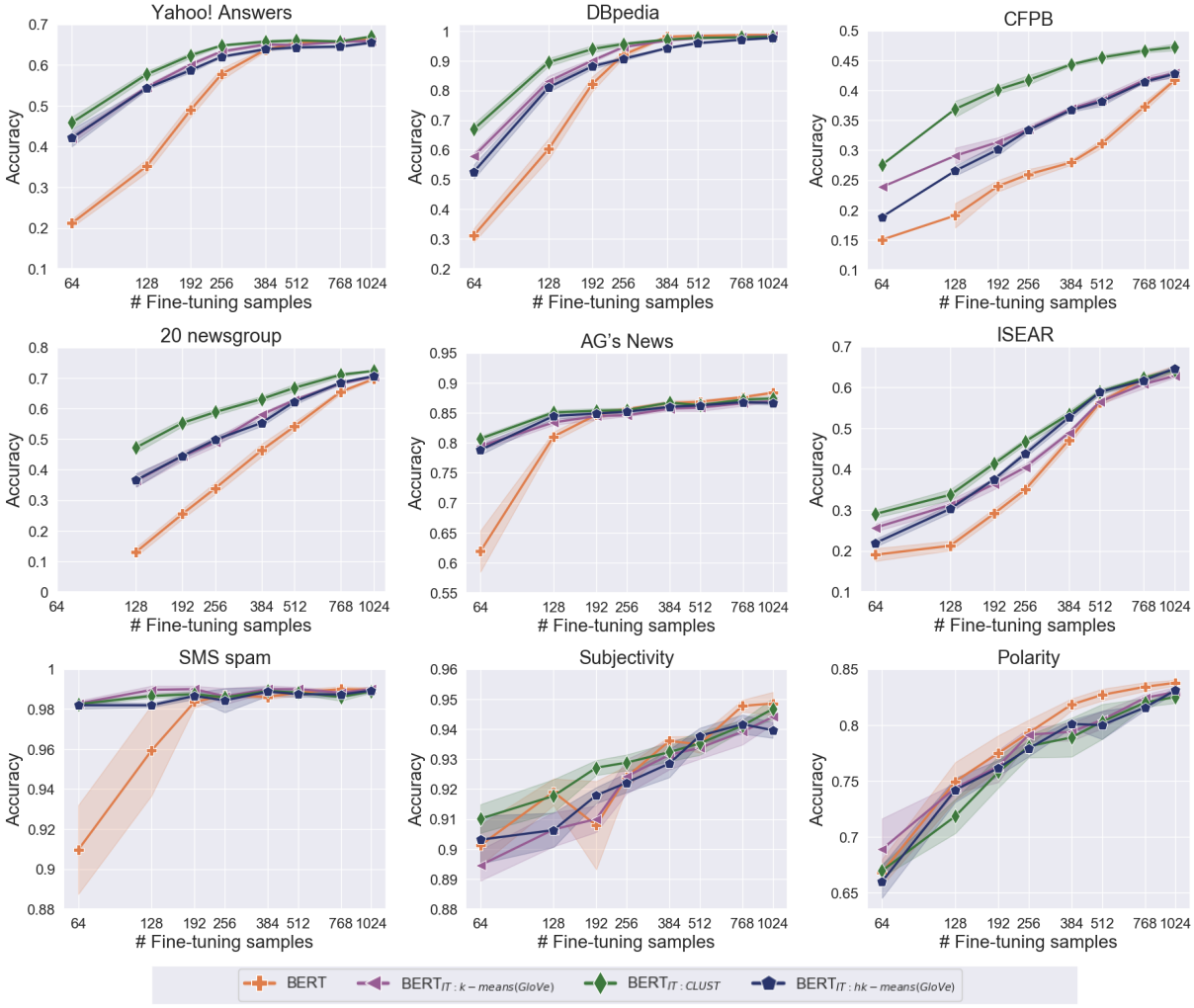

Figure 6: Comparison of clustering configurations for the intermediate task (hk-means stands for Hartigan's K-means). The results with no inter-training (BERT) are also presented for comparison. Each point is the average of five repetitions ($\pm$ SEM). X axis denotes the number of labeling instances used for fine-tuning (in log scale).