Leveraging the Structure of Pre-trained Embeddings to Minimize Annotation Effort

Cesar Gonzalez-Gutierrez **Ariadna Quattoni**
Universitat Politècnica de Catalunya, Barcelona, Spain
cesar.gonzalez.gutierrez@upc.edu,
aquattoni@cs.upc.edu

Abstract

Most current state-of-the-art approaches for text classification are based on fine-tuning the representations computed by large language models (LLMs). This strategy has led to significant improvements in classification performance and contributed to a reduction of the amount of labeled data required for training a model. However, for some challenging classification tasks, providing enough annotations to ensure a reliable classification continues to be the main bottleneck. This is especially true in settings of highly imbalanced class distributions. This paper proposes to tackle this bottleneck by exploiting structural properties of pre-trained embeddings. More precisely, we develop a label propagation method that uses pre-trained embeddings to spread information from the labeled samples to nearby samples in the induced space, ensuring the optimal use of annotations. Our approach is simple and relatively low-cost since it only requires computing some distances in the embedded space. We conduct experiments on different text classification datasets showing that the proposed method is efficient and significantly outperforms both self-training and random walk label propagation strategies.

1 Introduction

Currently, the dominant state-of-the-art approach for text classification is based on fine-tuning the representations computed by large language models (LLMs). This approach has led to significant improvements in classification performance and contributed to a reduction of the amount of supervised labeled data required for training a model. However, for some challenging classification tasks, providing enough annotations to ensure a reliable classification continues to be the main bottleneck in terms of cost and time. We would like to train models with minimal annotation effort.

In this paper, we focus on the problem of learning a text classifier under annotation budget constraints and with no prior trained model. The only assumption we make is access to a large unlabeled dataset. This is sometimes referred to as few-shot learning or the cold-start problem. Notice that our setting is different from the classical active learning scenario, where one assumes several training iterations and the algorithm alternates between sampling data and learning a new model. Our focus is on the initial setting and we exploit a semi-supervised strategy.

We are interested in classification scenarios with highly imbalanced class distributions. Class imbalance is a frequent phenomenon in real NLP applications because target categories tend to be skewed (spam detection, hate speech). Class imbalance poses an additional challenge when learning under budget constraints. This is because, even if we only require a small sample of the target class, to obtain such a sample we might still require a significant number of annotations. For instance, in a scenario with a 10% target class probability, obtaining 10 representative samples would, on average, require annotating 100 samples.

Self-training (Yarowsky, 1995) is a popular strategy for learning textual classifiers with tight annotation budgets and it is widely used by NLP practitioners (Yang et al., 2021; McClosky et al., 2006). This semi-supervised learning strategy starts by training a model with a labeled seed set. Its predictions on a large unlabeled dataset are then used to train a new model. Typically this process is repeated until the whole unlabeled dataset has been labeled.

Instead of training a model with its predictions, an alternative approach is to exploit the properties of the input space directly. Recent literature on analyzing textual representations (Gonzalez-Gutierrez et al., 2023; Yauney and Mimno, 2021; Zhou and Srikumar, 2021) shows that pre-trained embeddings exhibit some structural properties that make them especially well-suited for text classifica-

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 7000–7014
June 16-21, 2024

tion. Gonzalez-Gutierrez et al. (2023) showed that the best representations are those in which we can find latent clusters that are well aligned with the classification task. The paper concludes with the suggestion that the structure of the embedded space could be further exploited to improve the performance of classifiers when the budget for supervised annotations is tight.

Motivated by this study, we propose a strategy to leverage the structure of the embedded space to improve classifier performance under tight annotation budgets. If the latent structure of a representation is well aligned with a task, we should be able to exploit this with an appropriate label propagation strategy. With this in mind, we develop a label propagation strategy specifically designed for the imbalanced class scenario.

Additionally, a label propagation strategy might be more successful if the initial labeled set is diverse. Label propagation and diversity sampling are complementary ideas of how to exploit the structure of the embedded space. Diversity sampling prioritizes the selection of representative samples, while label propagation maximizes the utility of the chosen samples, therefore it is natural to combine them. In this paper, we combine both techniques and examine how the choice of initial samples impacts overall performance under annotation budget constraints.

Our experiments on text classification show that our approach is very effective for fine-tuning classifiers under tight annotation budgets, especially in imbalanced scenarios. Our technique outperforms both self-training and label propagation strategies. Furthermore, although the setting is different, we compare our approach against a classical active learning baseline and show that it leads to significant reductions in the annotations required to achieve a given classification performance. Overall, the results show that our proposed propagation strategy leads to significant performance improvements independent of the initial seed sampling strategy. However, combining label propagation and diversity sampling does lead to further improvements with very low annotation budgets.

In summary, our main contributions are:

- We develop a label propagation method for training text classifiers with minimal supervision. Our approach is designed to exploit structural properties of pre-trained embeddings. The proposed method is simple and low-cost, it only requires computing distances in the embedded space. We believe that it can become a practical tool for working with tight annotation budgets.
- We conduct experiments on various datasets showing that the proposed method is very effective and outperforms both self-training and label propagation strategies.
- We study the combination of semi-supervised learning with seed diversity sampling and show that in low-budget scenarios it can lead to further improvements.

2 Leveraging Pre-trained Embeddings for Learning with an Annotation Budget

In this section, we present our proposed semi-supervised approach, which leverages pre-trained embeddings to learn textual classifiers under tight annotation budgets.

2.1 Learning Setting

Let $\boldsymbol{U} = \{\boldsymbol{x}_i\}_{i=1}^{n}$ be an unlabeled dataset where $\boldsymbol{x}_i$ is a text document and n is the size of the dataset. We let $\mathcal{Y}$ be the set of label classes. In addition, we have at our disposal an oracle $O(\boldsymbol{x}_i) = y_i$ that we can query to obtain the label of a sample. Finally, in the learning-under-a-budget setting, we also assume that we have an annotation budget of B so that we can only query the oracle for B annotations (we assume a unit cost for each annotation).

We apply a seed selection mechanism to pick a subset of the documents and label them. This constitutes our seed set $\boldsymbol{L}$ of size B. Our goal is to leverage information from the remaining samples in $\boldsymbol{U}$ to obtain the best possible classifier under the given budget constraints.

2.2 Graph Label Propagation over Pre-trained Embeddings (GLPE)

The general steps of our method are the following. We first use an LLM to compute an embedding for each document in the dataset. Once both labeled and unlabeled documents have been projected to the embedding space, we compute affinity matrices between them for each target class. The affinity graphs are then used to propagate labels from the seeds to nearby unlabeled samples, generating a new set of pseudo-labeled samples. Later, we use both the seeds and the pseudo-labeled samples to fine-tune the final classification model. Figure 1 provides a high-level view of our method.

We assume that a small subset of the points have been labeled with their true class values, this is our

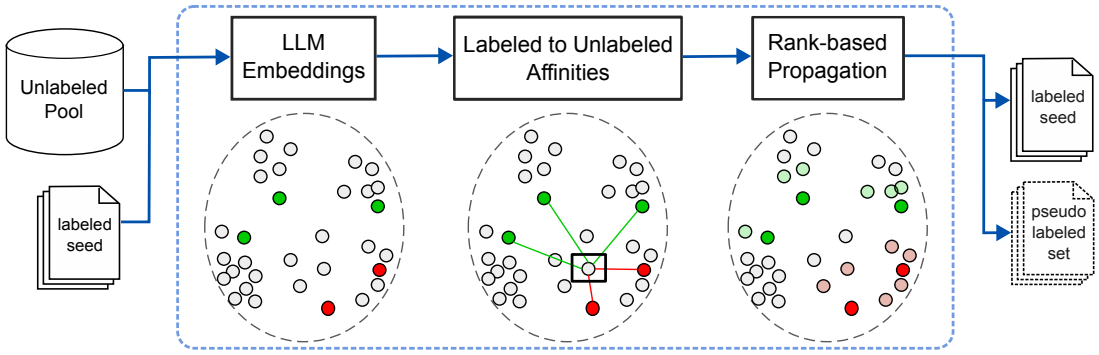


Figure 1: General schema of Graph Label Propagation over Pre-trained Embeddings (GLPE). First, an embedded representation is computed for the entire pool. The given seed samples serve to compute an affinity matrix for each label based on similarities between labeled and unlabeled points. Finally, label propagation is performed based on the highest aggregated score.

seed set $\boldsymbol{L}$. The goal of the label propagation step is to make the best use of the seed set by exploiting information in the unlabeled pool $\boldsymbol{U}$.

Gonzalez-Gutierrez et al. (2023) has shown that pre-trained embeddings for text classification have some key properties, usually referred to as *smoothness* and *manifold* (van Engelen and Hoos, 2020; Yang et al., 2021). These properties state that close samples in the embedded space will belong with high probability to the same class. Furthermore, the paper also showed that the embedded representation satisfied the *class-clustering alignment* property. This property ensures that points within the embedded space clusters will belong, with high probability, to the same class. We propose to exploit these properties via a graph-based label propagation strategy that will generate a new set of pseudo-labeled samples for fine-tuning a model.

Note that these properties refer to the relation between the embedding and the task, and differ from intrinsic (unsupervised) properties of the embedded space. Properties such as isotropy (Gao et al., 2019; Ethayarajh, 2019; Su et al., 2021) or clusterability (Xu et al., 2023) might be beneficial, but what we require is that the representation shows clustering alignment with the task.

We start by running an LLM to compute an embedded representation for every data point in the pool: $\boldsymbol{R} = \{\text{emb}(\boldsymbol{x}_i)\}_{i=1}^{n}$.

Similar to Zhu and Ghahramani (2002), we employ a kernel function to build an affinity matrix $\boldsymbol{A}^c \in \mathbb{R}^{n \times p}$ for each class label c. Where n is the number of unlabeled samples, and p is the number of samples of class c contained in the labeled set L. The affinity matrices are defined as:

$$\boldsymbol{A}^c_{ij} = \exp\left(-\frac{d(\boldsymbol{r}_i, \boldsymbol{r}_j)}{\sigma^2}\right) \tag{1}$$

where $d(\cdot)$ is some distance metric, σ is a bandwidth parameter, and $\boldsymbol{r}_i \in \boldsymbol{R}$ and $\boldsymbol{r}_j \in \boldsymbol{R}$ are the embeddings for the i^{th} unlabeled point and the j^{th} seed labeled as c. Every row of $\boldsymbol{A}^c$ measures the affinity between an unlabeled sample and each of the seeds labeled as c. Each column corresponds to the affinity between a labeled seed and each of the unlabeled samples.[1]

To create a new set of pseudo-labeled samples, we start by computing a single score for each unlabeled point and label class. Using the affinity matrix we define the aggregate score for the i^{th} unlabeled point and label class c as:

$$\boldsymbol{S}^c_i = \frac{1}{p}\sum_{j=1}^{p} \boldsymbol{A}^c_{ij} \tag{2}$$

In other words, the compatibility score of an unlabeled point and a target class is proportional to the average affinity of that point to the set of labeled seeds of that class.[2]

After computing the affinity scores we rank all unlabeled samples and create a new set of pseudo-labeled samples by selecting a total of k top-ranking labeled samples. Where k and its distribution among the class labels are parameters of the label propagation strategy. More details about how these parameters are set are provided in Section 3.

[1] Constructing the graph with edges solely between labeled-unlabeled pairs decreases the number of distances to calculate from quadratic on the number of samples (n^2), as in a fully connected graph, to linear given a fixed budget: $B \cdot |U|$.

[2] Other aggregations (min., max.) are also possible.

This algorithm can work iteratively, where k is set incrementally, and for each step, the affinity matrices are computed from the seed sample and the pseudo-labels computed in the previous step.

3 Experiments

3.1 Experimental Setup

With the data annotation bottleneck problem in mind, we evaluate our proposed method under annotation budget constraints on three highly imbalanced binary classification datasets and three multi-class datasets. We compare our strategy with other strategies for learning under tight annotation budgets. In all cases, the classification model, pre-trained embeddings, and fine-tuning step are the same.

In this section, we describe the datasets used, implementation details of the pre-trained embeddings and base classifier, and the different learning strategies to be evaluated. Further details and a list of hyper-parameters used in the experiments can be found in A.1.

3.1.1 Datasets

We performed experiments on three binary imbalanced datasets: AG News (Zhang et al., 2016), imbalanced as in Ein-Dor et al. (2020) and using world news as the target class; Wikipedia toxic comment detection (Wiki Toxic; Wulczyn et al., 2017); and Civil Comments (Borkan et al., 2019) using toxicity labels. We also study the multi-class setting on Swahili News (David, 2020) topic classification; Ohsumed (Hersh et al., 1994) medical abstract MeSH categorization; and TREC (Li and Roth, 2002; Hovy et al., 2001) coarse question classification. Table 1 shows the dataset statistics.

Dataset	$\lvert\mathcal{Y}\rvert$	Prior	Pool	Test	Len.
AG News	2	10.0	15k	2.5k	38
Wiki Toxic	2	9.6	160k	64k	68
Civil Comments	2	8.0	1.9M	97k	53
Swahili News	6	imb.	22.2k	7.34k	327
Ohsumed	23	imb.	54.7k	294k	269
TREC	6	$1/\lvert\mathcal{Y}\rvert$	5.45k	500	10

Table 1: Number of classes $|\mathcal{Y}|$, class prior (%), pool and test partition sizes, and average sequence length.

3.1.2 Pre-trained Embeddings and Base Classification Model

We obtained the embeddings for the dataset samples using a pre-trained $\text{BERT}_{\text{base}}$ model (Devlin et al., 2019). To generate sentence embeddings, we extracted the activations of the second-to-last layer and performed average-pooling of all tokens (BERT_2^{μ}).

The classification model of choice was a pre-trained $\text{BERT}_{\text{base}}$ implemented using HuggingFace Transformers library (Wolf et al., 2020). For Swahili News, we used SwahBERT (Martin et al., 2022), a pre-trained BERT for Swahili.

Model parameters were validated using the available labeled samples in an 80%/20% train-development partition (we do not assume a fixed development set). The model was then trained with 100% of the available labels. We repeated each experiment 5 times using different random initialization seeds and reported the average performance.

3.1.3 Learning Under a Budget Strategies

- **BERT**: As a baseline, we consider a BERT model fine-tuned with the seed sample alone.
- **Self-Training** (Yarowsky, 1995): A classic strategy that trains the model with its own predictions. It operates iteratively, training a model and making predictions over the unlabeled set. All samples whose prediction confidence is above a certain threshold are added to the training set and the model is trained again with the additional pseudo-labeled samples. This process is repeated until no unlabeled samples satisfy the confidence threshold condition or until all unlabeled samples have been added to the training set. We used a threshold of 0.75, fine-tuned the first model for 5 epochs and, for each iteration, further fine-tuned for 1 epoch on the propagated set.
- **Random Walk Label Propagation** (RW LP; Zhu and Ghahramani, 2002): A classic graph-based technique that works on a similarity matrix constructed by applying a kernel function to all data points in the embedded space. This algorithm performs stochastic random walks iteratively until a steady state is reached, clamping the known hard labels at each step.
- **Hierarchical Sampling Admissible Pseudo-Labeling** (HS+APL; Dasgupta and Hsu, 2008): This is an approach that combines a seed diversity hierarchical sampling strategy (described later in Section 4.1) with a hierarchical pseudo-labeling schema for predicting pseudo-labels for all the unlabeled data points. The pseudo-labeling algorithm assigns labels that minimize the expected

Method	AG News		Wiki Toxic		Civil Com.		TREC		Ohsumed		Swah. News	
	100	200	100	200	100	200	100	200	100	200	100	200
BERT	31.8_{27}	73.8_{12}	37.5_{13}	44.8_{10}	$9.3_{2.5}$	$8.4_{3.6}$	$15.9_{4.2}$	56.0_{38}	$7.1_{5.4}$	$11.0_{6.2}$	$59.8_{9.6}$	$75.0_{2.9}$
Self-Tr.	51.9_{23}	50.6_{21}	33.2_{23}	45.2_{17}	$7.6_{0.5}$	$8.5_{1.1}$	$18.4_{5.1}$	$18.2_{4.5}$	$11.5_{4.4}$	$13.8_{3.2}$	$41.9_{8.2}$	$48.6_{6.6}$
RW LP	$78.3_{5.8}$	$83.1_{3.3}$	$17.5_{4.2}$	$14.4_{6.3}$	19.7_{18}	$8.7_{1.3}$	40.4_{15}	$59.0_{2.8}$	$11.7_{4.7}$	$13.1_{4.6}$	$69.7_{0.5}$	$70.3_{0.6}$
HS+APL	$74.9_{9.8}$	$83.4_{2.3}$	$38.5_{6.3}$	46.8_{16}	$9.5_{1.7}$	$16.6_{5.1}$	42.0_{12}	$57.4_{5.8}$	$13.7_{2.0}$	$14.7_{0.8}$	**$76.0_{2.8}$**	$76.1_{2.6}$
MixText	49.9_{22}	65.0_{11}	38.9_{11}	46.0_{10}	$9.1_{1.1}$	$12.4_{4.9}$	$59.6_{5.4}$	$73.0_{5.8}$	$13.0_{2.6}$	$16.8_{4.1}$	$58.6_{6.0}$	$69.3_{4.0}$
GLPE	**$82.9_{1.3}$**	$83.8_{1.1}$	54.3_{11}	$62.3_{2.4}$	**$20.7_{9.5}$**	**$28.5_{7.5}$**	49.6_{21}	54.2_{22}	$15.3_{1.0}$	$18.6_{0.9}$	$74.6_{4.5}$	**$78.8_{1.2}$**
iGLPE	$82.4_{1.4}$	**$83.9_{1.0}$**	**$60.3_{5.3}$**	**$64.2_{2.6}$**	$14.6_{4.5}$	$18.6_{6.3}$	**$71.9_{2.3}$**	**$74.3_{1.0}$**	**$21.5_{1.5}$**	**$21.8_{1.3}$**	$74.7_{2.4}$	$78.1_{3.1}$
e-kNN	$82.1_{0.7}$	$82.3_{1.9}$	$48.4_{8.3}$	$50.4_{3.2}$	$16.8_{5.3}$	$20.8_{3.4}$	$25.1_{8.0}$	$23.8_{5.0}$	$12.8_{3.4}$	$17.9_{0.6}$	$66.7_{3.7}$	$68.1_{0.7}$

Table 2: Performance of semi-supervised methods on various datasets and annotation budgets. We report AUC% for binary imbalanced datasets (AG News, Wiki Toxic, Civil Comments) and accuracy% for multi-class datasets (TREC, Ohsumed, Swahili News) with standard deviation in subscripts.

error of the assignment, relying on the estimated purity of each node. The expected error is estimated by considering the structure of the dendrogram and the labeled samples.

- **MixText** (Chen et al., 2020): combines interpolation of samples in the model hidden space for consistency regularization with supervised learning using self-training pseudo-labeled samples. For a fair comparison, we have removed extrinsic data augmentation such as back translation.
- **GLPE**: Our strategy presented in Section 2.2. We use cosine similarity as the distance metric and $\sigma = 1$ for the bandwidth parameter. We fixed the parameter k (the number of pseudo-labeled samples to add to the training dataset) to 4000. As for the sampling ratio assigned to each label class, we first estimate the class priors using the available annotations in the seed and set the ratios proportional to its class prior. For example, for a binary dataset for which the estimated prior of the positive class is 10%, we will generate 400 positive samples and 3600 negative samples. In general, this parameter could also be cross-validated to obtain further improvements.
- **iGLPE**: Iterated variant of GLPE using steps of $k_s = 100$ until $k = 4000$. The other parameters are the same as the non-iterative version.
- **Embedding kNN**: In addition to using our base classifier, we employ the embedding representations as classifiers. Based on the label class distribution computed by our method (Section 2.2), we compute the k nearest neighbors and average their corresponding label probabilities. For consistency with our strategy, we set k to 4000.

3.2 Results

In this section, we compare the performance of different strategies by training our base classifier with the methods described in Section 3.1.3. For binary imbalanced datasets, we report the area under the precision-recall curve of the target class (AUC). This metric is a natural choice for highly imbalanced scenarios and avoids the need for validating thresholds. For multi-class datasets, we report accuracy as is common practice. We tested the performance of each method trained with two budget sizes: 100 and 200 annotations. Table 2 shows a summary of the results obtained.

Overall, we observe that self-training is an unstable strategy. When applied to datasets like Wiki Toxic and Civil Comments, self-training exhibits erratic performance, potentially indicating the presence of confirmation bias in its outcomes. Random walk label propagation shows a similar unstable behavior. Additionally, MixText fails to achieve significant performance improvements within this more constrained learning framework.

Directly using the embedded representation to make predictions with kNN outperforms the self-training strategy but falls behind the label propagation strategies. Interestingly, for very low annotation budgets, a kNN over the pre-trained representation outperforms consistently the fine-tuned model BERT baseline.

In contrast, our proposed strategy GLPE **consistently outperforms** all the other strategies and shows significant improvements over the baseline for all datasets and budgets. It also shows a surprisingly **stable** behavior, which is a well-known challenge when leveraging LLMs under tight anno-

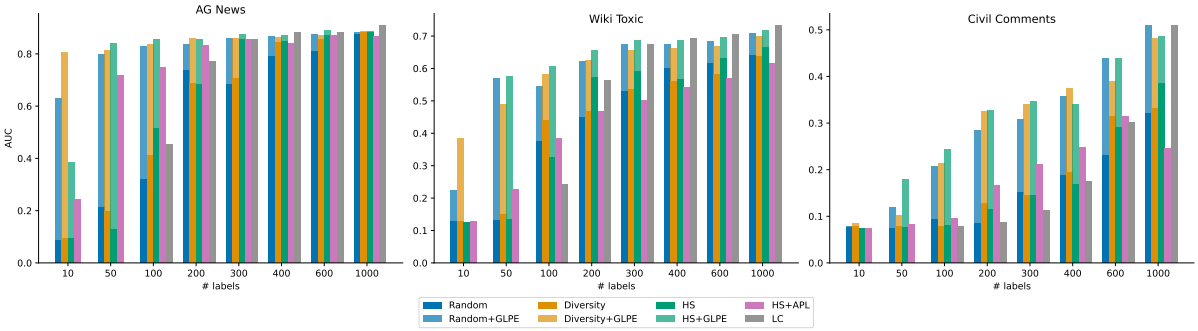


Figure 2: Performance (AUC) of different seed sample strategies with and without GLPE for increasing annotation budgets. HS+APL and LC are also shown for comparison.

tation budgets (Margatina et al., 2022; Zhang et al., 2021). Additionally, the generation of pseudo-labels is done more **efficiently** compared to other strategies (see A.3 for a runtime comparison).

4 GLPE Ablation Studies

4.1 Seed Diversity Sampling

To further exploit the structure of the embedded space we propose to select the seed set to label so that the resulting annotated data is as diverse as possible. In some cases, when the annotation budget is very tight we expect that selecting a diverse seed will lead to a more useful label propagation. This is because when the initial seed covers a larger portion of the space, the pseudo-labeled samples tend to contain more information.

We experiment with different diversity seed selection strategies designed to leverage the class-clustering alignment property of the embedded space. More precisely, we test two diversity sampling strategies, namely: clustering diversity and hierarchical sampling.

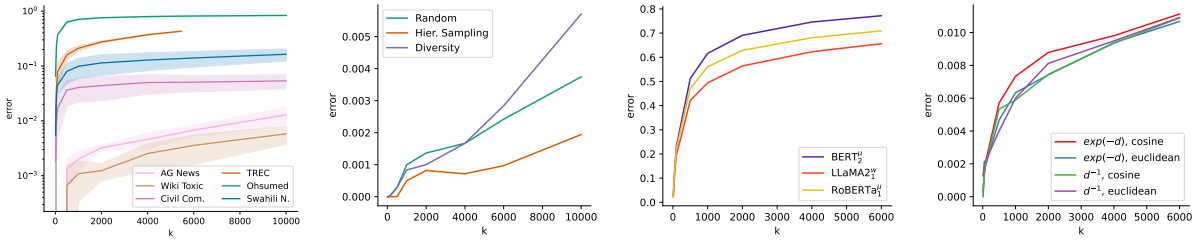
- **Clustering Diversity**: The main idea of this selection approach is to first cluster the data points and then select one representative sample per cluster for annotation. We employed hierarchical clustering with Ward's (1963) method to build the data dendrogram. We set the number of clusters to be equal to the annotation budget size B. Then for each of the B clusters we select the sample that is closest to the cluster's center and add it to the seed set.
- **Hierarchical Sampling** (HS; Dasgupta and Hsu, 2008): In this method, the seed set is built one sample at a time in an active manner. This strategy explores the dataset dendrogram obtained by hierarchical clustering to get the most informative and diverse set of seeds. Active exploration ensures that more seeds are selected from dendrogram nodes (clusters) whose class distribution is estimated to be less pure. That is, we select more seeds to label from clusters that are expected to contain samples from multiple classes.
- **Least Confidence** (LC; Lewis and Gale, 1994): While the active learning setting differs from the learning under an annotation budget scenario, it remains a popular strategy to mitigate the annotation data bottleneck. Therefore, we also compare our method to a classical uncertainty sampling strategy: least confidence.

We tested the performance of each method in binary imbalanced datasets when trained with annotation budgets of increasing sizes, thus obtaining learning curves with performance as a function of the budget size. Figure 2 shows a summary of the results obtained using GLPE with different seed sampling strategies. A.2 contains learning curves of other semi-supervised methods in combination with seed sampling.

The first observation is that HS+APL leads to some improvements compared to BERT, but the improvements are not consistent. For example, for Wiki Toxic and budgets larger than 200, HS without APL propagation seems to be better than with propagation.

In contrast, GLPE propagation always leads to better performance, irrespective of the initial seed selection strategy. The only exception is Civil Comments with very low budgets, where performance is the same with or without label propagation.

Most of the improvements in performance over the BERT baseline are obtained because of label propagation. Diversity and Hierarchical Sampling seed selection alone (without label propagation)



(a) Dataset (b) Sampling (c) Embedding (d) Kernel

Figure 3: GLPE propagation error as a function of parameter k, for a budget of 200, randomly sampled, and averaged over 5 seeds. a) Error curves for different datasets (y-axis in the log-scale). b) Seed sample selection strategies for Wiki Toxic dataset. c) Choices of embedding model for Ohsumed dataset. d) Kernel and distance functions for Wiki Toxic, AG News, and Civil Comments combined.

do lead to some improvement over random seed selection, but the improvements are not consistent across datasets and budgets.

When combined with GLPE, the initial seed selection method does not seem to be very important. In other words, regardless of the initial seed selection method, GLPE seems to generate an effective and informative set of pseudo-labeled samples. With one exception, for very low budgets diversity seed selection does lead to significant improvements over the random seed selection method for AG News and Wiki Toxic datasets.

Active learning performs well with medium-sized budgets. Least Confidence consistently surpasses Random with larger annotation budgets. However, with tighter annotation budgets, active learning shows the opposite tendency and falls short of the baseline.

4.2 Representation and Classifier Choice

We now turn our attention to the relative importance of the representation used to build the affinity matrices in GLPE and the classifier used with the pseudo-labels. We compare performance using three classifiers: BERT and kNN classifiers described in Section 3, and RoBERTa$_{\text{base}}$ (Liu et al., 2019). As a baseline, we train the models without pseudo-labels.

We compare three representations: BERT$_2^\mu$, our base representation; RoBERTa$_1^\mu$, a sentence-transformers (Reimers and Gurevych, 2019) RoBERTa representation, fine-tuned for the Semantic Textual Similarity Benchmark (Cer et al., 2017), using average pooling of the last layer; and LLaMA2$_1^w$, embeddings based on LLaMA2-17B (Touvron et al., 2023) using weighted average-pooling of the last layer. See A.1 for further details.

We test performance on Ohsumed dataset, the most challenging one. Table 3 shows the results obtained. As expected, we observe performance gains using representations fine-tuned for text similarity (RoBERTa$_1^\mu$) or with very long pre-training (LLaMA2$_1^w$). Classifiers show a similar tendency but their behavior is more erratic, probably due to different tolerances to noise. The performance variability of embedding kNN across representations for low budgets can be attributed to the parameter k, which was initially set according to the performance of BERT$_2^\mu$ representation in GLPE.

Represent.	**e-kNN**		**BERT**		**RoBERTa**	
	100	200	100	200	100	200
No label pr.	-	-	7.1	11.0	14.5	18.0
BERT$_2^\mu$	**12.8**	17.9	15.3	18.6	15.7	18.4
RoBERTa$_1^\mu$	6.5	15.7	21.9	25.0	21.8	25.6
LLaMA2$_1^w$	10.3	**20.2**	**24.5**	**27.8**	**23.7**	**28.0**

Table 3: GLPE performance (accuracy%) using different embeddings and classifiers on Ohsumed dataset.

4.3 Pseudo-Labeling Quality

In this section, we proceed to examine the pseudo-label propagation quality without employing the end model for classification. For this purpose, we use the gold labels of the datasets to calculate the error rate of the pseudo-labels generated by the different extrinsic semi-supervised methods.

Table 4 shows the error rate and size of the propagation of GLPE in comparison with other techniques. These values can explain the performances found in Table 2. Self-training tends to pseudo-label the whole unlabeled set, and incur big confirmation biases, or fails to pseudo-label at all when the model is less confident. Label propagation has a similar confirmation bias. Instead, GLPE achieves

Budget	Dataset	Self-Train. error	Self-Train. size	RW LP error	RW LP size	HS+APL error	HS+APL size	GLPE error	GLPE size	iGLPE error	iGLPE size
100	AG News	$9.9_{0.0}$	**14.9k**	$5.8_{1.0}$	**14.9k**	$6.7_{2.8}$	10k	$0.7_{0.4}$	4k	**0.3**$_{0.1}$	4k
	Wiki Toxic	$9.6_{0.0}$	**159.5k**	$9.6_{0.0}$	**159.5k**	$9.3_{0.2}$	10k	**0.2**$_{0.1}$	4k	**0.2**$_{0.1}$	4k
	Civil Com.	$8.0_{0.1}$	**1.9M**	$8.4_{0.5}$	**1.9M**	$7.8_{0.3}$	10k	**5.6**$_{2.1}$	4k	$7.1_{2.2}$	4k
	TREC	**0.0**$_{0.0}$	0	$56.3_{4.7}$	5.1k	$57.6_{3.4}$	**10k**	$37.2_{0.9}$	4k	$32.1_{1.5}$	4k
	Ohsumed	**0.0**$_{0.0}$	0	$86.6_{4.7}$	**10.3k**	$85.6_{2.2}$	10k	$79.6_{0.9}$	4k	$70.5_{1.0}$	4k
	Swah. News	**0.0**$_{0.0}$	0	$30.5_{0.5}$	**22.1k**	$23.6_{0.4}$	10k	$12.8_{4.6}$	4k	$13.2_{3.1}$	4k
200	AG News	$9.4_{0.9}$	**14.8k**	$4.7_{1.2}$	**14.8k**	$5.4_{2.4}$	10k	**0.5**$_{0.1}$	4k	$0.6_{0.4}$	4k
	Wiki Toxic	$9.5_{0.1}$	**159.4k**	$9.6_{0.0}$	**159.4k**	$8.5_{0.8}$	10k	**0.1**$_{0.0}$	4k	**0.1**$_{0.0}$	4k
	Civil Com.	$8.0_{0.1}$	**1.9M**	$8.3_{0.6}$	**1.9M**	$7.7_{0.3}$	10k	**3.3**$_{1.5}$	4k	$5.3_{2.2}$	4k
	TREC	**0.0**$_{0.0}$	0	$46.8_{2.1}$	5.2k	$51.3_{3.4}$	**10k**	$32.9_{1.4}$	4k	$29.6_{1.2}$	4k
	Ohsumed	**0.0**$_{0.0}$	0	$84.2_{4.8}$	**10.2k**	$83.4_{0.8}$	10k	$74.6_{1.4}$	4k	$68.4_{0.7}$	4k
	Swah. News	**0.0**$_{0.0}$	0	$29.5_{0.6}$	**22k**	$23.3_{0.6}$	10k	$11.1_{2.7}$	4k	$12.7_{1.9}$	4k

Table 4: Propagation error (%) and number of pseudo-labels added (size) for the extrinsic semi-supervised techniques studied in this work.

relatively low pseudo-labeling error and controls the propagation size by fixing the k parameter.

Figure 3 shows curves with the propagation error incurred by GLPE as a function of k. In Figure 3a, error rate curves for different datasets illustrate their relative difficulty and the trade-off between propagation size and propagation error. The choice of k heuristically balances this trade-off for diverse tasks, yet validation could enhance the propagation quality. Figure 3b shows how various seed sampling strategies obtain slightly different propagation errors, parallel to the performance observed in Section 4.1. These strategies do not obtain consistently better propagation and depend on the choice of k. Instead, Figure 3c demonstrates how the model used to generate the embeddings can reliably improve the quality of the propagation and the subsequent performance, as seen in Section 4.2. Finally, other distance measures and kernel functions (Figure 3d) obtain similar propagation qualities, showing the robustness of GLPE to this choice.

5 Related Work

Semi-supervised learning techniques that leverage unlabeled data to improve the performance of models trained with tight annotation budgets have a long history in NLP and are still widely used to mitigate the data annotation bottleneck. We can distinguish two main approaches: those based on self-training and those based on graph-based regularization.

Self-training is a popular semi-supervised strategy for text classification and has been explored by several recent works (Chen et al., 2020, 2022; Karamanolakis et al., 2021), with a focus on zero-shot or few-shot scenarios (Gera et al., 2022; Chen et al., 2021; Ye et al., 2020), light architectures (Liu et al., 2021), or the multilingual setup (Dong and de Melo, 2019).

One of the main known limitations of self-training is that it can suffer from *confirmation bias* which happens when the model repeatedly over-fits and assigns incorrect pseudo-labels. Since these pseudo-labels are used to retrain the model, this repeated over-fitting can make the model diverge from the true class distribution. This problem is especially critical with over-parameterized models such as transformers because they tend to overfit.

To avoid the *confirmation bias* problem an alternative approach is to consider strategies that directly exploit the structure of the unlabeled space. Graph-based regularization approaches for semi-supervised classification have been proposed in this context. These techniques first compute a similarity graph among unlabeled data points, using a representation space. Then *soft* label propagation is implemented implicitly by modifying the training loss function, using unlabeled data as a form of regularization. This is achieved by including a term in the optimization function that biases the classifier to provide labels for unlabeled points that are similar according to the graph. We term it *soft* label propagation because the regularization penalty implicitly imposes some regularities on the

label assignments of unlabeled points.

One of the first graph-based semi-supervised approaches for text classification was introduced in Ozaki et al. (2011); Ren et al. (2011), before the emergence of pre-trained LLMs. More recently, Saraiva et al. (2021) used a graph-based method for toxic comment detection. Similar to our work, they also use pre-trained embeddings to build the similarity graph over the unlabeled data points. However, their approach differs significantly from ours, as they use the graph to regularize the loss function of simpler classifiers like decision trees and support vector machines.

Another line of work has studied the use of graph-based semi-supervised techniques in multi-label classification (Taha et al., 2022). In this case, the objective is somehow different since the graph regularization is used to infer missing labels in the label set associated with a document. Unlabeled data might as well be multi-modal (e.g. images and text). Sirbu et al. (2022) proposed a semi-supervised graph-based method for this setting.

One potential disadvantage of graph-based methods is that they tend to be classifier-specific, that is, they are specifically designed with one classification function in mind. Another disadvantage is that they can be computationally expensive because typically to train a model the loss function is modified to include constraints between all (or a large subset) of pairs of unlabeled points.

To mitigate the inefficiency and lack of generality of graph-based methods, an alternative is to consider explicit label propagation techniques. These methods generate a set of pseudo-labeled samples to be used by any training algorithm. Different from graph-based methods, the model is regularized by giving it additional pseudo-labeled data instead of via loss constraints. This approach can have the advantages of more costly graph-based approaches while being simpler, more efficient, and general (applicable to any classification model).

In this paper, we decided to follow this approach and develop an efficient and effective label propagation strategy for text classification. Interestingly, D'Sa et al. (2020) studied the application of random walk label propagation over pre-trained embeddings and reached a negative conclusion. In contrast, we found our proposed label propagation technique to be very effective. As we have shown, not all label propagation techniques are equally effective. In particular, we have seen how random walk label propagation leads to noisy pseudo-labels.

To summarize, while there has been a significant amount of work in self-training and graph-based methods for text classification, to the best of our knowledge, we are the first ones to show that a simple and efficient semi-supervised explicit label propagation strategy in the pre-trained embedded space can lead to significant and consistent improvements when learning classifiers under tight annotation budgets.

6 Conclusion and Future Work

This study has illustrated the effectiveness of implementing label propagation within the embedded spaces derived from LLMs. The clustering alignment property of the LLM embedded space explains the method's success. One of the key distinctions that set our propagation strategy apart from self-training is its inherent resistance to over-fitting in the fine-tuning phase. This is because label propagation primarily exploits the embedded space, making it less prone to the pitfalls associated with over-fitting. Our findings suggest that the specific strategy employed for seed selection is less critical than previously assumed.

Overall, our paper demonstrates that a simple and efficient strategy can be effective for training classifiers in scenarios with imbalanced class distribution and limited annotated data. This opens up possibilities for improving semi-supervised learning strategies in scenarios closer to real-world applications.

Future work will explore the combination of label propagation with active learning techniques, such as uncertainty sampling. We intend to investigate the role of seed selection in this context, as it may still yield performance gains when combined with active strategies.

Limitations

We have focused our study on imbalanced binary classification and multi-class classification. The conclusions driven by the experiments can not be readily extrapolated to other tasks such as multi-label classification without further experiments. A more extensive empirical study would be necessary to draw robust conclusions for text classification under tight annotation budgets in those settings.

Our approach is essentially semi-supervised and while our empirical study has provided valuable insights about the combination of label propaga-

tion and seed sampling for learning under tight annotation budgets, additional research would be necessary to study the combination of active learning with semi-supervised learning.

Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 853459. The authors gratefully acknowledge the computer resources at ARTEMISA, funded by the European Union ERDF and Comunitat Valenciana as well as the technical support provided by the Instituto de Física Corpuscular, IFIC (CSIC-UV). This research is supported by a recognition 2021SGR-Cat (01266 LQMC) from AGAUR (Generalitat de Catalunya).

References

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2019. Nuanced Metrics for Measuring Unintended Bias with Real Data for Text Classification. *arXiv:1903.04561 [cs, stat]*.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Hui Chen, Wei Han, and Soujanya Poria. 2022. SAT: Improving semi-supervised text classification with simple instance-adaptive self-training. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6141–6146, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. MixText: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2147–2157, Online. Association for Computational Linguistics.

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.

Sanjoy Dasgupta and Daniel Hsu. 2008. Hierarchical sampling for active learning. In *Proceedings of the 25th International Conference on Machine Learning - ICML '08*, pages 208–215, Helsinki, Finland. ACM Press.

Davis David. 2020. Swahili : News classification dataset. The news version contains both train and test sets.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Xin Dong and Gerard de Melo. 2019. A robust self-learning framework for cross-lingual 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 6306–6310, Hong Kong, China. Association for Computational Linguistics.

Ashwin Geet D'Sa, Irina Illina, Dominique Fohr, Dietrich Klakow, and Dana Ruiter. 2020. Label propagation-based semi-supervised learning for hate speech classification. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 54–59, Online. Association for Computational Linguistics.

Liat Ein-Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2020. Active Learning for BERT: An Empirical Study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7949–7962, Online. Association for Computational Linguistics.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. 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 55–65, Hong Kong, China. Association for Computational Linguistics.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tieyan Liu. 2019. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*.

Ariel Gera, Alon Halfon, Eyal Shnarch, Yotam Perlitz, Liat Ein-Dor, and Noam Slonim. 2022. Zero-shot text classification with self-training. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1119, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Cesar Gonzalez-Gutierrez, Audi Primadhanty, Francesco Cazzaro, and Ariadna Quattoni. 2023. Analyzing text representations by measuring task alignment. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 70–81, Toronto, Canada. Association for Computational Linguistics.

William Hersh, Chris Buckley, T. J. Leone, and David Hickam. 1994. Ohsumed: An interactive retrieval evaluation and new large test collection for research. In *SIGIR '94*, pages 192–201, London. Springer London.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. 2001. Toward semantics-based answer pinpointing. In *Proceedings of the First International Conference on Human Language Technology Research*.

Giannis Karamanolakis, Subhabrata Mukherjee, Guoqing Zheng, and Ahmed Hassan Awadallah. 2021. Self-training with weak supervision. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 845–863, Online. Association for Computational Linguistics.

David D. Lewis and William A. Gale. 1994. A sequential algorithm for training text classifiers. *CoRR*, abs/cmp-lg/9407020.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Chen Liu, Zhang Mengchao, Fu Zhibing, Panpan Hou, and Yu Li. 2021. FLiText: A faster and lighter semi-supervised text classification with convolution networks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2481–2491, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

Katerina Margatina, Loic Barrault, and Nikolaos Aletras. 2022. On the importance of effectively adapting pretrained language models for active learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 825–836, Dublin, Ireland. Association for Computational Linguistics.

Gati Martin, Medard Edmund Mswahili, Young-Seob Jeong, and Jiyoung Woo. 2022. SwahBERT: Language model of Swahili. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 303–313, Seattle, United States. Association for Computational Linguistics.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 152–159, New York City, USA. Association for Computational Linguistics.

Kohei Ozaki, Masashi Shimbo, Mamoru Komachi, and Yuji Matsumoto. 2011. Using the mutual k-nearest neighbor graphs for semi-supervised classification on natural language data. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 154–162, Portland, Oregon, USA. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Yong Ren, Nobuhiro Kaji, Naoki Yoshinaga, Masashi Toyoda, and Masaru Kitsuregawa. 2011. Sentiment classification in resource-scarce languages by using label propagation. In *Proceedings of the 25th Pacific Asia Conference on Language, Information and Computation*, pages 420–429, Singapore. Institute of Digital Enhancement of Cognitive Processing, Waseda University.

Ghivvago Damas Saraiva, Rafael Anchiêta, Francisco Assis Ricarte Neto, and Raimundo Moura. 2021. A semi-supervised approach to detect toxic comments. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 1261–1267, Held Online. INCOMA Ltd.

Iustin Sirbu, Tiberiu Sosea, Cornelia Caragea, Doina Caragea, and Traian Rebedea. 2022. Multimodal semi-supervised learning for disaster tweet classification. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2711–2723, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval.

Adil Yaseen Taha, Sabrina Tiun, Abdul Hadi Abd Rahman, Masri Ayob, and Ali Sabah Abdulameer. 2022. Unified Graph-Based Missing Label Propagation Method for Multilabel Text Classification. *Symmetry*, 14(2):286.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton

Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Jesper E. van Engelen and Holger H. Hoos. 2020. A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440.

Joe H. Ward. 1963. Hierarchical Grouping to Optimize an Objective Function. *Journal of the American Statistical Association*, 58(301):236–244.

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.

Ellery Wulczyn, Nithum Thain, and Lucas Dixon. 2017. Ex Machina: Personal Attacks Seen at Scale. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 1391–1399, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.

Weijie Xu, Wenxiang Hu, Fanyou Wu, and Srinivasan Sengamedu. 2023. DeTiME: Diffusion-enhanced topic modeling using encoder-decoder based LLM. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9040–9057, Singapore. Association for Computational Linguistics.

Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. 2021. A Survey on Deep Semi-supervised Learning.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd Annual Meeting of the Association for Computational Linguistics*, pages 189–196, Cambridge, Massachusetts, USA. Association for Computational Linguistics.

Gregory Yauney and David Mimno. 2021. Comparing text representations: A theory-driven approach. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5527–5539, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhiquan Ye, Yuxia Geng, Jiaoyan Chen, Jingmin Chen, Xiaoxiao Xu, SuHang Zheng, Feng Wang, Jun Zhang, and Huajun Chen. 2020. Zero-shot text classification via reinforced self-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3014–3024, Online. Association for Computational Linguistics.

Tianyi Zhang, Felix Wu, Arzoo Katiyar, Kilian Q Weinberger, and Yoav Artzi. 2021. Revisiting few-sample {bert} fine-tuning. In *International Conference on Learning Representations*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2016. Character-level Convolutional Networks for Text Classification.

Yichu Zhou and Vivek Srikumar. 2021. DirectProbe: Studying representations without classifiers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5070–5083, Online. Association for Computational Linguistics.

Xiaojin Zhu and Zoubin Ghahramani. 2002. Learning from Labeled and Unlabeled Data with Label Propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon University.

A Appendix

A.1 Experimental Details

Datasets. For Wiki Toxic and Civil Comments, we have applied a pre-processing consisting of removing non-alphanumeric characters.

Models. The following table shows details of the models used in the experiments, including the dataset card, the number of parameters, the associated embedding, and its corresponding vector dimension.

Model	#par.	Embedding	dim.
$\text{BERT}_{\text{base-unc.}}$[3]	110M	BERT_2^{μ}	768
SwahBERT[4]	110M	SwahBERT_2^{μ}	768
$\text{RoBERTa}_{\text{base}}$[5]	125M	RoBERTa_1^{μ}	768
LLaMA2-13B[6]	13B	LLaMA2_1^{w}	5120

Hyper-parameters. Table 5 contains a summary of the hyper-parameters used in the experiments.

A.2 Learning Curves

Figure 4 extends the comparative analysis in Section 4.1 on the effect of the initial seed sample for other semi-supervised techniques considered in this work, namely self-training and random walk label propagation.

A.3 Time Efficiency

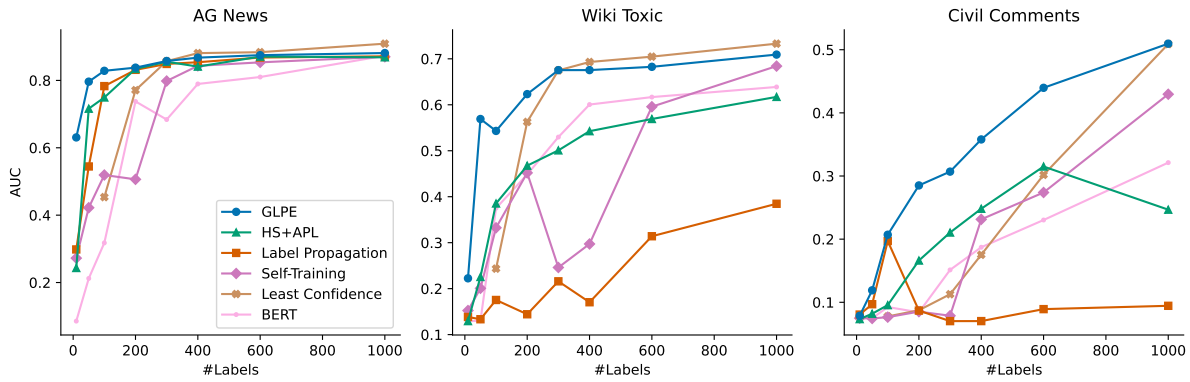
In Table 6 we show the average running times of the different tasks performed in our experiments.

Experiments were performed using a single Tesla V100 GPU or on the CPU, depending on the task.

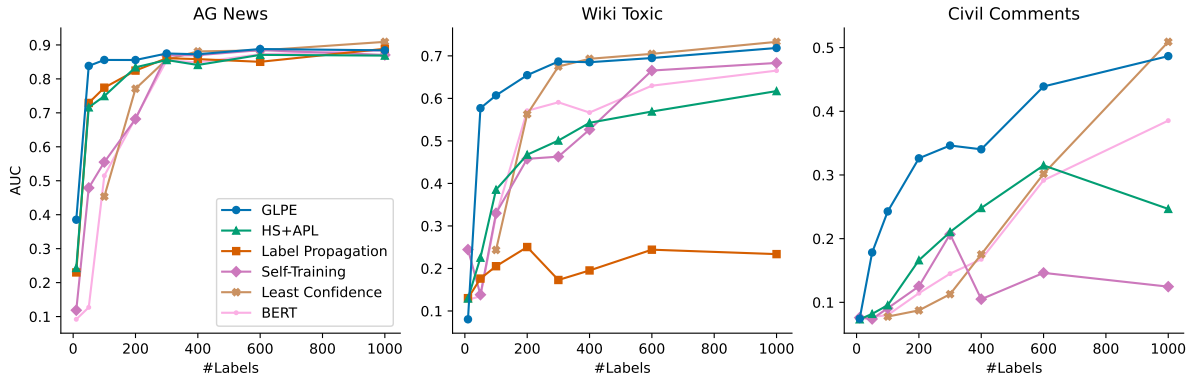
Hyper-parameter	**Value**
RWLP kernel	7-NN
RWLP max. iterations	1000
RWLP convergence tolerance	10^{-3}
Self-Training threshold	0.75
Self-Training max. iterations	10
Self-Tr. max. iter. (Civ.Com.)	1
HS+APL dendrogram leaves	10^4
HS+APL β	2.0
MixText sharpen temp.	0.5
MixText α	16
MixText mix layer set	$\{7, 9, 12\}$
kNN k	4000
GLPE k	4000
Training epochs	20
Learning rate	$5 \cdot 10^{-5}$
AdamW λ	0.0
AdamW β_1	0.9
AdamW β_2	0.999
Attention dropout	0.1
Hidden dropout	0.1
Mixed Precision	fp16
Seq. length (AG News)	128
Seq. length (Wiki Toxic)	150
Seq. length (Civil Comments)	150
Seq. length (TREC)	128
Seq. length (Ohsumed)	512
Seq. length (Swahili News)	512
Batch size (AG News)	32
Batch size (Wiki Toxic)	50
Batch size (Civil Comments)	50
Batch size (TREC)	32
Batch size (Ohsumed)	32
Batch size (Swahili News)	32

Table 5: Summary of hyper-parameters used in the experiments.

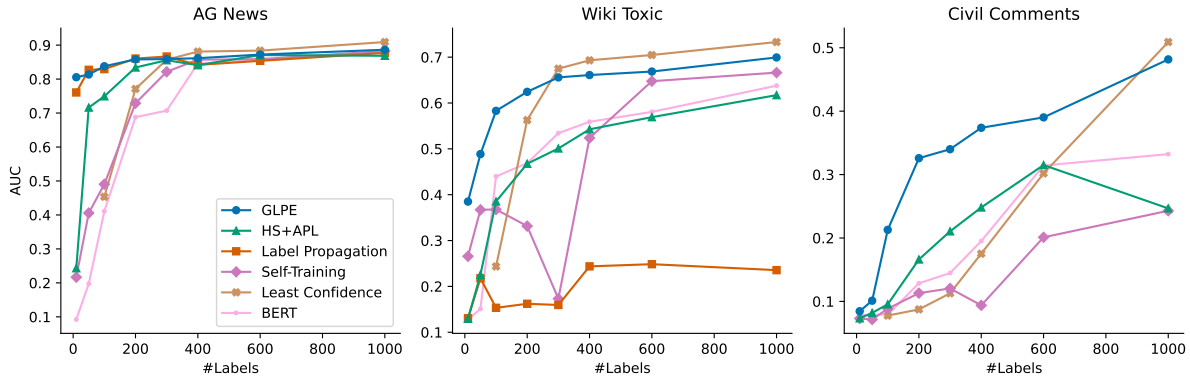
[3] huggingface.co/bert-base-uncased
[4] huggingface.co/pranaydeeps/SwahBERT-base-cased
[5] huggingface.co/roberta-base
[6] huggingface.co/meta-llama/Llama-2-13b-hf



(a) Random



(b) Hierarchical Sampling



(c) Cluster Diversity

Figure 4: Learning curves showing the performance (AUC) as a function of the annotation budget for various imbalanced datasets, comparing different semi-supervised techniques and Least Confidence active learning strategy. Each sub-figure corresponds to a seed sampling selection strategy.

Task	AG News	Wiki Toxic	Civil Com.	TREC	Ohsumed	Swahili News
BERT Embedding	29m 56s	20m 03s	4h 04m	43s	1m 29s	5m 28s
Hier. Clustering	23s*	26s*	23s*	5s*	26s*	22s*
RW lab. prop.	2.7s*	59.4s*	1h 46m*	1.1s*	3.4s*	5.2s*
GLPE lab. prop.	34ms*	25ms*	289ms*	12ms*	48ms*	58ms*
BERT train	23s	1m 32s	1m 51s	33s	2m 45s	2m 38s
Self-Training	7m 54s	1h 19m	20h 46m	42s	54s	54s
Label Prop. train	10m 54s	1h 24m	5h 54m	4m 6s	31m	1h 9m
HS+APL train	5m 48s	9m 18s	10m 48s	3m 54s	22m 54s	30m 18s
MixText train	43m 41s	1h 24m	1h 35m	31m 29s	1h 16m	1h 47m
SAT train	3m 32s	15m 47s	21m 56s	1m 09s	14m 44s	19m 04s
kNN	0.2s*	20.6s*	91.1s*	0.1s*	0.8s*	0.8s*
GLPE train	3m 47s	3m 22s	3m 55s	2m 56s	13m 17s	12m 55s
iGLPE train	7m 01s	10m 19s	9m 23s	6m 38s	29m 05s	28m 11s

Table 6: Average running times of different tasks. '*' means computed in the CPU.