Analyzing Text Representations by Measuring Task Alignment

Cesar Gonzalez-Gutierrez, Audi Primadhanty, Francesco Cazzaro, Ariadna Quattoni
Universitat Politècnica de Catalunya, Barcelona, Spain
{cesar.gonzalez.gutierrez, audi.primadhanty, francesco.cazzaro}@upc.edu,
aquattoni@cs.upc.edu

Abstract

Textual representations based on pre-trained language models are key, especially in few-shot learning scenarios. What makes a representation good for text classification? Is it due to the geometric properties of the space or because it is well aligned with the task? We hypothesize the second claim. To test it, we develop a task alignment score based on hierarchical clustering that measures alignment at different levels of granularity. Our experiments on text classification validate our hypothesis by showing that task alignment can explain the classification performance of a given representation.

1 Introduction

Recent advances in text classification have shown that representations based on pre-trained language models are key, especially in few-shot learning scenarios (Ein-Dor et al., 2020; Lu et al., 2019). It is natural to ask: What makes a representation good for text classification in this setting? Is the representation good due to intrinsic geometric properties of the space or because it is well *aligned* with the classification task? The goal of this paper is to answer this question to better understand the reason behind the performance gains obtained with pre-trained representations.

Our hypothesis is that representations better aligned with class labels will yield improved performance in few-shot learning scenarios. The intuition is simple: in this setting, the limited number of labeled samples will only provide a sparse coverage of the input domain. However, if the representation space is properly aligned with the class structure, even a small sample can be representative. To illustrate this, take any classification task. Suppose we perform clustering on a given representation space that results in a few pure clusters (with all samples belonging to the same class). Then, any training set that 'hits' all the clusters can be representative. Notice that there is a trade-off between the number of clusters and their purity. A well-aligned representation is one for which we can obtain a clustering with a small number of highly pure clusters. Based on this, we propose a task alignment score based on hierarchical clustering that measures alignment at different levels of granularity: Task Hierarchical Alignment Score (THAS).

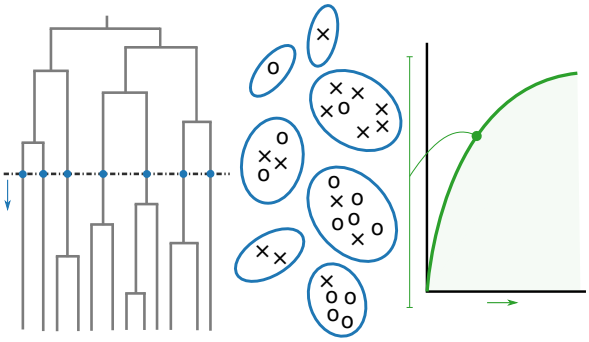

Figure 1: Three-step process for computing THAS.

To test our hypothesis that task alignment is key we conduct experiments on several text classification datasets comparing different representations. Our results show that there is a clear correlation between the THAS of a representation and its classification performance under the few-shot learning scenario, validating our hypothesis and showing that task alignment can explain performance. In contrast, our empirical study shows that intrinsic geometric properties measured by classical clustering quality metrics fail to explain representation performance in the few-shot learning scenario.

Our study suggests an answer to our main question: A good efficient representation (i.e. one that enables few-shot learning) is a representation that induces a good alignment between latent input structure and class structure. Our main contributions are: 1) We develop a score based on hierarchical clustering (§2) that measures the extent to which a representation space is aligned with a given class structure and 2) We conduct an empirical study using several textual classification datasets

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 2: Short Papers, pages 70–81
July 9-14, 2023

(§3) that validates the hypothesis that the best representations are those with a latent input structure that is well aligned with the class structure.

2 Task Hierarchical Alignment Score

We now present the Task Hierarchical Alignment Score (THAS) designed to measure the alignment between a textual representation and the class label for a given task. The idea is quite simple, in a good representation space, points that are close to each other should have a higher probability of belonging to the same class. Therefore, we could perform clustering of the points and obtain *high purity* clusters, where most points belong to the same class. We assume that we are given: a dataset $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$ of n labeled data points where $\boldsymbol{x} \in \mathcal{X}$ is a text fragment and $y \in \mathcal{Y}$ its corresponding class label (e.g., a sentiment classification label) and a representation function $r : \mathcal{X} \to \mathbb{R}^d$ mapping points in $\mathcal{X}$ to a d-dimensional representation space $\mathbb{R}^d$ (e.g., a sparse bag-of-words).

Our goal is to compute a metric $\tau(S, r)$ that takes some labeled domain data and a representation function and computes a real value score. Fig. 1 illustrates the steps involved in computing THAS. There are three main steps: 1) hierarchical clustering, 2) computing clustering partition alignments, and 3) computing the aggregate metric. In the first step, we compute the representation of each point and build a data dendrogram using hierarchical clustering. The data dendrogram is built by merging clusters, progressively unfolding the latent structure of the input space. Traversing the tree, for each level we get a partition of the training points into k clusters. In step 2, for each partition, we measure its alignment with the class label distribution producing an alignment curve as a function of k. Finally, we report the area under this curve. Algorithm 1 summarizes the whole procedure. Implementation details and performance information can be found in A.1.

Algorithm 1: THAS

Input: Dataset $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$, representation function r

Output: $\tau(S, r)$

1 Get representation:
$\boldsymbol{R} = \{r(\boldsymbol{x}_i) \mid \boldsymbol{x}_i \in \boldsymbol{X}\}$

2 Run Hierarchical Clustering:
$\mathcal{D} = \text{HC}(\boldsymbol{R}) = \{\mathcal{P}_k\}_{k=1}^n$

3 Traverse the dendrogram:
foreach *partition* $\mathcal{P}_k \subset \mathcal{D}$ **do**

4 Predict scores for all points:
foreach *point* $\boldsymbol{x}_i \in \boldsymbol{X}$ **in** $i = 1, \ldots, n$
where $r(\boldsymbol{x}_i) \in C \subset \mathcal{P}_k$ **do**

5 Label prediction scores:
foreach $y'_j \in \mathcal{Y}$ **in** $j = 1, \ldots, |\mathcal{Y}|$
do $\hat{\boldsymbol{Y}}_{k,i,j} = s(\boldsymbol{x}_i, y'_j)$

6 Partition alignment score:
$a(\mathcal{P}_k) = \text{AUC}_{y^+}(\hat{\boldsymbol{Y}}_k, \boldsymbol{Y})$

7 Final aggregate metric:
$\tau(S, r) = \frac{1}{n} \sum_{k=1}^n a(\mathcal{P}_k)$

2.1 Hierarchical Clustering

In the first step, we will consider the input points $\boldsymbol{X} = \{\boldsymbol{x}_i \mid (\boldsymbol{x}_i, y_i) \in S\}$ and the representation function r to obtain a representation of all points $\boldsymbol{R} = \{r(\boldsymbol{x}_i) \mid \boldsymbol{x}_i \in \boldsymbol{X}\}$.

We then apply Hierarchical Clustering (HC) to the points in $\boldsymbol{R}$ obtaining a dendrogram $\mathcal{D} = \text{HC}(\boldsymbol{R}) = \{\mathcal{P}_k\}_{k=1}^n$ that defines a set of n cluster partitions. Fig. 1 (left) shows a diagram of a dendrogram. The root of this tree is the whole set and, at the leaves, each point corresponds to a singleton. At intermediate levels, top-down branching represents set splitting.

For each level $k = 1, \ldots, n$ of the dendrogram there is an associated clustering partition of the input points into k clusters $\mathcal{P}_k = \{C_j\}_{j=1}^k$. That is, for any particular level we have a family of k nonempty disjoint clusters that cover the representation $\boldsymbol{R} = \bigcup_{j=1}^k C_j$, where each representation point $r(\boldsymbol{x}) \in \boldsymbol{R}$ is assigned to one of the k clusters.

2.2 Partition Alignment Score

We use the gold labels $\boldsymbol{Y} = \{y_i \mid (\boldsymbol{x}_i, y_i) \in S\}$ to compute an alignment score $a(\mathcal{P}_k)$ for each partition $\mathcal{P}_k \subset \mathcal{D}$. We compute it in two parts.

First, for every point $\boldsymbol{x} \in \boldsymbol{X}$ and label $y' \in \mathcal{Y}$ we compute a label probability score by looking at the gold label distribution of the cluster C to which the point belongs in the clustering partition:

$$s(\boldsymbol{x}, y') = \frac{1}{|C|} \#[y' \in C] \tag{1}$$

where $\#[y' \in C]$ is the number of samples in cluster C with gold label y'. Intuitively, this assigns to a point $\boldsymbol{x}$ a label probability that is proportional to the distribution of that label in the cluster C.

Second, we use the label probability scores of all points $\hat{\boldsymbol{Y}}_k = \{s(\boldsymbol{x}_i, y'_j) \mid \boldsymbol{x}_i \in \boldsymbol{X}, y'_j \in \mathcal{Y}\}$ and the

Repr.	ALC					THAS					ADBI				
	IM	WT	CC	S1	μ	IM	WT	CC	S1	μ	IM	WT	CC	S1	μ
BERT_{all}	*.84*	*.50*	*.32*	*.79*	**.61**	*.84*	*.67*	*.27*	*.75*	**.63**	2.87	3.03	3.31	3.25	3.11
GloVe	.80	.48	.26	.74	.57	.80	.63	.26	.73	.60	*2.62*	*2.12*	2.01	*2.47*	**2.31**
BERT_{cls}	.80	.48	.23	.74	.56	.80	.56	.22	.74	.58	2.81	2.97	3.15	2.92	2.96
fastText	.75	.41	.18	.66	.50	.77	.57	.21	.71	.56	2.78	2.13	*1.93*	*2.47*	2.33
BoW	.76	.32	.11	.59	.45	.71	.50	.20	.68	.52	3.14	3.83	4.23	3.86	3.76

Table 1: Learning curve performance (ALC), task alignment (THAS), and unsupervised clustering quality (ADBI) for different representations and datasets. (Rows are sorted by average ALC.)

dataset gold labels $\boldsymbol{Y}$ to compute a partition alignment score. We choose as a single metric the area under the precision-recall curve (AUC) because it has the nice property that it applies to tasks with both balanced and unbalanced class distributions.[1] More specifically, we compute the AUC of the target (positive) class $y^+ \in \mathcal{Y}$ of the dataset (more details in the experimental part in §3):

$$a(\mathcal{P}_k) = \text{AUC}_{y^+}(\hat{\boldsymbol{Y}}_k, \boldsymbol{Y}) \tag{2}$$

2.3 Final Aggregate Metric: THAS

Once we have an alignment score for every level of the hierarchical dendrogram, we are ready to define our final Task Hierarchical Alignment Score (THAS). Consider the alignment scoring function a applied to the partition corresponding to the lowest level of the dendrogram. The alignment score will be $a(\mathcal{P}_n) = 1$ because every cluster in this partition is a singleton and therefore $\#[y' \in C]$ will be 1 for the gold label and 0 for any other label. At the other end, for the partition corresponding to the root of the dendrogram (where all points belong to a single cluster), the alignment score $a(\mathcal{P}_1)$ is the AUC corresponding to assigning to every point $\boldsymbol{x} \in \boldsymbol{X}$ a prediction score for each label $y' \in \mathcal{Y}$ equal to the relative frequency of y' in $\boldsymbol{Y}$.

Consider now the alignment score as a function of the size of the partition. As we increase k we will get higher scores. A good representation is one that can get a high score while using as few clusters as possible. Instead of choosing a predefined level of granularity, we propose to leverage the alignment information across all levels. To achieve this, we consider the alignment score as a function of the number of clusters and measure the area under $a(\mathcal{P}_k)$.[2] We are ready to define our final metric:

$$\tau(S, r) = \frac{1}{n} \sum_{k=1}^{n} a(\mathcal{P}_k) \tag{3}$$

3 Experimental Setup

In this section we empirically study the correlation of few-shot learning performance with 1) THAS and 2) an unsupervised clustering quality metric.

We use four text classification datasets with both balanced and imbalanced label distributions: IMDB (IM; Maas et al., 2011), WikiToxic (WT; Wulczyn et al., 2017), Sentiment140 (S1; Maas et al., 2011) and CivilComments (CC; Borkan et al., 2019).

We will compare the following representations: a sparse bags-of-words (BoW); BERT embeddings (Devlin et al., 2019) using two token average pooling strategies (BERT_{all} and BERT_{cls}); GloVe (Pennington et al., 2014); and fastText (Bojanowski et al., 2017; Joulin et al., 2016).

For further details, please refer to A.2.

3.1 Few-Shot Performance vs. THAS

Since the focus of these experiments is comparing representations, we follow previous work on probing representations and use a simple model (Tenney et al., 2019; Lu et al., 2019). More precisely, we use a linear max-entropy classifier trained with $l2$ regularization.

To simulate a few-shot learning scenario, we create small training sets by selecting N random samples, from 100 to 1000 in increments of 100. For each point N in the learning curve we create an

[1] F1 could be a valid alternative, but this metric requires the validation of decision thresholds.

[2] We could consider weighting methods that neutralize uninformative areas in the curve. In particular, we could subtract the scores originating from a random clustering. However, this contribution is solely determined by the sample size and the prior distribution. As a result, it would not have any impact when comparing representations.

80%/20% 5-fold cross-validation split to find the optimal hyper-parameters. We then train a model using the full N training samples and measure its performance on the test set. We repeat the experiment with 5 random seeds and report the mean results. As the evaluation metric, we use accuracy for the balanced datasets (IMDB and Sentiment140) and F1 for the imbalanced datasets (WikiToxic and CivilComments).

We generate learning curves for each dataset and representation (A.3). To study the correlation between task alignment and few-shot learning performance, it is useful to have a single score that summarizes the learning curve: We use the area under the learning curve (ALC). Representations with a larger ALC perform better in the few-shot learning scenario.[3] We observe that BERT_{all} is consistently the best representation followed by BERT_{cls} and GloVe performing similarly. Representations based on word embeddings are better than the sparse baseline for all datasets, except for fastText which does not exhibit a consistent improvement.

To test for correlation, we also computed THAS for each representation and dataset. (The corresponding curves can be found in A.3.) Since this metric is a measure of the alignment between a label distribution and an input representation, there is a THAS score per label.[4] In the classification tasks that we consider there is always a single target class (e.g., toxicity for WikiToxic). We measure the alignment score with respect to this class.

Table 1 summarizes the results showing ALC (left) and corresponding THAS (center) for all representations and datasets. Overall, BERT_{all} is the best representation for few-shot learning followed by GloVe and BERT_{cls}. All the representations based on pre-trained word embeddings significantly outperform the baseline sparse BoW representation. THAS predicts accurately the relative ranking between representations and the larger gap between BERT_{all} and the rest. Fig. 2 shows a scatter plot of THAS as a function of ALC (blue dots; each point corresponds to a dataset and representation). We compute the correlation coefficients, which are displayed in Table 2. We observe a clear positive correlation between the two metrics, providing supporting evidence for our main hypothesis that a good representation under few-shot learning is a representation that is well aligned with the classification task.

[3]Alternatively, we could have picked a single point but we believe that ALC provides a more robust measure of few-shot learning performance and allows for a more concise analysis.

[4]We could also aggregate the scores of different classes, for example taking the average of the scores over all labels.

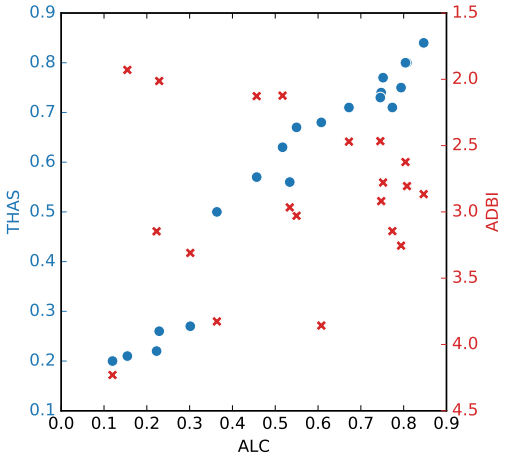

Figure 2: Few-shot performance (ALC) vs. task alignment (THAS) and clustering quality (ADBI).

(μ)ALC vs	r_p (p-value)	r_s (p-value)
THAS	0.98 ($< 10^{-12}$)	0.99 ($< 10^{-17}$)
ADBI	0.11 (0.62)	0.07 (0.76)
μTHAS	0.98 (0.002)	1.0 (0.017)
μADBI	-0.41 (0.48)	-0.3 (0.68)

Table 2: Pearson correlation coefficient (r_p) and Spearman's correlation coefficient (r_s) with the corresponding p-values for ALC vs. THAS and ALC vs. ADBI, and similar analysis for mean scores across all datasets.

3.2 Unsupervised Clustering Quality

We now look at standard metrics of cluster quality and test if they can explain few-shot learning performance. We use the Davies and Bouldin (1979) index (DBI) to measure the quality of the cluster partitions at every level of the dendrogram. This metric measures the compactness of each cluster and their separation, with better cluster partitions scoring lower. Similar to the computation of THAS described in §2, we compute DBI as a function of the number of clusters k corresponding to each level of the dendrogram. As an aggregate metric, we calculate the area under these curves to obtain a single ADBI score. (The curves are shown in A.3.)

The right side of Table 1 shows the results for the same datasets and representations used for THAS. GloVe induces the best clusters according to the ADBI metric. BERT_{all} does not produce particularly good clusters despite being the strongest few-

shot representation. Fig. 2 (red crosses) and Table 2 show that there is a low correlation between the two metrics. This suggests that the geometric properties of the clusters alone can not explain few-shot performance.

4 Related Work

Representation choice has recently gained significant attention from the active learning (AL) community (Schröder and Niekler, 2020; Shnarch et al., 2022; Zhang et al., 2017). Some work has attempted to quantify what representation is best when training the initial model for AL, which is usually referred to as the cold start problem (Lu et al., 2019). The importance of word embeddings has been also studied in the context of highly imbalanced data scenarios (Sahan et al., 2021; Naseem et al., 2021; Hashimoto et al., 2016; Kholghi et al., 2016). Most research conducted by the AL community on textual representations has focused on determining *which* representations lead to higher performance for a given task. However, our paper aims to investigate *why* a certain representation performs better in the few-shot scenario.

Our work, focused on examining properties of various textual representations, is closely related to recent research on evaluating the general capabilities of word embeddings. Many studies are interested in testing the behavior of such models using probing tasks that signal different linguistic skills (Conneau et al., 2018; Conneau and Kiela, 2018; Marvin and Linzen, 2018; Tenney et al., 2019; Miaschi and Dell'Orletta, 2020). Others have targeted the capacity of word embeddings to transfer linguistic content (Ravishankar et al., 2019; Conneau et al., 2020).

Looking at approaches that analyze the properties of representations directly, without intermediate probes, Saphra and Lopez (2019) developed a correlation method to compare representations during consecutive pre-training stages. Analyzing the geometric properties of contextual embeddings is also an active line of work (Reif et al., 2019; Ethayarajh, 2019; Hewitt and Manning, 2019). While these previous works focus on analyzing representation properties independently, without considering a specific task, our study investigates the relationship between representations and task labels. We conduct a comparison between this relationship and the unsupervised analysis of representation properties.

Our work falls in line with broader research on the relationship between task and representation. Yauney and Mimno (2021) proposed a method to measure the alignment between documents and labels in a given representation space using a data complexity measure developed in the learning-theory community. In the computer vision area, Frosst et al. (2019) introduced a loss metric and investigated the entanglement of classes in the representation space during the learning process. Zhou and Srikumar (2021) proposed a heuristic to approximate the version space of classifiers using hierarchical clustering, highlighting how representations induce the separability of class labels, thereby simplifying the classification task. In contrast, our work specifically examines the few-shot performance and emphasizes the importance of unbalanced scenarios. We find that in these more realistic situations, the choice of representation plays a critical role, paving the way for advanced strategies in active learning.

5 Conclusion

In this paper, we asked the question: What underlying property characterizes a good representation in a few-shot learning setting? We hypothesized that good representations are those in which the structure of the input space is well aligned with the label distribution. We proposed a metric to measure such alignment: THAS. To test our hypothesis, we conducted experiments on several textual classification datasets, covering different classification tasks and label distributions (i.e. both balanced and unbalanced). We compared a range of word embedding representations as well as a baseline sparse representation.

Our results showed that when labeled data is scarce the best-performing representations are those where the input space is well aligned with the labels. Furthermore, we showed that the performance of a representation can not be explained by looking at classical measures of clustering quality.

The main insight provided in this work could be leveraged to design new strategies in active learning. The fact that good representations induce clusters of high purity at different granularities creates opportunities for wiser exploration of the representation space in an active manner. Similar to the work of Dasgupta and Hsu (2008), we could employ the data dendrogram to guide this exploration.

Limitations

In this paper, we focused on analyzing the properties of textual representations in the few-shot learning scenario. Its applicability to broader annotation scenarios could be presumed but is not supported by our empirical results.

Our experimental setup is based on binary classification tasks using English datasets. While our approach is general and could be easily extended to multi-class scenarios, more work would be required to extend it to other more complex structured prediction settings such as sequence tagging.

We see several ways in which this work could be extended. The most obvious extension consists of trying to generalize the notion of alignment to other tasks beyond sequence classification, such as sequence tagging. In this paper, we have used THAS to understand the quality of a given textual representation. However, since THAS is a function of a labeling and a representation, it could also be used to measure the quality of a labeling (Yan and Huang, 2018), given a fixed representation. For example, this might be used in the context of hierarchical labeling, to measure which level of label granularity is better aligned with some input representation.

The goal of this paper was to provide an explanation for the success of pre-trained word embeddings for text classification in the few-shot learning scenario. We believe that with our proposed methodology we have successfully achieved this goal. However, it should be clear to the reader that we do not provide a method for picking the best representation, i.e. for model selection. This is because our analysis requires access to labeled data and if labeled data is available the best way to select a model will be via cross-validation.

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

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. Technical Report arXiv:1607.04606, arXiv. ArXiv:1607.04606 [cs].

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]*. ArXiv: 1903.04561.

Alexis Conneau and Douwe Kiela. 2018. SentEval: An evaluation toolkit for universal sentence representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Alexis Conneau, Shijie Wu, Haoran Li, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Emerging cross-lingual structure in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6022–6034, Online. 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.

David Davies and Don Bouldin. 1979. A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1:224–227.

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.

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.

Nicholas Frosst, Nicolas Papernot, and Geoffrey Hinton. 2019. Analyzing and improving representations with the soft nearest neighbor loss. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2012–2020. PMLR.

Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature*, 585(7825):357–362.

Kazuma Hashimoto, Georgios Kontonatsios, Makoto Miwa, and Sophia Ananiadou. 2016. Topic detection using paragraph vectors to support active learning in systematic reviews. *Journal of Biomedical Informatics*, 62:59–65.

John Hewitt and Christopher D. Manning. 2019. A structural probe for finding syntax in word representations. 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 4129–4138, Minneapolis, Minnesota. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of Tricks for Efficient Text Classification. Technical Report arXiv:1607.01759, arXiv. ArXiv:1607.01759 [cs].

Mahnoosh Kholghi, Lance De Vine, Laurianne Sitbon, Guido Zuccon, and Anthony Nguyen. 2016. The Benefits of Word Embeddings Features for Active Learning in Clinical Information Extraction. In *Proceedings of the Australasian Language Technology Association Workshop 2016*, pages 25–34, Melbourne, Australia.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jinghui Lu, Maeve Henchion, and Brian Mac Namee. 2019. Investigating the Effectiveness of Representations Based on Word-Embeddings in Active Learning for Labelling Text Datasets. ArXiv:1910.03505 [cs, stat].

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

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

Alessio Miaschi and Felice Dell'Orletta. 2020. Contextual and Non-Contextual Word Embeddings: an in-depth Linguistic Investigation. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 110–119, Online. Association for Computational Linguistics.

F. Murtagh. 1983. A Survey of Recent Advances in Hierarchical Clustering Algorithms. *The Computer Journal*, 26(4):354–359.

Usman Naseem, Matloob Khushi, Shah Khalid Khan, Kamran Shaukat, and Mohammad Ali Moni. 2021. A Comparative Analysis of Active Learning for Biomedical Text Mining. *Applied System Innovation*, 4(1):23.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in

Python. *Journal of Machine Learning Research*, 12:2825–2830.

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.

Vinit Ravishankar, Lilja Øvrelid, and Erik Velldal. 2019. Probing multilingual sentence representations with X-probe. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 156–168, Florence, Italy. Association for Computational Linguistics.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. 2019. Visualizing and Measuring the Geometry of BERT. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Marko Sahan, Vaclav Smidl, and Radek Marik. 2021. Active Learning for Text Classification and Fake News Detection. In *2021 International Symposium on Computer Science and Intelligent Controls (ISCSIC)*, pages 87–94. IEEE Computer Society.

Naomi Saphra and Adam Lopez. 2019. Understanding learning dynamics of language models with SVCCA. 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 3257–3267, Minneapolis, Minnesota. Association for Computational Linguistics.

Christopher Schröder and Andreas Niekler. 2020. A Survey of Active Learning for Text Classification using Deep Neural Networks. ArXiv:2008.07267 [cs] version: 1.

Eyal Shnarch, Ariel Gera, Alon Halfon, Lena Dankin, Leshem Choshen, Ranit Aharonov, and Noam Slonim. 2022. Cluster & tune: Boost cold start performance in text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7639–7653, Dublin, Ireland. Association for Computational Linguistics.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R. Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R. Bowman, Dipanjan Das, and Ellie Pavlick. 2019. What do you learn from context? Probing for sentence structure in contextualized word representations. ArXiv:1905.06316 [cs].

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.

Yi-Fan Yan and Sheng-Jun Huang. 2018. Cost-effective active learning for hierarchical multi-label classification. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 2962–2968. International Joint Conferences on Artificial Intelligence Organization.

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.

Ye Zhang, Matthew Lease, and Byron Wallace. 2017. Active Discriminative Text Representation Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1).

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.

A Appendix

A.1 THAS Implementation Details

The data dendrogram is obtained via hierarchical agglomerative clustering. More precisely, we use a bottom-up algorithm that starts with each sample as a singleton cluster and consecutively merges clusters according to a similarity metric and merge criterion until a single cluster is formed.

We apply Ward's (1963) method, which uses the squared Euclidean distance between samples and then minimizes the total within-cluster variance by finding consecutive pairs of clusters with a minimal increase. The clustering algorithm produces a list of merges that represent a dendrogram and can be traversed to generate a clustering partition for each value of k. It was implemented using Scikit-learn (Pedregosa et al., 2011) and NumPy (Harris et al., 2020).

Expressed as a nearest-neighbor chain algorithm, Ward's method has a time complexity of $\mathcal{O}(n^2)$ (Murtagh, 1983). THAS experiments have been performed using sub-samples of size 10K and averaged over 5 seeds. Using 32 CPUs and 16GiB of RAM, each agglomerative clustering took on average 3.3 minutes. Each task alignment curve took 3 minutes on average. In contrast, DBI curves took 7.8 hours on average.

A.2 Experimental Details

Datasets. Table 3 shows the statistics of the datasets used in this paper. They were extracted from HuggingFace Datasets (Lhoest et al., 2021). For WikiToxic and CivilComments, we have applied a pre-processing consisting of removing all markup code and non-alpha-numeric characters.

Dataset	Size	Prior	Task
IMDB	50K	50%	sentiment
WikiToxic	224K	9%	toxicity
Sentiment140	1.6M	50%	sentiment
CivilComments	2M	8%	toxic behav.

Table 3: Datasets statistics with the number of samples, target (positive) class prior, and classification task.

Representations. The following is a detailed description of the text representations used in our experiments:

BoW: this is a standard sparse term frequency bag-of-words representation.

BERT_{all}: word embeddings from Devlin et al.'s (2019) $\text{BERT}_{\text{BASE}}$ uncased model, average pooling of 2nd to last layers and average pooling of all tokens.

BERT_{cls}: the same as above but using the `[CLS]` token alone.

GloVe: Pennington et al.'s (2014) word vectors pre-trained on Common Crawl with average pooling.

fastText: word vectors from Bojanowski et al. (2017); Joulin et al. (2016) pre-trained on Wikipedia with average pooling.

BERT representations were extracted using the HuggingFace Transformers library (Wolf et al., 2020) implemented in PyTorch (Paszke et al., 2019).

Models. The parameters for max-entropy learning curves were validated using 5-fold cross-validation and the results averaged over sub-samples from 5 seeds.

A.3 Curves

Fig. 3 presents the curves used to compute the main results in §3. The left column contains the learning curves used to compute the few-shot learning performance of the different datasets and representations. The center column shows task alignment scores as a function of the number of clusters. THAS is computed as the area under these curves. The pre-trained word embeddings, in particular BERT, tend to achieve the best results. In the curves, they show higher values of alignment for a small number of clusters. The relative performance of the representations in the learning curves is paralleled in the task hierarchical alignment curves. BERT_{all} (i.e. using average pooling over all tokens) seems to be superior to BERT_{cls} (i.e. using only the `[CLS]` token).

The right column in Fig. 3 shows the DBI curves as a function of the number of clusters. These curves were used to compute the unsupervised clustering metric (ADBI) results presented in §3.2. As shown in the figure, these curves do not preserve the relative ranking we find in the corresponding learning curves.

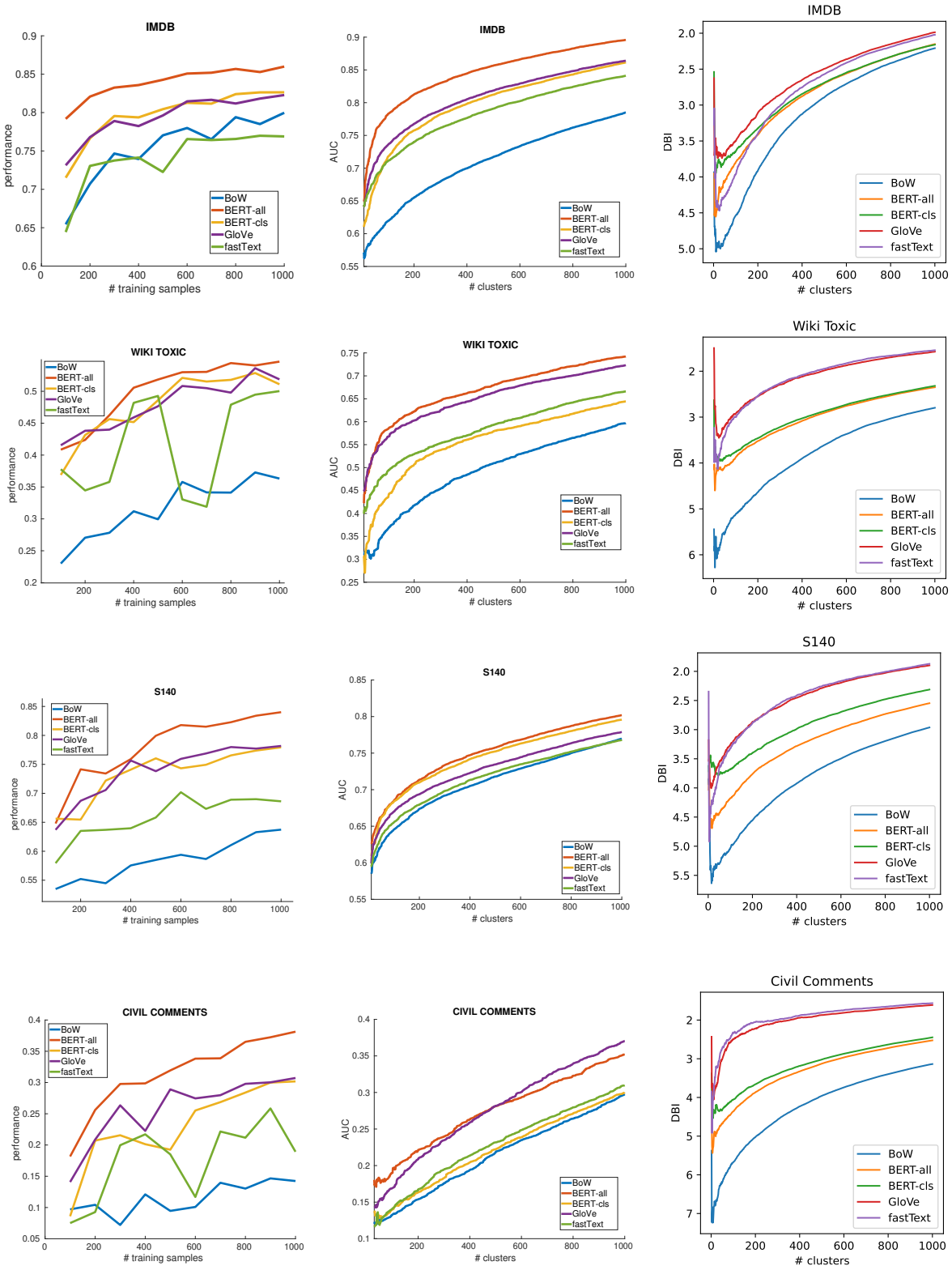

Figure 3: Learning curves (left), task hierarchical alignment curves (center), and DBI curves (right) for all the datasets: IMDB, WikiToxic, Sentiment140, and CivilComments.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Limitations (unnumbered)

☐ A2. Did you discuss any potential risks of your work?
Not applicable. Left blank.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract, 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

3

☑ B1. Did you cite the creators of artifacts you used?
3

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
A.2

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
A.2

C ☑ Did you run computational experiments?

3

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
A.2

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
3, A.2

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
3, A.2

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
A.2

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.