Unsupervised Out-of-Domain Detection via Pre-trained Transformers

Keyang Xu[1], **Tongzheng Ren**[2], **Shikun Zhang**[3], **Yihao Feng**[2], **Caiming Xiong**[4]
[1] Columbia University [2] University of Texas at Austin
[3] Carnegie Mellon University [4] Salesforce Research
kx2155@columbia.edu shikunz@cs.cmu.edu
{tzren, yihao}@cs.utexas.edu cxiong@salesforce.com

Abstract

Deployed real-world machine learning applications are often subject to uncontrolled and even potentially malicious inputs. Those out-of-domain inputs can lead to unpredictable outputs and sometimes catastrophic safety issues. Prior studies on out-of-domain detection require in-domain task labels and are limited to supervised classification scenarios. Our work tackles the problem of detecting out-of-domain samples with only unsupervised in-domain data. We utilize the latent representations of pre-trained transformers and propose a simple yet effective method to transform features across all layers to construct out-of-domain detectors efficiently. Two domain-specific fine-tuning approaches are further proposed to boost detection accuracy. Our empirical evaluations of related methods on two datasets validate that our method greatly improves out-of-domain detection ability in a more general scenario.[1]

1 Introduction

Deep neural networks, despite achieving good performance on many challenging tasks, can make overconfident predictions for completely irrelevant and out-of-domain (OOD) inputs, leading to significant AI safety issues (Hendrycks and Gimpel, 2017). Detecting out-of-domain inputs is a fundamental task for trustworthy AI applications in real-world use cases, because those applications are often subject to ill-defined queries or even potentially malicious inputs. Prior work on out-of-domain detection (e.g., Hendrycks and Gimpel, 2017; Lee et al., 2018; Liang et al., 2018; Hendrycks et al., 2019, 2020; Xu et al., 2020) mostly requires in-domain task labels, limiting its usage to supervised classification. However, deployed applications rarely receive controlled inputs and are susceptible to an ever-evolving set of user inputs that are scarcely labeled. For example, for many non-classification tasks, such as summarization or topic modeling, there are no available classifiers or task labels, which limits the practical usage of recently proposed out-of-domain detection methods. Therefore, it is natural to ask the following question:

Can we detect out-of-domain samples using only unsupervised data without any in-domain labels?

We regard the out-of-domain detection problem as checking whether the given test samples are drawn from the same distribution that generates the in-domain samples, which requires a weaker assumption than prior work (e.g., Lee et al., 2018; Hendrycks et al., 2020). We suppose that there are only in-domain samples, which allows us to understand the properties of data itself regardless of tasks. Therefore, methods developed for this problem are more applicable than task-specific ones and can be further adapted to tasks where no classification labels are present, such as active learning or transfer learning.

To solve the problem, we utilize the latent embeddings of pre-trained transformers (e.g., Vaswani et al., 2017; Devlin et al., 2019; Liu et al., 2019) to represent the input data, which allow us to apply classical OOD detection methods such as one-class support vector machines (Schölkopf et al., 2001) or support vector data description (Tax and Duin, 2004) on them.

However, the best practice on how to extract features from BERT is usually task-specific. For supervised classification, we can represent the text sequence using the hidden state of [CLS] token from the top layer. Meanwhile BERT's intermediate layers also capture rich linguistic information that may outperform the top layer for specific NLP tasks. By performing probing tasks on each layer, Jawahar et al. (2019) suggest bottom layers

[1]Code is available at https://github.com/rivercold/BERT-unsupervised-OOD.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 1052–1061
August 1–6, 2021.

of BERT capture more surface features, middle layers focus more on syntax and semantic features are well represented by top ones.

As no prior knowledge about OOD samples is usually provided in practice, deciding which layer of features is the most effective for OOD detection is itself non-trivial. Some OOD samples may just contain a few out-of-vocabulary words; while others are OOD due to their syntax or semantics.

Based on the observations above, this paper studies how to leverage all-layer features from a pre-trained transformer for OOD detection in an unsupervised manner. Our contributions are three-fold:

• By analyzing all layers of (Ro)BERT(a) models, we empirically validate that it is hard to extract features from a certain layer that work well for any OOD datasets.

• We propose a computationally efficient way to transform all-layer features of a pre-trained transformer into a low-dimension one. We empirically validate that the proposed method outperforms baselines that use one-layer features or by simple aggregations of all layers.

• We propose two different techniques for fine-tuning a pre-trained transformer to further improve its capability of detecting OOD data.

2 Problem Setup

Assume that we have a collection of text inputs $\mathcal{D}_n := \{\boldsymbol{x}_i\}_{i=1}^n$, we want to construct an out-of-domain detector that takes an unseen new input $\boldsymbol{u}$ and determines whether $\boldsymbol{u}$ comes from the same distribution that generates $\mathcal{D}_n$. We adopt a more practical setting where we have **no prior knowledge** of what out-of-domain inputs look like. In this case, training a domain classifier directly is not feasible. The out-of-domain detector can be described mathematically as:

$$g(\boldsymbol{u}, \epsilon) = \begin{cases} \textbf{True} & \text{if } \mathcal{I}(\boldsymbol{u}) \leq \epsilon\,, \\ \textbf{False} & \text{if } \mathcal{I}(\boldsymbol{u}) > \epsilon\,, \end{cases}$$

where $\mathcal{I}(\cdot)$ denotes the anomaly score function, and ϵ is a chosen threshold to ensure that the true positive rate is at a certain level (e.g., 95%) (Hendrycks and Gimpel, 2017; Liang et al., 2018; Lee et al., 2018). The OOD detection problem boils down to designing $\mathcal{I}(\cdot)$ such that it assigns in-domain inputs lower scores than out-of-domain inputs.

There are two different scenarios, considering if we have any in-domain labels for data $x_i \in \mathcal{D}_n$. Here we define in-domain labels as any specific supervised task labels, such as sentiments, intents or topics of the text.

With in-domain labels Suppose that we have multi-class label $y_i \in [K]$ and $\mathcal{D}_n = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$. Given a classifier h trained with $\mathcal{D}_n$, we can use maximum calibrated softmax probability with temperature scaling as the anomaly score (Liang et al., 2018; Hinton et al., 2015):

$$\mathcal{I}(\boldsymbol{x}) := -\max_{i\in[K]} \frac{\exp\left(h_i(\boldsymbol{x})/T\right)}{\sum_{j=1}^K \exp\left(h_j(\boldsymbol{x})/T\right)}\,,$$

where $h_i(\boldsymbol{x})$ is the output logits of the multi-class classifier, and T is the temperature that is selected such that the true positive rate is at a given rate (e.g., 95% in Liang et al. (2018)). This method is known as Maximum Softmax Probability (MSP), which requires multi-class labels to train a classifier and thus limits its application in practice. We argue that requiring in-domain labels is a less practical scenario for OOD detection and will not be further discussed it in this paper.

Without in-domain labels The setting of no in-domain labels is our major focus. Under this assumptin, the models we can obtain in hand are usually not classifiers, but feature extractors instead. Then it is natural to resort to classic outlier detection methods like one-class support vector machine (Schölkopf et al., 2001), support vector data description (Tax and Duin, 2004) or kernel density estimation (KDE) for estimating the support or the density of the in-domain data distribution.

When applying such methods to text data, the major focus of prior work is to design a good network structure or learning objectives (Ruff et al., 2018). Instead, in this paper we mainly focus on how to obtain good representations from pre-trained transformers and design new anomaly scores without modifying its structure, while still obtaining good OOD detection performance.

3 Model and Feature Learning

BERT and its variants such as RoBERTa (e.g., Devlin et al., 2019; Liu et al., 2019) are pre-trained on large-scale public data (denoted as $\mathcal{D}_{\text{pub}}$) using self-supervised tasks, such as language model and next sentence prediction. These models show

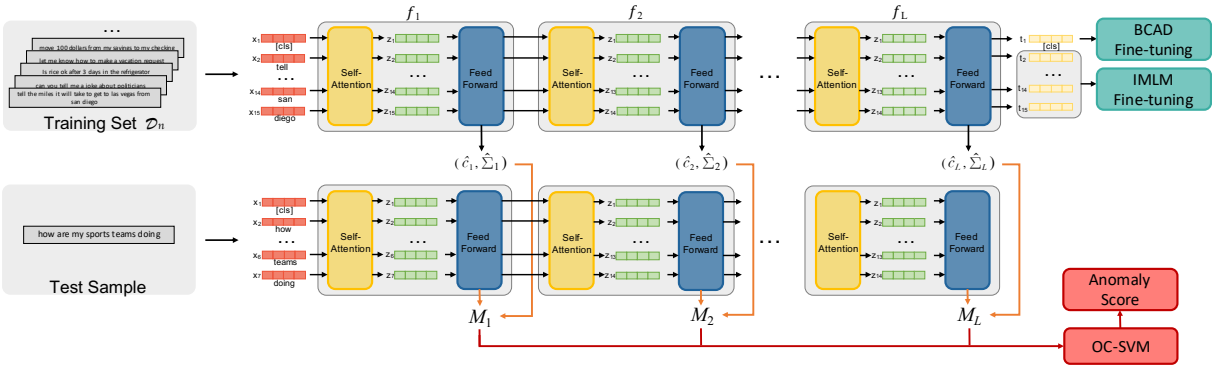


Figure 1: An overview of using Mahalanobis distance features (MDF) extracted from a pre-trained transformer f to detect out-of-domain data. We estimate mean $\hat{c}_l$ and covariance matrix $\hat{\Sigma}_l$ for each layer of f by samples from an unsupervised training set $\mathcal{D}_n$; and then extract MDF of $\mathcal{D}_n$ to optimize a OC-SVM. Given an unseen test sample, its feature M is extracted using $\hat{c}_l$ and $\hat{\Sigma}_l$ and then fed into OC-SVM for an anomaly score. Two domain-specific fine-tuning methods, IMLM and BCAD, can be further applied to BERT to boost detection accuracy.

promising results when transferred to tasks in other domains. We aim to leverage features obtained from pre-trained transformers to construct OOD detectors in lieu of in-domain labels in $\mathcal{D}_n$.

3.1 BERT features for OOD detection

After pretraining, we can obtain a BERT/RoBERTa model f with L layers. We denote $f_\ell(\boldsymbol{x}) \in \mathbb{R}^d$ as the d-dimensional feature embeddings corresponding to the ℓ-th layer for input $\boldsymbol{x}$, and $f(\boldsymbol{x})$ is the overall representation using all layers of f. We explore the following methods to extract BERT features to construct OOD detectors.

Features from the ℓ-th layer f_ℓ Options to extract $f_l(\boldsymbol{x})$ include using the hidden states of [CLS] token or averaging all contextualized token embeddings at the ℓ-layer. Then we can directly construct an OOD detector based on features from f_ℓ of each input $\boldsymbol{x}$ in $\mathcal{D}_n$ using existing pure sample based methods, such as one-class support vector machine (OC-SVM).[2]

Features from all layers Using BERT features from only one layer might not be sufficient, as prior work (Jawahar et al., 2019) has explored that different layers of BERT capture distinct linguistic properties, e.g., lower-level features capturing lexical properties, middle layers representing syntactic properties, and semantic properties surfacing in higher layers. The effects of BERT features from different layers on detecting OOD data are yet to be investigated. One straightforward way that leverages all L layers is to concatenate all layer-wise features $f_\ell(\boldsymbol{x})$, which has no information loss. However, this solution is computationally expensive and thus hard to optimize OC-SVM or kernel based methods. Another solution is to perform aggregation likes max- or mean-pooling along the feature dimension across all layers, sacrificing some information in exchange for efficiency.

In this paper, we propose a simple yet effective method (described below) to use latent representations from all layers of a pre-trained transformer and can automatically decide features from which layers are important. Besides, this method is computationally efficient, only requiring us to solve a low-dimensional constrained convex optimization.

Mahalanobis distance as features (MDF) for all layers Support Vector Data Description (SVDD) (Tax and Duin, 2004) is a technique related to OC-SVM where a hypersphere is used to separate the data instead of a hyperplane. However, the features provided by deep models may not be separable by hyperspheres. We focus on a generalization of the hypersphere called hyper-ellipsoid to account for such surface shapes.

Suppose that we use the concatenated features from all layers $\Phi(\boldsymbol{x}) = [f_1(\boldsymbol{x}), \ldots, f_L(\boldsymbol{x})]^\top \in \mathbb{R}^{d \cdot L}$ and consider the following optimization problem to find the hyper-ellipsoid, which is similar to the optimization formula of SVDD:

$$\min_{R, \boldsymbol{c}, \boldsymbol{\Sigma}, \boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{\Sigma}\|_{\text{Fr}}^2 + \left(R^2 + \frac{1}{\nu n}\sum_i \xi_i\right),$$
$$\text{s.t. } \|\Phi(\boldsymbol{x}_i) - \boldsymbol{c}\|_{\boldsymbol{\Sigma}^{-1}}^2 \leq R^2 + \xi_i\,,\ \xi_i \geq 0\,, \forall i\,, \quad (1)$$

[2] It is also possible to use other related one-class classification methods, such as Isolation Forest. However, in practice we find OC-SVM works the best and we use it in our empirical evaluations.

where Φ is the feature map, $\boldsymbol{c}$ is the center of the hyper-ellipsoid, and $\boldsymbol{\Sigma}$ is a symmetric positive definite matrix that reflects the shape of the ellipsoid. And R reflects the volume of the hyper-ellipsoid.[3] Here we also introduce a regularization term $\frac{1}{2}\|\boldsymbol{\Sigma}\|_{\text{Fr}}^2$ to constrain the complexity of $\boldsymbol{\Sigma}$. If $\boldsymbol{\Sigma} = \mathbf{I}$, then the optimization problem is identical to one-class SVDD.

Solving Eq (1) exactly can be difficult, since it involves finding the optimal $\boldsymbol{\Sigma}$ of shape $D \times D$, where $D = d \cdot L$ is the dimension of the features. For the concatenated features $\Phi(\boldsymbol{x})$, D can be tens of thousands or even hundreds of thousands, which makes the exact solution computationally intractable. To tackle the problem, we consider a **simple and computationally efficient** approximation of the solution, which can be useful in practice.

First, we decompose the feature space into several subspaces, based on the features from different layers, i.e., assume $\boldsymbol{\Sigma}$ is a block diagonal matrix, and $\boldsymbol{\Sigma}_\ell$ reflects the shape of feature distribution at layer ℓ. By a straightforward calculation, we have:

$$\|\Phi(\boldsymbol{x}) - \boldsymbol{c}\|_{\boldsymbol{\Sigma}^{-1}}^2 = \sum_{\ell=1}^{L} \|f_\ell(\boldsymbol{x}) - \boldsymbol{c}_\ell\|_{\boldsymbol{\Sigma}_\ell^{-1}}^2 ,$$

where we decompose the center $\boldsymbol{c}$ to be the center of each layer $\boldsymbol{c} = [\boldsymbol{c}_1, \ldots, \boldsymbol{c}_L]^\top$. Still, optimizing $\boldsymbol{c}_\ell$ and $\boldsymbol{\Sigma}_\ell$ can be difficult since the dimension of $f_\ell(\boldsymbol{x})$ can be high. Based on the intuition that $\boldsymbol{c}_\ell$ and $\boldsymbol{\Sigma}_\ell$ should not deviate from the empirical mean and covariance estimation $\widehat{\boldsymbol{c}}_\ell$ and $\widehat{\boldsymbol{\Sigma}}_\ell$ from the training data, we can replace $\boldsymbol{c}$ and $\boldsymbol{\Sigma}_\ell$ with the following approximation:

$$\boldsymbol{c}_\ell \approx \widehat{\boldsymbol{c}}_\ell = \frac{1}{n}\sum_{i=1}^{n} [f_\ell(\boldsymbol{x}_i)] ,$$
$$\boldsymbol{\Sigma}_\ell \approx \frac{\widehat{\boldsymbol{\Sigma}}_\ell}{w_\ell} = \frac{1}{(n-1)w_\ell}\sum_{i=1}^{n}(f_\ell(\boldsymbol{x}_i) - \widehat{\boldsymbol{c}}_\ell)(f_\ell(\boldsymbol{x}_i) - \widehat{\boldsymbol{c}}_\ell)^\top ,$$

where w_ℓ is a layer-dependent constant. Now we only need to find proper $\{w_\ell\}_{\ell=1}^L$ as well as the corresponding R and $\{\xi_i\}_{i=1}^n$, which is a low-dimension optimization problem that only scales linearly with the number of layer L. We further define:

$$M_\ell(\boldsymbol{x}_i) = (f_\ell(\boldsymbol{x}_i) - \widehat{\boldsymbol{c}}_\ell)^\top \widehat{\boldsymbol{\Sigma}}_\ell^{-1}(f_\ell(\boldsymbol{x}_i) - \widehat{\boldsymbol{c}}_\ell) ,$$

where the square root of $M_\ell(\boldsymbol{x}_i)$ is also referred to as the *Mahalanobis distance* of the features of data $\boldsymbol{x}_i$ from layer ℓ. Assume $\boldsymbol{w} = [w_1, \ldots, w_L]^\top \in \mathbb{R}^L$ and $M(\boldsymbol{x}) = [M_1(\boldsymbol{x}), \ldots, M_L(\boldsymbol{x})]^\top \in \mathbb{R}^L$, then we have:

$$\|\Phi(\boldsymbol{x}) - \boldsymbol{c}\|_{\boldsymbol{\Sigma}^{-1}}^2 = \langle \boldsymbol{w}, M(\boldsymbol{x}) \rangle .$$

As $\|\boldsymbol{\Sigma}\|_{\text{Fr}}^2 = \sum_{\ell=1}^{L} \frac{\|\widehat{\boldsymbol{\Sigma}}_l\|_{\text{Fr}}^2}{w_\ell^2}$ is not convex w.r.t $\boldsymbol{w}$, we instead minimize $-\frac{1}{2}\|\boldsymbol{w}\|_2^2$, which has a similar regularization effect on $\boldsymbol{\Sigma}$ (as we don't want $\|\boldsymbol{w}\|_2$ to be small, which can make $\|\boldsymbol{\Sigma}\|_{\text{Fr}}$ very large). So the final optimization problem to solve is:

$$\min_{R, \boldsymbol{w}, \boldsymbol{\xi}} -\frac{1}{2}\|\boldsymbol{w}\|_2^2 + R^2 + \frac{1}{\nu n}\sum_i \xi_i,$$
$$\text{s.t.} \quad \langle \boldsymbol{w}, M(\boldsymbol{x}_i) \rangle \le R^2 + \xi_i, \xi_i \ge 0 , \forall i , \quad (2)$$

which in fact is a one-class SVM with a linear kernel, with Mahalanobis distance of each layers as features (MDF), and it can be simply solved with the standard convex optimization. We illustrate our proposed algorithm in Figure 1.

Remark Note that the optimization in Eq (2) is not identical as that in Eq (1), since we are using empirical sample mean $\{\widehat{\boldsymbol{c}}_\ell\}_{\ell=1}^L$ and covariance $\{\widehat{\boldsymbol{\Sigma}}_\ell / w_\ell\}_{\ell=1}^L$ to replace the original parameters $\boldsymbol{c}$ and $\boldsymbol{\Sigma}$ in Eq (1), which are hard to optimize when the dimension of the concatenated features $\Phi(\boldsymbol{x})$ is high. Also, our approximation from Eq (1) to Eq (2) is different from the known result that when $\Phi(\boldsymbol{x})$ is the infinite-dimensional feature map of the widely used Gaussian RBF kernels, OC-SVM and SVDD are equivalent and asymptotically consistent density estimators (Tsybakov et al., 1997; Vert et al., 2006). In our case, $\Phi(\boldsymbol{x})$ is the concatenated features from all layers of pre-trained transformers, which makes our approximation fundamentally different from prior work.

3.2 Feature fine-tuning

We can also fine-tune the pre-trained transformer f on the **unsupervised** in-domain dataset $\mathcal{D}_n$ so that $f(\boldsymbol{x})$ can better represent the distribution of $\mathcal{D}_n$. We explore two domain-specific fine-tuning approaches.

In-domain masked language modeling (IMLM) Gururangan et al. (2020) find that domain-adaptive masked language modeling (Devlin et al., 2019) would improve supervised classification capability of BERT when it is transferred to that domain.

[3] We can further assume $\|\boldsymbol{\Sigma}\| = 1$, where the norm can be the operator norm or Frobenius norm, which can give the definition of the hyper-ellipsoid with unique $\boldsymbol{\Sigma}$ and R.

Cross-corpus Examples (SST)		
Type	**Source**	**Text**
In-Domain	SST	*if you love reading and or poetry , then by all means check it out*
In-Domain	SST	*there 's no disguising this as one of the worst films of the summer*
Out-of-Domain	RTE	*capital punishment is a deterrent to crime*
Out-of-Domain	SNLI	*a crowd of people are sitting in seats in a sports ground bleachers*
Out-of-Domain	Multi30K	*a trailer drives down a red brick road*
Cross-intent Examples (CLINIC150)		
Type	**Intent**	**Text**
In-Domain	Transfer	*move 100 dollars from my savings to my checking*
In-Domain	PTO Request	*let me know how to make a vacation request*
In-Domain	Food Last	*is rice ok after 3 days in the refrigerator*
In-Domain	Tell Joke	*can you tell me a joke about politicians*
Out-of-Domain	—	*how are my sports teams doing*
Out-of-Domain	—	*create a contact labeled mom*
Out-of-Domain	—	*what's the extended zipcode for my address*

Table 1: Examples of in-domain/out-of-domain samples for SST and CLINIC150. The source labels for SST and the intent labels for CLINIC150 are here just for illustration and are not included in $\mathcal{D}_n$. None of the above OOD samples are provided in training as well.

Similarly, we can do MLM on $\mathcal{D}_n$ and argue this would make the features of $\mathcal{D}_n$ concentrate, bringing benefits to downstream OOD detection.

Binary classification with auxiliary dataset (BCAD) Another way of fine-tuning the model f is to use the public dataset $\mathcal{D}_{\text{pub}}$ that pretrains it. We consider the training data in $\mathcal{D}_n$ as in-domain positive samples and data in the public dataset $\mathcal{D}_{\text{pub}}$ as OOD negative samples. We add a new classification layer on top of f and update this layer together with all parameters of f by performing a binary classification task. In practice, we only need a small subset of $\mathcal{D}_{\text{pub}}$, denoted as $\tilde{\mathcal{D}}_{\text{pub}}$, for fine-tuning. Since $\tilde{\mathcal{D}}_{\text{pub}}$ is publicly available and has no labels, we do not violate the unsupervised setting. $\tilde{\mathcal{D}}_{\text{pub}}$ does not provide any information about the OOD samples at test time as well.

Besides, the added classification layer can actually be applied for OOD detection using the MSP method, and this is exactly the setting of *zero-shot* classification, which we use as a baseline for comparison in our experiments.

4 Experiments

Datasets We consider two distinct datasets for experiments, where one is to regard text from unseen corpora as OOD, and the other one is to detect class-level OOD samples within the same corpus.

- **Cross-corpus dataset (SST)** We follow the experimental setting in Hendrycks et al. (2020), by providing in-domain $\mathcal{D}_n$ with the original training set of SST dataset (Socher et al., 2013) and considering samples from four other datasets (i.e., 20 Newsgroups (Lang, 1995), English-German Multi30K (Elliott et al., 2016), RTE (Dagan et al., 2005) and SNLI (Bowman et al., 2015)) as OOD data. For evaluation, we use the original test data of SST as in-domain positives and randomly pick 500 samples from each of the four datasets as OOD negatives. We do not include any sentiment labels from SST to $\mathcal{D}_n$ for training.

- **Cross-intent dataset (CLINIC150)** This is a crowdsourced dialog dataset (Larson et al., 2019), including in-domain queries covering 150 intents and out-of-domain queries that do not fall within any of the 150 intents. We use all 15,000 queries that are originally in its training data as in-domain samples but discard their intent labels. For evaluation, we mix the 4,500 unseen in-domain test queries with 1,000 out-of-domain queries and wish to separate two sets by their anomaly scores.

Examples taken from the two datasets can be found in Table 1. Note that for both datasets, only the in-domain samples are used for training, and the source/intent labels are not used in our experiments.

Evaluation metrics We rank all test samples by their anomaly scores and follow Liang et al. (2018) to report four different metrics, namely, Area Under the Receiver Operating Characteristic Curve (**AUROC**), Detection Accuracy (**DTACC**), and

Area under the Precision-Recall curve (AUPR) for in-domain and out-of-domain testing sentences respectively, denoted by **AUIN** and **AUOUT**.

Model configurations We evaluate all methods with both BERT and RoBERTa (`base` models with 768 latent dimensions and 12 layers).

Choice of $\tilde{\mathcal{D}}_{\text{pub}}$ for BCAD We adopt the BooksCorpus (Zhu et al., 2015) and English Wikipedia, which are the sources used in common by BERT and RoBERTa for pre-training. We split paragraphs into sentences and sample $\tilde{\mathcal{D}}_{\text{pub}}$ to have the same size as $\mathcal{D}_n$ for BCAD.

Baselines To examine the effectiveness of our newly proposed anomaly score based on MDF that utilizes the representations of all layers, we compare it with the following baselines.

• (Ro)BERT(a)-Single layer: It uses $f_\ell(\boldsymbol{x})$ mentioned above. We iterate all 12 layers and detailed results of each layer are discussed in Section 5.1.

• (Ro)BERT(a)-Mean pooling: we construct all-layer representation by averaging all $f_\ell(\boldsymbol{x})$, which has 768 dimensions.

• (Ro)BERT(a)-Max pooling: we aggregate all layers by picking largest values along each feature dimension and get a 768-dimension vector.

• (Ro)BERT(a)-Euclidean distance as features (EDF): we replace Mahalanobis distance with Euclidean distance and still obtain a 12-dimension vector.

• TF-IDF: we extract TF-IDF features and adopt SVD to reduce high-dimensional features to 100 dimensions for computational efficiency.

All of the above methods extract features as the input to OC-SVM to compute anomaly scores.

• BCAD + MSP: It performs zero-shot classification after BCAD fine-tuning, as discussed in Section 3. The temperature scaling is tuned to achieve the best result. This method is not applicable when no $\tilde{\mathcal{D}}_{\text{pub}}$ is provided.

5 Results and Discussions

In this section, we present the results for our experiments and summarize our findings.

5.1 Using single-layer feature $f_\ell(x)$

Table 2 shows results obtained from using the [CLS] embedding or averaging token embeddings (AVG) at each layer of (Ro)BERT(a) models in the cross-corpus and the cross-intent dataset.

Layer	SST				CLINIC150			
	BERT		RoBERTa		BERT		RoBERTa	
	CLS	AVG	CLS	AVG	CLS	AVG	CLS	AVG
12	**92.7**	**81.7**	**89.8**	**87.8**	61.5	60.2	53.4	51.6
11	88.8	66.3	88.8	68.8	57.3	59.0	51.6	55.5
10	87.7	52.1	79.6	68.4	56.6	55.4	53.8	56.2
9	85.5	50.7	84.2	67.2	56.8	56.5	58.3	56.5
8	82.9	57.6	78.7	67.7	61.6	55.8	**58.9**	56.0
7	85.8	59.2	83.6	67.5	**62.3**	63.0	57.5	56.4
6	76.4	61.9	73.0	67.8	58.2	62.3	55.5	56.7
5	74.2	58.2	63.5	67.2	56.3	62.8	56.2	57.1
4	66.7	67.4	70.0	69.8	61.9	60.9	52.7	57.8
3	65.8	67.5	62.9	69.3	54.3	59.4	51.0	58.5
2	62.6	63.2	75.7	68.8	60.4	58.6	55.6	**59.9**
1	68.1	63.5	70.0	71.0	60.9	**64.6**	55.6	58.5

Table 2: The **AUROC** scores of OOD detection on the SST/CLINIC150 dataset for each layer of BERT/RoBERTa. CLS denotes using the hidden state of the [CLS] token and AVG represents averaging all token embeddings in the same layer. Layer 12 indicates the top layer and layer 1 is the bottom layer right after the word embedding layer. The best result for each column is marked in bold.

We observe that detecting cross-intent OOD samples in CLINIC150 is more challenging than that of cross-dataset OOD data in SST. This is mainly because the OOD samples in CLINIC150 are sorted by humans and the differences between intents can be subtle. We will further compare the performance of these two settings in Figure 2.

The best $f_\ell(\boldsymbol{x})$ for OOD is dataset-specific For the cross-corpus dataset (SST), we find that the best results come from the top layer of both (Ro)BERT(a). However, for the cross-intent dataset (CLINIC150), the middle layers perform the best when using [CLS], while the bottom layers achieve the best results with AVG. This indicates that OOD distributions are not simply based on certain types of linguistic features and the strategy of choosing $f_\ell(\boldsymbol{x})$ is dataset-specific; for some dataset, semantic features play a more important role, while sometimes we need to focus on syntactic or lexical features. This validates the assumption that it is beneficial to fully utilize all layers of the hidden representations from pre-trained transformers to detect OOD instances.

We find using $f_\ell(\boldsymbol{x})$ of BERT is generally better

	#feats	SST				CLINIC150			
		AUROC	DTACC	AUIN	AUOUT	AUROC	DTACC	AUIN	AUOUT
BERT-Single layer (best)	768	92.7	85.8	93.4	91.7	64.6	60.9	88.4	26.7
RoBERTa-Single layer (best)	768	89.8	91.5	79.2	93.8	59.9	57.6	86.8	22.7
BERT + Mean-Pooling	768	81.8	76.5	77.2	82.8	62.9	59.9	87.0	27.9
BERT + Max-Pooling	768	67.2	66.1	64.2	59.4	63.0	60.0	88.0	25.8
RoBERTa + Mean-Pooling	768	91.0	92.3	80.9	94.5	57.1	56.2	85.7	20.5
RoBERTa + Max-Pooling	768	93.2	91.9	89.3	95.1	54.9	54.4	84.8	19.4
BERT + EDF	12	90.1	84.8	92.8	84.2	55.3	55.2	84.3	20.3
BERT + MDF	12	93.3	87.5	94.9	89.1	76.7	71.1	93.4	38.2
BERT + IMLM + MDF	12	93.6	88.1	97.5	89.4	77.8	72.2	93.8	39.1
BERT + BCAD + MDF	12	97.0	94.5	98.0	94.8	81.2	74.5	94.6	47.4
BERT + IMLM + BCAD + MDF	12	98.1	95.4	98.7	95.9	82.1	75.6	95.0	47.6
RoBERTa + EDF	12	99.5	95.8	99.5	99.4	56.9	56.9	86.3	19.6
RoBERTa + MDF	12	99.8	97.7	99.8	99.8	78.6	71.9	93.8	42.6
RoBERTa + IMLM + MDF	12	99.9	97.8	99.8	99.8	80.1	73.1	94.5	44.9
RoBERTa + BCAD + MDF	12	99.2	96.6	99.4	98.7	80.5	72.9	94.3	49.4
RoBERTa + IMLM + BCAD + MDF	12	**99.9**	**98.6**	**99.9**	**99.9**	**84.4**	**76.7**	**95.4**	**59.9**
TF-IDF + SVD	100	78.0	72.0	78.2	73.2	58.5	56.5	86.2	21.8
BERT + BCAD + MSP	-	68.5	69.0	61.5	65.4	68.3	63.5	89.7	34.1
RoBERTa + BCAD + MSP	-	73.7	69.3	69.0	75.3	62.1	59.6	85.9	27.8

Table 3: OOD detection performance on SST and CLINIC 150 for all models. OC-SVM is used for computing anomaly scores except MSP, and its parameters size is #feats. For (Ro)BERT(a)+Single-layer, the best results in Table 2 are reported. For all MDF-based model, we only report results of AVG as sequence representation at each layer due to space limit. Larger values of all four metrics indicate better performances. The best result for each metric is marked in bold.

than RoBERTa, especially with [CLS]. We guess next sentence prediction may cause this, which pre-trains on [CLS] and is exclusive for BERT.

In later sections, (Ro)BERT(a)-Single layer will refer to the best one in Table 2.

5.2 Overall OOD detection performance

We report the empirical results of OOD detection in Table 3 and the following observations.

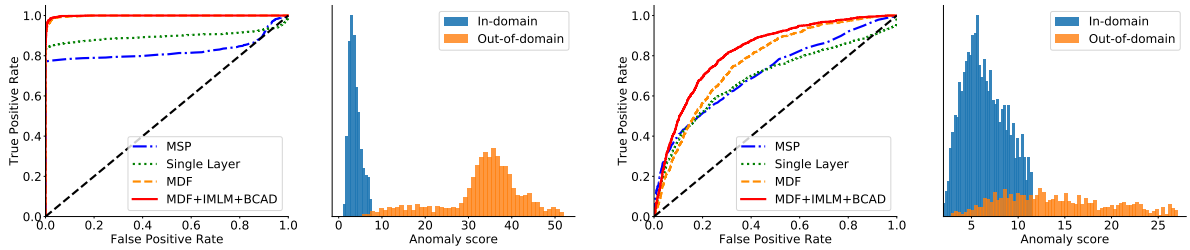
Pre-trained transformers produce good feature representations Methods using single-layer feature f_ℓ outperforms frequency-based features (TF-IDF) and zero-shot classification (MSP), which validates the strong representation capability granted by self-supervised pre-training.

Simple aggregations of all layers are not so effective The results of max-pooling and mean-polling are not very promising. Even though we observe an absolute 0.5% boost in SST using max-pooling, using the best single layer actually outperforms those simple aggregations in CLINIC150.

MDF is more effective MDF consistently outperforms methods that directly use features $f_\ell(\boldsymbol{x})$, simple aggregations of $f_\ell(\boldsymbol{x})$, or TF-IDF features on all four metrics. In terms of AUROC, MDF outperforms the best single-layer of (Ro)BERT(a) by absolute 7.1% on SST and 14.0% on CLINIC150.

MDF also performs better than EDF. Note that Euclidean distance is a special case of Mahalanobis distance when the covariance is an identity matrix. Empirically, the features generated by neural models are not invariant across all dimensions; and the comparison between MDF and EDF validates SVDD with a hyper-ellipsoid is better than a hypersphere.

MDF is more efficient in training OC-SVM Notice that our approach is also more computationally efficient when obtaining optimal $\boldsymbol{w}$ and $\boldsymbol{R}$ since the optimization is performed on a new transformed low dimensional data space ($d = 12$ is number of layers in f). See column #feats in Table 3 for detailed comparisons.



(a) ROC Curve on SST (b) $\mathcal{I}(\boldsymbol{x})$ on SST (c) ROC Curve on CLINIC150 (d) $\mathcal{I}(\boldsymbol{x})$ on CLINIC150

Figure 2: (a): ROC curves on the SST dataset. (b): Distribution of anomaly scores generated by IMLM + BCAD + MDF. Both figures are based on the BERT model. (c): ROC curves on the CLINIC150 dataset. (d): Distribution of anomaly scores generated by IMLM + BCAD + MDF.

	Sentence	GT	TF-IDF	Single	MDF
(a)	is a visa necessary for traveling to south africa	In	In	In	In
(b)	can you tell me who sells dixie paper plates	Out	In	Out	Out
(c)	can you tell me how to solve simple algebraic equations with one variable	Out	Out	In	Out
(d)	what oil is best for chicken	Out	In	In	In

Table 4: Examples of CLINIC150 with predictions from three models, which is "In" when sample's anomaly score is lower than 25th percentile and "Out" when larger than 75th percentile. GT is the ground truth and Single stands for BERT-Single.

Fine-tuning techniques improve performance From Table 3, we can see both MILM and BCAD improve OOD detection performance when incorporated with MDF separately. The overall best detecting performance is achieved by MILM + BCAD + MDF, combining both proposed fine-tuning methods with MDF.

We also find that RoBERTa outperforms BERT when using MDF, even though features from a single layer prefers BERT in Table 2.

5.3 Visualizations

We plot the ROC curves of four different anomaly scores on SST in Figure 2 (a) and on CLINIC150 in Figure 2 (c), confirming that our proposed MDF and two fine-tuning techniques improve the ability in detecting OOD samples. We also present the distributions of anomaly scores $\mathcal{I}(\boldsymbol{x})$ generated by our best method in Figure 2 (b) for SST and in Figure 2 (d) for CLINIC150. For SST, the OOD detector can clearly separate $\mathcal{I}(\boldsymbol{x})$ of in-domain and out-domain samples, and the in-domain scores are densely concentrated on the low-score region. Although for CLINIC150, we do observe some OOD samples mixing with in-domain ones, accounting for the gap of metric scores between two datasets.

5.4 Case Studies

We present some examples from CLINIC150 together with their corresponding predictions by TF-IDF, BERT-single layer and MDF methods in Table 4. TF-IDF predicts false positives for examples (b) and (d) because most of the words in the example test query are seen in the training set, like "i would like you to buy me some paper plates" (intent: order), "i need to know how long to cook chicken for" (intent: cooking time) and etc. BERT-single layer learns the syntax of "can you tell me how to ...", which is frequently seen in the training data, but it fails to discern that the semantic meaning is out-of-domain. For example (d), all models make the mistake, potentially associating it with the intent: recipe ("i need to find a good way to make chicken soup" or "what's the best way to make chicken stir fry").

6 Related Work

Out-of-domain detection is essentially an important component for trustworthy machine learning applications. There are two lines of work proposed to perform out-of-domain detection. One is to tackle the problem in specific multi-class classification tasks, where well-trained classifiers are utilized to design anomaly scores (e.g., Hendrycks and Gimpel, 2017; Liang et al., 2018; Lee et al., 2018; Card et al., 2019; Hendrycks et al., 2020; Xu et al., 2020), Those methods can only be useful when multi-class labels are available, which limits their application in more general domains. Our proposed work goes beyond this limitation and can utilize large amounts of unsupervised data.

Another line of work is based on support estimation or density estimation, which assumes that the in-domain data is in specific support or from the high density region (Schölkopf et al., 2001; Tax

and Duin, 2004). In principle, our work is closely related to this line of work. Besides, Zhai et al. (2016); Ruff et al. (2018); Zong et al. (2018) also leverage the features of neural networks, though these methods require designing specific network structures for different data. Our work circumvents the issues of prior work by designing a computationally efficient method that leverages the powerful representations of pre-trained transformers.

Finally, the fine-tuning techniques we use to improve the representation of data are closely related to unsupervised pre-training for transformers (Devlin et al., 2019; Yang et al., 2019), and recently proposed contrastive learning (e.g., He et al., 2020; Chen et al., 2020). Lately, Gururangan et al. (2020) discover that performing pre-training (MLM) on the target domain with unlabeled data can also help to improve downstream classification performance. To the best of our knowledge, our method is the first to incorporate transformers and pre-training techniques to improve out-of-domain detection.

7 Conclusion

We study the problem of detecting out-of-domain samples with unsupervised in-domain data, which is a more general setting for out-of-domain detection. We propose a simple yet effective method using Mahalanobis distance as features, which significantly improves the detection ability and reduces computational cost in learning the detector. Two domain-adaptive fine-tuning techniques are further explored to boost the detection performance.

In the future, we are interested in deploying our OOD method to real-world applications, such as detecting unseen new classes for incremental few-shot learning (Zhang et al., 2020; Xia et al., 2021) or filtering OOD samples in data augmentations.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable feedback and comments.

References

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, pages 632–642.

Dallas Card, Michael Zhang, and Noah A Smith. 2019. Deep weighted averaging classifiers. In *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT* 2019*, pages 369–378.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, pages 1597–1607.

I. Dagan, Oren Glickman, and B. Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine Learning Challenges, Evaluating Predictive Uncertainty, Visual Object Classification and Recognizing Textual Entailment, First PASCAL Machine Learning Challenges Workshop, MLCW 2005*.

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, NAACL-HLT 2019*, pages 4171–4186.

Desmond Elliott, S. Frank, K. Sima'an, and Lucia Specia. 2016. Multi30k: Multilingual english-german image descriptions. In *Proceedings of the 5th Workshop on Vision and Language*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, pages 8342–8360.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*, pages 9729–9738.

Dan Hendrycks and Kevin Gimpel. 2017. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *5th International Conference on Learning Representations, ICLR 2017*.

Dan Hendrycks, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. Pretrained transformers improve out-of-distribution robustness. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*, pages 2744–2751.

Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. 2019. Deep anomaly detection with outlier exposure. In *7th International Conference on Learning Representations, ICLR 2019*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does bert learn about the structure of language? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019*, pages 3651–3657.

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

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, EMNLP 2019*, pages 1311–1316.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems, NeurIPS 2018*, pages 7167–7177.

Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. 2018. Enhancing the reliability of out-of-distribution image detection in neural networks. In *6th International Conference on Learning Representations, ICLR 2018*.

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. *arXiv preprint arXiv:1907.11692*.

Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. 2018. Deep one-class classification. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, pages 4393–4402.

Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. 2001. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471.

R. Socher, Alex Perelygin, J. Wu, Jason Chuang, Christopher D. Manning, A. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013*, pages 1631–1642.

David MJ Tax and Robert PW Duin. 2004. Support vector data description. *Machine learning*, 54(1):45–66.

Alexandre B Tsybakov et al. 1997. On nonparametric estimation of density level sets. *The Annals of Statistics*, 25(3):948–969.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Régis Vert, Jean-Philippe Vert, and Bernhard Schölkopf. 2006. Consistency and convergence rates of one-class svms and related algorithms. *Journal of Machine Learning Research*, 7(5).

Congying Xia, Wenpeng Yin, Yihao Feng, and Philip Yu. 2021. Incremental few-shot text classification with multi-round new classes: Formulation, dataset and system. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021*, pages 1351–1360.

Hong Xu, Keqing He, Yuanmeng Yan, Sihong Liu, Zijun Liu, and Weiran Xu. 2020. A deep generative distance-based classifier for out-of-domain detection with mahalanobis space. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1452–1460.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Shuangfei Zhai, Yu Cheng, Weining Lu, and Zhongfei Zhang. 2016. Deep structured energy based models for anomaly detection. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1100–1109.

Jian-Guo Zhang, Kazuma Hashimoto, Wenhao Liu, Chien-Sheng Wu, Yao Wan, Philip S Yu, Richard Socher, and Caiming Xiong. 2020. Discriminative nearest neighbor few-shot intent detection by transferring natural language inference. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020*, pages 5064–5082.

Y. Zhu, Ryan Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. *2015 IEEE International Conference on Computer Vision, ICCV 2015*, pages 19–27.

Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. 2018. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *6th International Conference on Learning Representations*.